Mining arguments in scientific abstracts

Application to argumentative quality assessment

Pablo Accuosto

DOCTORAL THESIS UPF / Year 2021

THESIS SUPERVISOR

Prof. Horacio Saggion

Departament de Tecnologies de la Informació i les Comunicacions

Universitat Pompeu Fabra Barcelona

para Álvaro

Acknowledgments

Pursuing a PhD implies embarking on a journey where it is very easy to become disoriented. In moments of uncertainty it is crucial to have someone whose judgment you can trust, knowing that their advice will help you get back on track. Having walked this path, I am convinced that I could not have done it without the guidance and support of my supervisor, Horacio Saggion, to whom I am deeply grateful.

I would like to thank current and past members of the Large-Scale Text-Understanding Systems Lab (LaSTUS). Thanks in particular to Francesco Ronzano for his teachings, advice, and patience. I have been fortunate to be part of a very generous group of people, from whom I have learned a great deal. Among them, I am particularly grateful to Àlex Bravo, Francesco Barbieri, Ahmed AbuRa'ed and Daniel Ferrés. To carry out my research in the context of the Natural Language Processing Research Group (TALN) at UPF was a true privilege. I feel indebted in one way or another to all its members. Many thanks in particular to Mónica Domínguez for putting up with my mock presentations and for always giving valuable feedback. Thanks also to Laura Pérez Mayos for welcoming me to TALN, and for correcting the abstract of the thesis in Catalan.

Núria Bel was the first person I contacted to explore the possibility of doing my PhD at UPF. I especially value and appreciate Núria's help at that time and at many other times over the past five years when I asked her for advice.

I would also like to thank Aurelio Ruiz García for his constant support, patience and generosity, and Lydia García, who is always attentive to the doctoral students' needs, making things flow. Moltes gràcies. Many thanks also to the UPF IT people, especially those in charge of research support: Miquel Martínez, Daniel Valdés, Cristina Ocaña and Javier Ortega.

Thanks to Ricardo Baeza-Yates for listening to my very preliminary ideas for the thesis, and for giving me advice to which I have returned many times throughout this process. I also thank very much Ana Freire for sitting on my thesis proposal committee, and being one of the first people to give me such necessary feedback. Many thanks also to Laura Alonso Alemany for always being willing to listen, and for her accurate comments and suggestions.

I am especially grateful to Mariana Neves, who has been extraordinary generous and has played a fundamental role in my research, both with her annotations and with her feedback. I am also very grateful to Alicia Burga for her commitment, good disposition, and valuable suggestions. Thank you very much also to my co-authors. In particular, to Naiara Pérez, Montse Cuadros, and Germán Rigau.

One of the experiences I have to thank for in my time as PhD candidate is to have had the opportunity of doing a research stay at the Data and Web Science Group (DWS) at the University of Mannheim. My heartfelt thanks to Simone Ponzetto, Anne Lauscher and Goran Glavaš, who have been incredibly generous by hosting me and making me feel at home. Vielen Dank!

Many thanks to the anonymous reviewers of accepted and rejected papers, whose comments have been, in some cases, decisive in guiding my research. Thanks also to the members of the best paper committee of the 6th Argument Mining Workshop, Ivan Habernal, Graeme Hirst, and Jodi Schneider. Their recognition was fundamental to know that I was on the right track and pushed me to move forward. I would also like to thank very much Henning Wachsmuth and Elena Cabrio for agreeing to be on my jury, which I especially appreciate.

My PhD is the result of support that I have received from several people over the years. I am especially grateful to the Computer Science Institute (InCo) at the University of the Republic, and most especially to Dina Wonsever, to whom I have returned looking for advice many times after my period at InCo. Many thanks also to Verónica Dahl for her generosity and for always being an inspiration. I was very happy that my PhD gave us the opportunity to meet again after so many years.

Thanks to Mauricio Olivera for helping me to take the step to come to Barcelona and for continuing to help me once here in many ways. I also thank Ana Laura Rivoir for giving me the opportunity to return to an academic context after several years away, and for her constant friendship and support. I would also like to thank new and old friends who have played a decisive role in my being able to complete this process. Special thanks to Inés Campanella, without whom none of this would have been possible, and to Ana Martín, who has given me fundamental support during these years. Many thanks to my family, especially my parents. And to Álvaro Mailhos, thank you for everything, always.

This thesis has been possible thanks to funding received from the Spanish Government under the María de Maeztu Units of Excellence Programme (MDM-2015-0502) and from the National Agency for Research and Innovation (ANII) of Uruguay (POS_EXT_2016_1_135299).

Abstract

Argument mining consists in the automatic identification of argumentative structures in natural language, a task that has been recognized as particularly challenging in the scientific domain. In this work we propose SciARG, a new annotation scheme, and apply it to the identification of argumentative units and relations in abstracts in two scientific disciplines: computational linguistics and biomedicine, which allows us to assess the applicability of our scheme to different knowledge fields. We use our annotated corpus to train and evaluate argument mining models in various experimental settings, including single and multi-task learning. We investigate the possibility of leveraging existing annotations, including discourse relations and rhetorical roles of sentences, to improve the performance of argument mining models. In particular, we explore the potential offered by a sequential transfer-learning approach in which supplementary training tasks are used to fine-tune pre-trained parameter-rich language models. Finally, we analyze the practical usability of the automatically-extracted components and relations for the prediction of argumentative quality dimensions of scientific abstracts.

Resum

La mineria d'arguments consisteix en la identificació automàtica d'estructures argumentatives en el llenguatge natural, una tasca considerada com a especialment complexa en textos científics. En aquest treball proposem SciARG, un nou esquema d'anotació, i l'apliquem a la identificació d'unitats i relacions argumentatives en resums científics en dues disciplines: lingüística computacional i biomedicina, la qual cosa ens permet avaluar l'aplicabilitat del nostre esquema en diferents camps del coneixement. Utilitzem el nostre corpus per entrenar i avaluar models de mineria d'arguments en diversos contextos experimentals, entrenant cada tasca per separat i en un entorn multitasca. Investiguem la possibilitat d'aprofitar anotacions existents, incloent relacions de discurs i funcions retòriques d'oracions, per millorar el rendiment dels models de mineria de arguments. En particular, explorem el potencial que ofereix un enfocament d'aprenentatge per transferència en el qual s'utilitzen tasques d'entrenament suplementàries per afinar models lingüístics pre-entrenats. Finalment, analitzem l'ús pràctic dels components i relacions extretes automàticament dels textos per la predicció de diversos aspectes de la qualitat argumentativa de resums científics.

Contents

Chapter 1

INTRODUCTION

The accelerated pace at which new scientific knowledge is produced and communicated makes it challenging for scholars to keep up with the latest advances, even in their own research areas.[1] The growth in the generation and dissemination of scientific information also poses a barrier for the discovery and assessment of relevant findings by editors, research managers and decision makers, limiting and/or delaying the impact of scientific outcomes in the definition of evidence-based policies (Rogers, 2010).

Some elements in the evaluation of the scientific production necessarily require the intervention of human experts. This includes weighting the relevance of the problem at stake and the in-depth appraisal of the potential impact of the solutions proposed. Language technologies, however, can facilitate the assessment of other aspects of scientific communication. The verifiability of claims included in articles, their effectiveness with respect to its communication objectives, and their reliability in terms of the provided evidence are some areas in which natural language processing (NLP) tools can make a contribution.

Different quality aspects need to be considered when assessing the argumentative structure of a research paper, including its *logic*, *rhetoric* and *dialectic* dimensions

[1] In its most recent overview of scientific and scholarly publishing, in 2018, the International Association of Scientific, Technical and Medical Publishers (STM) indicate that *"the number of articles published each year and the number of journals have both grown steadily for over two centuries, by about 3% and 3.5% per year respectively. However, growth has accelerated to 4% per year for articles and over 5% for journals in recent years"* (Johnson et al., 2018).

(Wachsmuth et al., 2017a). This, in general, involves not only to identify the information that the authors provide in relation to what they do in their work and the conclusions that they draw from it–the *claims*–but also to consider the motivations/justifications for the proposed intervention and the evidence that they offer to support their assertions–the *premises*.

The automatic identification of arguments, its components and relations in texts is known as *argument mining* or *argumentation mining* (Lawrence and Reed, 2020). The steps involved in the automatic extraction of arguments from texts, including the identification of *claims* and *premises* and the prediction of the *argumentative structure* (how they are linked together) is not substantially different to other text-mining tasks for which supervised learning methods are generally applied (e.g., *text segmentation*, *sequence labeling* and *entity linking*) (Lippi and Torroni, 2016b). However, state-of-the-art results for these tasks are currently obtained by means of parameter-rich neural-based architectures that require large amounts of annotated data. The identification of *argumentative units* and *relations* in scientific texts, in particular, has been identified as a particularly challenging task–even for humans–due to the inherent complexity of the scientific discourse (Stab et al., 2014). Scarcity of annotated corpora, therefore, can hinder the advance of argumentation mining in this important domain.

1.1 Objectives

In the **first part** of this thesis we address the identification of argumentative *components* in scientific abstracts and the *relations* between them. The main goals of this part include:

i. To propose a new *annotation scheme* for the argumentative structure of scientific abstracts that can contribute to bridge an existing gap between various overlapping research areas, including: argument mining in scientific texts, rhetorical analysis of scientific discourse, and full-fledged discourse parsing.

ii. To apply and evaluate the proposed scheme in the annotation of abstracts in two scientific disciplines: *computational linguistics* and *biomedicine*, making the resulting corpus available as a contribution to the research community;

iii. To use the newly created corpus to *train and evaluate machine learning models* aimed at predicting the argumentative structure of abstracts–both at *sentence* and *intra-sentence* levels;

iv. To explore the possibility of *adapting* models trained with texts in one scientific discipline to predict the argumentative structure of abstracts in another discipline;[2]

v. To explore potential benefits obtained by *leveraging annotations available for related tasks*–in particular, discourse parsing and rhetorical classification of sentences–for mining arguments in scientific text;

vi. To evaluate two specific *transfer learning* approaches in the context of our tasks and domains: *supplementary training on intermediate tasks* and *multi-task learning*;

vii. To investigate whether benefits can be obtained–in particular, for the identification of argumentative components in scientific texts–by implementing a *rhetorical-complexity-aware* pipeline that allows *sentence-level* and *intra-sentence* level tasks to be addressed individually.

In the **second part** of the thesis we analyze the practical usability of the gold annotations and the predictions obtained with the models developed in the first part for the automatic *assessment of argumentative quality dimensions*. This includes:

viii. To analyze whether features obtained from the argumentative structure of scientific abstracts can contribute to predict scores reflecting *argumentative quality dimensions* of the abstracts and/or the full papers;

ix. To explore the potential benefits of incorporating *annotation-confidence* information in the training process for models aimed at predicting *quality scores*.[3]

[2]This, in turn, would shed some light about whether the argumentative structure of the abstracts encoded in these models is tied to the scientific discipline in which they are trained.

[3]A task with high levels of subjectivity and where with mixed levels of reliability can be obtained for the annotations, as we see in Chapter 8.

1.2 Contributions and outline

In the **first part** of the thesis we focus on the prediction of the *argumentative structure* of scientific abstracts:

- In Chapter 2 we contextualize our work within the fields or *argumentative mining* and the analysis of the *rhetorical* and *discourse* structure of scientific texts.
- In Chapter 3 we present the SciARG corpus of scientific abstracts. We describe our proposed *annotation scheme* and its application to the annotation of 225 abstracts in computational linguistics (SciARG-CL). We describe the annotation process and evaluate the agreement of the produced annotations.
- In Chapter 4 we use the SciARG-CL corpus to train and evaluate BERT-based (Devlin et al., 2019) models aimed at *predicting the argumentative structure of the abstracts*. We consider models trained for each task independently, as well as jointly, in multi-task settings. We investigate, in particular, the possibility of *leveraging existing discourse-level annotations* by considering, as an intermediate task, the prediction of discourse relations between sentences before fine-tuning the models for our target tasks.

When considering the first series of experiments, in Chapter 4, we argue that different methods should be applied for the identification of argumentative components at *sentence* and *intra-sentence* levels, and that the *rhetorical complexity* of sentences should be considered to decide which method(s) to apply in each case.

- In Chapter 5 we explore different ways of determining the *rhetorical complexity* of sentences and conduct experiments to identify the inner *rhetorical* and *argumentative* structures of sentences. We also investigate the possibility of leveraging existing annotations for these tasks. In this case, we consider existing annotations aimed at describing the *rhetorical role* both of sentences and intra-sentence segments in scientific abstracts, as well as annotations that establish *discourse relations* within sentences.

- In Chapter 6 we investigate the *adaptability* of the proposed annotation scheme to a scientific discipline different to the one for which it was originally developed, by extending the SciARG corpus with 285 abstracts in

bio-medicine (SciARG-BIO). We analyze similarities and differences between the new annotations and the ones in SciARG-CL, and report results obtained in experiments conducted with them. In particular, we examine whether language models fine-tuned with annotations in computational linguistics can be directly plugged-in in an architecture used to predict the argumentative structure of biomedical abstracts without further fine-tuning.

In the **second part** of the thesis we explore the potential use of argumentative units and relations predicted by means of the methods described in the first part in a downstream application. In particular, we analyze their potential to predict argumentative quality dimensions of the texts.

- In Chapter 7 we consider related work and antecedents in the area of argumentative quality assessment.
- In Chapter 8 we conduct experiments aimed at predicting quality scores assigned by annotators to the abstracts included in the SciARG-CL corpus.
- In Chapter 9 we use scores assigned by reviewers to manuscripts included in the ACL, CoNLL and ICLR sections of the PeerRead dataset (Kang et al., 2018), in order to investigate whether a set of features extracted from the argumentative structure of scientific abstracts can be used as predictors of specific argumentative quality dimensions of the papers in which they are included.

Finally, in Chapter 10 we summarize the main conclusions of our work and describe potential follow-ups.

1.3 Publications

Research conducted in the process of developing this thesis gave origin to peer-reviewed publications. We list them below.

- Accuosto, P., Neves, M., and Saggion, H. (2021). Argumentation mining in scientific literature: From computational linguistics to biomedicine. In Frommholz, I., Mayr, P., Cabanac, G., and Verberne, S., editors. *Proceedings of the 11th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 43rd European Conference on Information Retrieval (ECIR 2021)*; Lucca, Italy, Apr 1, 2021, Volume 2847 of CEUR Workshop Proceedings, p. 20-36.
- Accuosto P., and Saggion H. (2020). Mining arguments in scientific abstracts with discourse-level embeddings. *Data & Knowledge Engineering*, Volume 129, 101840, 2020.
- Accuosto P., and Saggion H. (2019a) Discourse-driven argument mining in scientific abstracts. In: Métais E, Meziane F, Vadera S, Sugumaran V, Saraee M, editors. *Natural Language Processing and Information Systems. 24th International Conference on Applications of Natural Language to Information Systems*; Salford, UK, Jun 26-28, 2019. Heidelberg: Springer; 2019. p. 182-94. (LNCS, no. 11608).
- Accuosto P., and Saggion H. (2019b). Transferring knowledge from discourse to arguments: A case study with scientific abstracts. In: Stein B., and Wachsmuth H., editors. *Proceedings of the 6th Workshop on Argument Mining*; Florence, Italy, Aug 1, 2019. Stroudsburg: Association for Computational Linguistics; 2019. p. 41-51. (Best paper award)

Part of the contents of Chapters 3 and 4 refer to work reported in (Accuosto and Saggion, 2019a,b, 2020), while part of the contents of Chapter 6 is reported in (Accuosto et al., 2021).

Some of the publications refer to preliminary work that is not included in the thesis but which has informed many of the decisions taken in the PhD research process. In Appendix A we summarize preliminary experiments and results reported in (Accuosto and Saggion, 2019b) to better contextualize some of these decisions.

Other works carried out in the period of the PhD research, and in the context of the María de Maeztu project *Mining the knowledge of scientific publications* include:

- Pérez N., Accuosto P., Bravo À., Cuadros M., Martínez-García E., Saggion H., and Rigau G. (2020). Cross-lingual semantic annotation of biomedical literature: Experiments in Spanish and English. *Bioinformatics*, Volume 36, Issue 6, 2020, p. 1872â1880.
- Bravo, À, Accuosto, P., and Saggion, H. (2019). LaSTUS-TALN@IberLEF 2019 eHealth-KD Challenge: Deep learning approaches to information extraction in biomedical texts. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019) co-located with 35th Conference of the Spanish Society for Natural Language Processing (SEPLN 2019)*; Bilbao, Spain, Sep 24, 2019. p. 51-59.
- Accuosto P, and Saggion H. (2018). Improving the accessibility of biomedical texts by semantic enrichment and definition expansion. *Procesamiento del Lenguaje Natural*, Issue 61, 2018, p. 57-64.
- Accuosto P., Ronzano F., Ferrés D., and Saggion H. (2017). Multi-level mining and visualization of scientific text collections. In *WOSP 2017 Proceedings of the 6th International Workshop on Mining Scientific Publications*; Jun 19, 2017; Toronto, Canada. New York: ACM; 2017. p. 9-16.
- Saggion H., Ronzano F., Accuosto P., and Ferrés D. (2017). MultiScien: A bi-lingual natural language processing system for mining and enrichment of scientific collections. In: Mayr P, Chandrasekaran MK, Jaidka K, editors. *Proceedings of the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017)*; Aug 11, 2017; Tokyo, Japan. CEUR Workshop Proceedings; 2017. p. 26-40.
- Abura'ed A., Chiruzzo L., Saggion H., Accuosto P., and Bravo À. (2017). LaSTUS/TALN @ CLSciSumm-17: Cross-document sentence matching and scientific text summarization systems. In: Jaidka K, Chandrasekaran MK, Kan MY. *Proceedings of the Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017)*; Aug 11, 2017; Tokyo, Japan. [Aachen]: CEUR-WS; 2017. p. 55-66.

Part I

Argumentative mining in scientific abstracts

Chapter 2

ARGUMENT MINING IN SCIENTIFIC TEXTS: BACKGROUND AND RELATED WORK

In this chapter we introduce *argument mining*, the research area in which our work can be contextualized, and briefly review literature related to it. The chapter is organized as follows:

- In Section 2.1 we describe the main tasks involved in argument mining and motivate research in the area based on its potential to support informed decision-making processes. We review the main domains and applications in which the identification of arguments have been focused, departing from in-depth surveys developed by other authors that offer a landscape of the progress of the area in the last decade.
- In Section 2.2 we focus on initiatives that have addressed the possibility of leveraging existing resources aimed at the analysis of discourse relations for the identification and characterization of arguments.
- In Section 2.3 we consider the few initiatives that have previously addressed the analysis of scientific texts from an argument mining perspective.

- In Section 2.4 we refer to key antecedents in the computational analysis of the rhetorical organization of scientific texts.
- Finally, in Section 2.5 we briefly refer to works in the area of transfer learning. In particular, to the transfer-learning methods that we adopt in our experiments: multi-task learning and sequential transfer of parameters trained with an intermediate supplementary task.

2.1 Argument mining

Argument mining (or *argumentation mining*) is a research area aimed at the automatic identification of arguments in natural language. This involves several tasks, which range from recognizing the internal components and structure of the arguments, to classifying them according to their characteristics, and to identify how they relate to each other.

In the past decade argument mining has progressively gained attention in the context of natural language processing (NLP) and computational linguistics (CL) research (Lawrence and Reed, 2020; Lippi and Torroni, 2016b). The increasing relevance of argument mining as a research area is evidenced by the inclusion of the topic in the calls for papers of the main venues in NLP/CL, such as the Annual Meeting of the Association for Computational Linguistics (ACL),[1] the International Conference on Computational Linguistics (COLING),[2] and the Conference on Empirical Methods in Natural Language Processing (EMNLP),[3] among others, as well as by the growing participation in the Argument Mining Workshop series (ArgMining), the premier research forum in the area, which is held annually at major NLP/CL conferences since 2014.[4]

After several years of active research in argument mining the area has expanded to a point were an in-depth review of all its branches is beyond the scope of this thesis. Stede and Schneider (2018) have taken on the task of writing a monograph volume on the subject, in which they provide a thorough description of the argument mining field and its evolution.

[1] aclweb.org/portal/acl
[2] aclanthology.org/venues/coling/
[3] aclanthology.org/venues/emnlp/
[4] uncg.edu/cmp/ArgMining2014/

They first contextualize the analysis of arguments within its philosophical and linguistic roots to then delve into the study of the main issues around computational approaches to argumentation–in particular, from an NLP perspective. They establish links to other related research fields in NLP, such as *subjectivity and sentiment analysis*, *semantic relation extraction*, and *discourse parsing*, among others, and offer a systematic review of the main corpora developed for argument mining research. A large proportion of the book is dedicated to analyzing the various sub-tasks involved in the identification of arguments in texts, their constituent parts and the relations between them, and it also provides insights into the automatic assessment of arguments' quality. Finally, the authors consider some of the main potential applications of argument mining.

In the following section we describe motivations for conducting research in argument mining and point to some of the key works in the area. For this, we rely on existing surveys, including the mentioned review by Stede and Schneider (2018).

2.1.1 Decision-making support systems

Being able to extract not only *what* is stated by the authors of a text or their *stance* towards a particular issue but also the *reasons* they provide to back up their claims can support a wide range of applications across multiple domains. An objective that captures great interest both in academia and the industry is the possibility of developing decision-making support systems based on arguments automatically extracted from natural language. Part of this interest has been fueled by IBM's Project Debater (Slonim et al., 2021), in development since 2012, which has recently found considerable echo in media.[5,6,7] The project is targeted at developing an argument-mining-based system that can debate humans on complex topics, with the ultimate goal of helping people to make well-informed decisions.[8]

[5] newyorker.com/news/annals-of-populism/the-limits-of-political-debate

[6] thetimes.co.uk/article/ibms-robot-debater-holds-its-own-against-human-opponents-zb8kwhxsl

[7] elpais.com/ciencia/2021-03-18/las-maquinas-ya-nos-ganaban-ahora-tambien-nos-convencen.html

[8] research.ibm.com/artificial-intelligence/project-debater/

Valuable resources have been made available to the research community in the context of this project,[9] including one of the first large-scale corpus with argumentative information, in which 547 Wikipedia pages were annotated with *claims* and *evidence* considered relevant for a set of pre-established topics (Aharoni et al., 2014; Rinott et al., 2015).

The development of argument-based systems to support decision-making processes is also relevant from an academic perspective, as it requires to find solutions to various difficult tasks, including *argument search and retrieval*, *summarization* and *visualization* of arguments. Active research in all of these areas is currently being conducted. We briefly summarize some initiatives below.

Initiatives in the area of *argument summarization* include (Wang and Ling, 2016), who address the generation of abstractive summaries for opinionated text by means of an attention-based neural model, and (Egan et al., 2016), who generate structured summaries of argumentative discussions based on relationships between *points* (short statements formed by a verb and its syntactic arguments) around five political debates from the Internet Argument Corpus (IAC) (Walker et al., 2012). (Syed et al., 2020) create a corpus of summaries for opinionated news editorials and evaluate two unsupervised extractive summarization models to identify the editorials' main argumentative thesis, and (Alshomary et al., 2020) propose an extractive *snippet* generation method to represent the *main claim* and *reason* of an argument. They rank the sentences of an argument with a variant of PageRank based on measures of a sentence *centrality* in its context and its degree of *argumentativeness*.[10] In line with these initiatives, in the 2021 Argument Mining workshop a new *key point analysis* shared-task was proposed (Bar-Haim et al., 2020) as a form of *quantitative summarization*. The goal of the task was, given a collection of argumentative texts on a certain topic, *"to produce a succinct list of the most prominent key-points in the input corpus, along with their relative prevalence"*. The idea is that such a summary could be used to gain insights from public opinions on topics of interest from multiple sources, which would *"give rise to a new form of a communication channel between decision makers and people that might be impacted by the decision"*.[11]

[9] `research.ibm.com/haifa/dept/vst/debating_data.shtml`

[10] These measures are based on the representation of sentences' embeddings as nodes in a graph where edges are weighted by the nodes similarity.

[11] `github.com/ibm/KPA_2021_shared_task`

In relation to argument search and retrieval research, Wachsmuth et al. (2017c) propose an adaptation to the PageRank algorithm (Page et al., 1999) to objectively *assess arguments relevance* for a search query at web scale. In addition, in (Wachsmuth et al., 2017b), the authors introduce an *argument search framework* and apply it to build a prototype search engine[12] as a practical demonstration of its capabilities.

Initiatives for the *visualization* of arguments, in turn, include the *VisArgue*[13] project for the analysis of political communication (El-Assady et al., 2017), and *DebateVis* (South et al., 2020), a tool aimed at the visualization of political debate transcripts, including the policies proposed by the different candidates and their debate techniques.[14] In turn, the Centre for Argument Technology at the University of Dundee (ArgTech) developed the *Argument Analytics* platform (Reed et al., 2018) which was used to generate visual analytics of debates in the context of a special BBC programme on the 50th anniversary of the UK's Abortion Act (Lawrence et al., 2018).[15]

The evaluation of various dimensions of *argumentative quality* in natural language is essential for ranking arguments in decision-making systems, but its relevance goes well beyond this specific use. In fact, it is a key component of most argument mining applications, including *argumentative writing support*–an issue addressed in depth by Stab (2017), as well as the related area of *essay scoring*–a topic for which Ke and Ng (2019) provide a thorough state-of-the-art review. In Chapter 7, in the second part of the thesis, we describe some of the most influential work in the area of *argumentative quality assessment*.

Several applications require–or could benefit from–the identification of arguments in natural language, and substantial research has been conducted to generate annotated data for multiple tasks and types of texts, as seen in Section 2.1.2. New emerging applications make us foresee an even greater development of the area in the coming years. This includes, for instance, the development of *argumentative-aware conversational search engines* (Kiesel et al., 2021) and a greater integration of argument mining approaches into related research fields, such as *automated fact-checking* (Guo et al., 2021).

[12] `args.me`
[13] `visargue.uni-konstanz.de/de`
[14] `osf.io/6jefk`
[15] `bbc.co.uk/programmes/b097c1g3`

Interest in the detection of argumentative information in *social media* is also growing. In particular, in relation to the detection of media bias and mis-information, the topic of a recent PhD thesis by Kailas (2021). In previous research, Lytos et al. (2019) consider the potential of argument mining applied to the analysis of social media posts' *reliability* as a way to detect fake news and prevent rumour diffusion. The application of argument mining in Twitter, in particular, is addressed by Schaefer and Stede (2021), where they review initiatives for the annotation of tweets with argumentative units, relations and stance. Another survey, by Vecchi et al. (2021), considers argumentation and social media from a different perspective: they focus on the potential of computational argumentation to address socio-political issues, and propose to leverage argument mining technologies for *social good*, which includes the development of applications for semi-automatic moderation. They suggest to integrate, in the definition of *argument quality*, the notion of *deliberative quality*, which considers the effects of the assessed contribution in the development of the upcoming discourse.

2.1.2 Application areas and domains

In-depth surveys of argument mining initiatives, including (Lippi and Torroni, 2016b; Cabrio and Villata, 2018; Lawrence and Reed, 2020), as well as one chapter of Christian's Stab PhD thesis (Stab, 2017), testify the evolution of the area at different points in time over the last decade. These surveys analyze initiatives for the generation of annotated corpora in various domains, as well as proposed approaches to address the sub-tasks involved in the identification of arguments, their structure and the relations between them.

Cabrio and Villata (2018) conduct a data-driven analysis of argument mining initiatives between 2015 and 2018, which provides a valuable insight into the applications, domains and textual genres that captured research interest in the years in which the area was consolidating.[16] They consider nine application areas organized in the following categories:

[16] As mentioned, the first Argument Mining workshop took place in 2014.

- Education
  - Persuasive essays
  - Scientific articles
- Web-based content
  - Wikipedia articles
  - Microblogs and web debating platforms
  - Online product reviews
  - Newspaper articles
  - Social media
- Legal documents
- Political debates and speeches

Stab (2017) classifies argument mining annotation initiatives based on various criteria, including the *argument granularity*, which indicates whether annotations are done at *macro* or *micro* levels in the taxonomy proposed by Bentahar et al. (2010). In the *macro* level properties of whole arguments and/or relations between them are considered, while, in the *micro* level, the annotations are aimed at identifying components within arguments. For initiatives at the *micro* level, the *component granularity* is also considered, which indicates the annotation unit (e.g., sentence, clause).

Lawrence and Reed (2020) focus, in particular, on datasets available on AIFdb (Lawrence and Reed, 2014),[17] a database created and maintained by ArgTech that contains corpora annotated with argument components and relations in the form of *argument maps*, which are made available in the argument interchange format (AIF). The goal of the initiative is to provide a standardized methodology for annotation, as well as a central location for the storage and retrieval of annotated corpora. The argument maps can be created and edited interactively by means of an online tool,[18] in which arguments are represented as directed graphs with two types of nodes: argumentative units and argumentative relations, labeled according to several possible argumentative frameworks.

[17] `aifdb.org`
[18] `ova.arg-tech.org`

Source/Genre	Works
Class discussions	(Lugini and Litman, 2020), (Olshefski et al., 2020)
Debate platforms	(Ajjour et al., 2019), (Al-Khatib et al., 2016), (Anand et al., 2011), (Biran and Rambow, 2011), (Boltužić and Šnajder, 2014), (Cabrio and Villata, 2012b), (Cabrio and Villata, 2013), (Habernal and Gurevych, 2017), (Kwon et al., 2007), (Liebeck et al., 2016), (Park and Cardie, 2018), (Rosenthal and McKeown, 2012), (Somasundaran and Wiebe, 2010), (Walker et al., 2012)
Essays	(Nguyen and Litman, 2018), (Stab and Gurevych, 2017a)
Legal texts	(Mochales-Palau and Moens, 2011), (Poudyal et al., 2020), (Teruel et al., 2018)
Micro-texts	(Peldszus and Stede, 2015a)
News	(Eckle-Kohler et al., 2015), (El Baff et al., 2018), (Sardianos et al., 2015)
Political discourse	(Dumani et al., 2021), (Duthie et al., 2016), (Lippi and Torroni, 2016a), (Menini et al., 2018), (Naderi and Hirst, 2015)
Product reviews	(García-Villalba and Saint-Dizier, 2012), (Ibeke et al., 2017)
Scientific art.	(Kirschner et al., 2015a), (Lauscher et al., 2018b)
Social media	(Dusmanu et al., 2017), (Schaefer and Stede, 2020), (Wührl and Klinger, 2021)
Multiple	(Cabrio and Villata, 2014), (Florou et al., 2013), (Goudas et al., 2014), (Lippi and Torroni, 2016c), (Niculae et al., 2017), (Reed et al., 2008), (Rinott et al., 2015)
Wikipedia	(Biran and Rambow, 2011), (Levy et al., 2014)

Table 2.1: Argument mining domains / textual genres. Based on surveys by Lippi and Torroni (2016b); Stab (2017); Cabrio and Villata (2018); Lawrence and Reed (2020).

From the referred surveys we extract a list of works (Table 2.1) that intend to be representative of the main types of corpora that have been produced to train and/or evaluate argument mining systems.[19] Since these surveys cover works until 2018, we complement them with some examples of corpora produced in the last couple

[19]In many cases, the annotated corpora evolve over time, being enriched/re-purposed for different tasks. In these cases, in general, we also include only one work in which the corpora is developed or used.

of years. We indicate, in each case, the domain/textual genre, adopting the taxonomy proposed in (Cabrio and Villata, 2018), which we extend with the category *Class discussions* (as a sub-category of *Education*), an emerging application area in argument mining.[20]

In the referred surveys, authors consider reported measures of inter-annotator agreement for the reviewed initiatives. While agreement levels vary depending on the complexity of the considered tasks and domains (Stab, 2017), there is a general consensus with respect to difficulty involved in the identification of arguments and its parts (Cabrio and Villata, 2018; Stab et al., 2014), due to the wide range for subjective interpretation of the speaker's intentions that frequently exists in the analysis of arguments. The identification of argument components and relations can become even more difficult when dealing with complex types of texts and/or in highly specialized domains, as we see in Section 2.3.

2.1.3 Tasks and schemes

Argument mining involves three main sub-tasks. Most works in the area deal with one or more of them, which can be processed independently, sequentially or jointly, as we see in more detail in Chapter 4. These tasks are:

1. *Identification of argumentative spans of text*
   This task involves classifying parts of a text (most frequently, sentences) either as *argumentative* or *non-argumentative*.
2. *Identification of argumentative components*
   This task consists in identifying the boundaries and/or types or the argumentative components within text previously classified as argumentative.
3. *Identification of the structure of arguments*
   This task involves establishing argumentative relations between argumentative components and/or whole arguments.

[20]We include in the category *Debate platforms* works that target citizen-participation platforms, such as (Kwon et al., 2007; Liebeck et al., 2016; Park and Cardie, 2018).

Lippi and Torroni (2016b) establish correspondences between each of these argument-mining tasks and other NLP tasks: *sentence classification* (e.g., sentiment analysis), *sequence labeling* (e.g., named-entity recognition), and *link prediction* (e.g., semantic textual similarity), respectively.

The relations considered in order to represent the structure of the arguments, as well as the types used to identify the arguments' components vary depending on the specific needs of each application. In the majority of the cases, the annotation schemes used in argument mining corpora derive from theoretical frameworks intended to formalize argumentative reasoning, such as Toulmin's model of arguments (Toulmin, 1958). Toulmin's model describes the different parts necessary in a well-formed argument (*claim*, *data*, *warrant*, *qualifier*, *rebuttal*, *backing*), and it has been adapted in several ways for its application to the computational analysis of the internal structure of arguments. Habernal and Gurevych (2017) analyze the suitability of Toulmin's model for mining arguments in a corpus of user-generated web content. They conclude that a modified version of the model containing five argument components (*claim*, *premise*, *backing*, *rebuttal*, and *refutation*) is suitable only for short persuasive documents with a clear standpoint on a controversial topic.

Various authors have pointed out limitations of Toulmin's model when applying it in practice to argument mining. Some of these limitations include the ambiguity in the definition of arguments' components,[21], as well as the narrow scope of the *rebuttal* type, which is not well suited to model other types of *attacks* (Freeman, 2011; Stab, 2017; Lauscher et al., 2018b).

Freeman (2011) proposes a theory of the structure of arguments that considers a hypothetical dialectical exchange between a *proponent* who defends (*supports*) a claim and an *opponent*, who questions (*attacks*) it. Peldszus and Stede (2013) propose a graph-based annotation scheme that incorporates this dialectical-based perspective with two main argumentative relations: *support* and *attack*. In this scheme, an annotated text is represented by means of a graph in which the nodes stand for argument components, called *argumentative discourse units* (ADU), of which one is identified as representing the *central claim*. Relations can be established between ADUs or between an ADU and another relation.

[21] Such as *data*, *warrant*, *backing*, which makes it difficult to choose one over another when annotating a text.

The model allows to represent that two (or more) ADUs can participate in a *joint support* relation with another one. The *support* relation can be further specified to indicate that one ADU provides an *example* for another one, while the *attack* relation can be sub-divided into two types: *rebuttal*, which is established when one ADU attacks another one (i.e., one ADU is intended to undermine the credibility of another one), and *undercut*, which is established when an ADU attacks a relation between two ADUs, (i.e., one ADU is intended to challenge the acceptability of the inference from the source to the target node). The GraPAT[22] web-based annotation tool contemplates the annotation of texts based on this scheme (Sonntag and Stede, 2014). The scheme and the tool were used to annotate a corpus of argumentative microtexts (Peldszus and Stede, 2015a), which we describe in more detail in Section 2.2.

In contrast to Toulmin's *micro-level* model of arguments, Dung's argumentation framework (Dung, 1995) is aimed at representing *attack* relations at the *macro* level (i.e., between whole arguments). Dung's framework has been used, for instance, by Cabrio and Villata (2012a), to predict the *acceptability* of arguments in online debates by considering how they are *attacked* by other arguments. This is done in practice by mapping Dung's *attack* relation to the *contradiction* relation in a textual entailment classification system.

Walton's *Argumentation Schemes* (Walton et al., 2008) is another influential model in argument mining. The *argumentation schemes* are templates intended to capture common structures of arguments used in everyday reasoning. One one hand, the schemes can be used to guide a reasoning/argumentation process and, on the other, to evaluate it. Each scheme therefore includes a set of *critical questions*, which represent defeasibility conditions that can be used to identify potential weaknesses of the arguments. The number and types of proposed schemes have changed over time. (Macagno et al., 2017) provide a detailed description of their evolution and potential uses.[23] Walton's schemes were adopted and used extensively, for instance, in the *Araucaria* system (Reed and Rowe, 2004), as a way of facilitating the identification and visualization of the structure of arguments in terms of their constituents and the relationships between them.

[22]Graph-based Potsdam Annotation Tool: `github.com/discourse-lab/GraPat`

[23]A list of schemes is available at `reasoninglab.com/patterns-of-argument/argumentation-schemes/waltons-argumentation-schemes`

2.2 Relation between argument mining and discourse analysis

Previous works have explored correspondences between the structure of arguments and discourse coherence relations considered in discourse-analysis frameworks, such as the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) or the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), including (Peldszus and Stede, 2013; Cabrio et al., 2013; Biran and Rambow, 2011; García-Villalba and Saint-Dizier, 2012; Stab et al., 2014; Green, 2018). According to Stede and Schneider (2018), in spite of all the existing analyses, this relation is *is so far not entirely clear* (Stede and Schneider, 2018, p.80).

Peldszus and Stede (2013) explore the possibilities offered by RST to represent argumentative structures. An RST analysis applies a set of relations[24] between adjacent spans of text. In most of the relations,[25] the linked spans do not have the same *relevance* in terms of the communicative intentions of the writer: the segment that carries the most significant information is called the *nucleus* and the segment with a supportive role is called the *satellite*. Most relations can, therefore, be seen as directed links from the *nucleus* to the *satellite*. An RST analysis is performed in a hierarchical way, linking together *elementary discourse units* (EDU)–the shortest meaningful segments–and then applying the same procedure recursively in a bottom-up fashion, where the *nucleus* of a relation is linked to another *nucleus*. This process ends up producing a *discourse tree* representing the text. An RST tree should therefore reflect a *compositional* criterion, in the sense that a relation between two nuclei reflects the relation between the two larger spans of text that contains them (Marcu, 2000). Not all RST relations are considered to have the same relevance from an argumentative point of view (although this depends on the argumentative dimensions considered, as we see in Chapter 7). Azar (1999), for instance, identifies only five RST relations[26] as necessary for an argument-oriented analysis of a text.

[24] The full set of 32 relations currently considered in RST, with their definitions and examples is available at `sfu.ca/rst/01intro/definitions.html`.

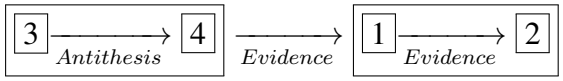
[25] Except seven relations considered as *multi-nuclear*.

[26] *Evidence*, *Motivation*, *Justify*, *Antithesis*, and *Concession*

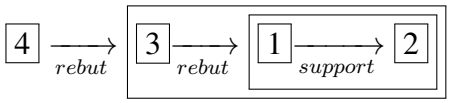
Peldszus and Stede (2013) identify obstacles that arise when trying to apply an RST analysis directly to the identification of the argumentative structure of a text, including the fact that argumentative relations do not necessarily respect the adjacency criteria adopted in *pure* RST. Another problem that they identify is the impossibility of representing some argumentative relations in RST should the compositionality principle be respected. This is the case, in particular, of *rebuttal* relations. To illustrate this problem, they provide the following example[27] with a fragment of an argumentation containing two sentences, with the EDUs enclosed in square brackets and the respective nuclei in bold:

> [The building is full of asbestos,]$_1$ [**so we should tear it down.**]$_2$
> [In principle it is possible to clean it up,]$_3$ [**but according to the mayor that would be forbiddingly expensive.**]$_4$

The relations in an RST analysis would be:



The problem is that this does not reflect the full argumentation contained in the text, which could be represented as:



In particular, the RST analysis does not contemplate the argumentative relation of *rebuttal* between the satellite EDU 3 and [1-2] (i.e: the RST analysis cannot explicitly reflect the fact that the *possibility of cleaning up the building* diminishes the strength of the argumentation conveyed by the first sentence.)

The example is useful to illustrate the differences in terms of the goals of the two types of analyses, but it also shows that relations between the two tasks, and triggers the question of whether argumentative graphs could be derived from discourse trees obtained by means of RST analyses. This is, in fact, is one of the main issues addressed by Peldszus and Stede (2016, 2015a); Stede et al. (2016).

[27] Based on an example of *rebuttal* and *counter-rebuttal* by (Freeman, 2011).

Stede et al. (2016) annotate 112 argumentatively rich texts with three annotation layers, one *argumentative* and two *discourse-oriented*, in order to study the relationship between discourse and argumentation structures. The arguments included in the corpus were generated in an experiment in which several participants wrote short texts of controlled linguistic and rhetorical complexity discussing a controversial issue from a pre-defined list of topics. The argumentation-level annotations are made by means of the scheme adapted from (Freeman, 2011) and described in (Peldszus and Stede, 2013), which we summarize in Section 2.1.3. For the discourse annotations they consider RST as well as the Segmented Discourse Representation Theory (SDRT) (Asher et al., 2003) and map the three annotation layers to a common dependency tree format in order to facilitate the analysis. At the argumentative level, the annotation units (ADUs) consist of one or more EDUs (which are shared by both discourse-level representations). They include a non-argumentative meta-relation, *join*, to link together EDUs that are part of the same ADU. Based on this corpus, the authors provide quantitative and qualitative analyses of commonalities and differences between the different levels of representation.[28]

As a follow-up to this work, Peldszus and Stede (2016) conduct experiments to explore the possibility of automatically deriving argumentative components and relations from RST to argumentative trees. They define the following four tasks to predict the constituents and relations in the argumentative tree:

- *attachment*: Given a pair of EDUs, are they connected?
- *central claim*: Given an EDU, is it the central claim of the text?
- *role*: Given an EDU, is it in the proponent's or the opponent's voice?
- *function*: Given an EDU, what is its argumentative function?[29]

They compare three approaches: i) a simple heuristic tree-transformation, which they consider as baseline, in which RST relations are mapped to argumentative functions based on the frequency in the respective edge alignments observed in (Stede et al., 2016), ii) an aligner based on matching sub-graphs in the RST structure to sub-graphs of the argumentative structure,[30] and iii) an evidence-graph

[28] Venant et al. (2013) compare RST and SDRT and explore transformations of representations from one formalism to the other.

[29] One of the fine-grained relations considered in the scheme: *support*, *example*, *rebut*, *undercut*, *link*, to represent *linked supports*, and the *join* non-argumentative meta-relation.

[30] A probability-weighted graph is generated from the RST tree -based on probabilities learned

model[31] (Peldszus and Stede, 2015b). While the baseline tree-transformation model yields the better performance in the prediction of the argumentative *role*,[32] for the three other tasks the evidence-graph model is the one for which best results are obtained.

Peldszus and Stede (2016) conduct the experiments with gold RST annotations. They consider that their results encourage investigating the use of discourse-level relations obtained by means of discourse parsers to predict the argumentative structure of texts. This is addressed by Hewett et al. (2019), where the authors apply RST and PDTB parsers to documents in the microtext corpus. They first perform a qualitative analysis of correlations between the predicted discourse relations and the annotations in the argumentative layer and, in a second step, they investigate whether automatically-obtained discourse features can contribute to improve the predictions of the evidence-graph argument mining model.[33] They find that, in spite of the fact that they observe a low level of alignment between PDTB and argumentative relations, features derived from the output of the PDTB parser contribute more than those obtained by means of the RST parser to improve the performance of argument mining classifiers. In particular, this is the case for the *attachment* and *function* tasks. As they indicate, additional work would be necessary to fully understand the contribution of the parsers' output. In an evaluation of the RST parser against the gold RST annotations in the microtext corpus they report a low performance of the RST parser.[34] This can explain in part the poorer contribution of the RST predictions to the argument mining task.

The work by Peldszus and Stede (2016) is particularly relevant in the context of our research, as it directly inspired our proposal to leverage RST annotations for argument mining. In our pilot experiment (Accuosto and Saggion, 2019b)[35] we do not use features obtained as the result of a full-fledged discourse parsing task,

in the training phase with a sub-graph alignment algorithm- and then a the minimum spanning tree (MST) algorithm is used to generate the argumentative graph.

[31] In this case a classifier is first trained for each of the four tasks and, in order to predict the edges of the argumentative tree, the predictions of the four classifiers are combined.

[32] Which the authors consider as expected, since the sequence of contrastive relations in the RST tree highly correlates to the sequence of proponent and opponent role assignments in the argumentative tree

[33] In these experiments they use an improved version of the evidence-graph model described in (Afantenos et al., 2018).

[34] F_1 scores of 0.338, 0.264, and 0.115 for *span*, *nuclearity*, and *relation* tasks, respectively

[35] Summarized in Appendix A.

but experiment with contextualized word embeddings pre-trained with RST annotations and find that they do contribute to improve the prediction of the argumentative structure of scientific abstracts. The results of these experiments motivate part of the work described in Chapter 4.

Another line of research explores the possibility of establishing mappings between argumentation schemes such as those propose by Walton et al. (2008) and discourse relations. Cabrio et al. (2013) explore to what extent a subset of Walton's schemes[36] can be mapped to categories of discourse relations used in PDTB. They do this by comparing the definitions of the argumentation schemes and the definitions of the discourse relations to find candidate mappings, which are then evaluated on examples extracted from PDTB. From the analysis of the discourse relations considered in PDTB they find that in some cases no clear mappings can be established with existing schemes. The authors suggest the possibility of introducing two new schemes to deal with these cases. Musi et al. (2018), in turn, propose to extend the microtexts corpus with a new annotation layer based on schemes derived from the *Argumentum Model of Topics* (Rigotti and Morasso, 2010). They conduct a pilot annotation of 40 microtexts, which they use to analyze correspondences between the new schemes-based annotations and the corpus RST annotation layer.

2.3 Mining arguments in scientific text

There is an important body of work on the identification of the *rhetorical components* of scientific texts.[37] Few initiatives, nevertheless, are aimed at the computational analysis of scientific articles from an argument mining perspective (Al Khatib et al., 2021).[38] The reason for this can lie in the fact that the annotation of arguments in scientific texts has proven to be particularly difficult due to the complexity and ambiguity of the scientific discourse Stab et al. (2014); Lauscher et al. (2018b); Green (2015).

[36] In this work five schemes are considered: *Argument from Example*, *Argument from Cause to Effect*, *Argument from Effect to Cause*, *Practical Reasoning*, and *Argument from Inconsistency*.

[37] See Section 2.4

[38] Stede and Schneider (2018) also say that scientific papers constitute a textual genre that *"somewhat surprisingly, so far has not received very much attention in the argumentation mining community."* (Stede and Schneider, 2018, p. 150)

In a recent survey on the subject, Al Khatib et al. (2021) consider a set of specific challenges faced when identifying arguments, their structure and relations in scientific texts, which include:

1. The lack of adequate argumentation models for the scientific discourse;
2. The specificities of the language used and the document structures employed in different scientific disciplines;
3. The multiplicity of document types in the scientific domain (including reviews, method papers, and experimental reports);
4. The extended use of enthymemes[39] in scientific arguments;
5. The fact that multiple valid interpretations of the structure of arguments is particularly challenging in the scientific domain (in particular, by non-domain-expert annotators);
6. The need to fulfil both the persuasive role and the presentation of objectivity which scientific writing demands gives origin to texts structurally complex (e.g., the distance between a claim and its premise may be particularly wide in scientific discourse).

As mentioned, one of the goals of this thesis is to address the first issue. In Chapters 3 and 6 we confirm some of the other challenges–in particular, points 2, 5, and 6–when we analyze our own annotations.

The work by Kirschner et al. (2015a), who annotates the introduction and conclusion sections of 24 German scientific articles in the educational domain, is one of the first works intended for the analysis of the argumentative structure of scientific texts (considering not only argumentative *components* but also how they are *linked* to each other). In this work, argumentative units are considered at the sentence level and they are linked by four types of relations, three directed: *support*, *attack*, *detail*, and one undirected: *sequence*. Fig. 2.1 shows an argumentative graph resulting from an annotation included in (Kirschner et al., 2015a).

The *support* and *attack* relations are adopted from the annotation scheme proposed by Peldszus and Stede (2013), based on Freeman's proponent/opponent model (Freeman, 2011).

[39] An implicit (unstated) premise or conclusion.

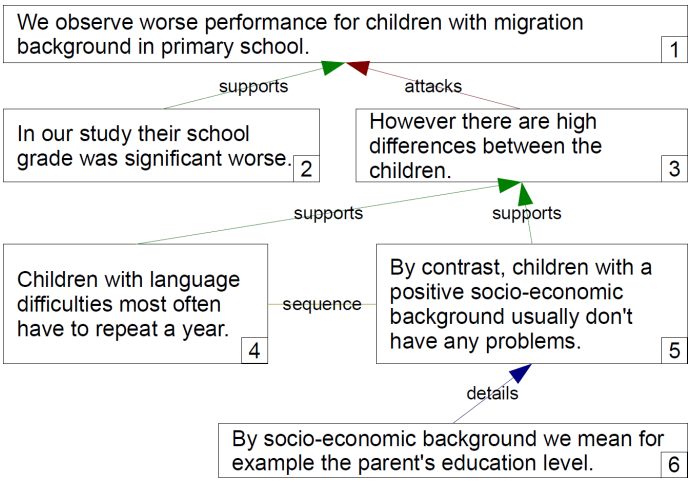


Figure 2.1: Example of an argument with six components annotated with the scheme proposed by Kirschner et al. (2015a). Source: (Kirschner et al., 2015a)

The *sequence* relation is introduced to link together sentences that could be considered in combination to form an greater argumentative unit, while *detail* subsumes various discourse relations, including non-explicit types of support (such as *background* and *elaboration* in discourse analysis). In order to simplify the analysis, only relations between sentences separated by at most 5 other sentences are considered. When they compute the inter-annotator agreement obtained with the annotation of relation types without considering direction, they observe low to moderate pair-wise Hubert's κ[40] values, which lie between 0.25 and 0.47.[41]

The work by Lauscher et al. (2018b) is one of the few other examples to address the analysis of argumentation in scientific texts. In this work, they enrich the *DrInventor Scientific Corpus* (Fisas et al., 2016) with an argumentation layer. The DrInventor corpus contains 40 computer graphics articles with four annotation layers, including *citation contexts*, *rhetorical role* of sentences, potential *relevance* of sentences for their inclusion in a summary, and indication of *subjective information* (e.g., novelty/advantages). Lauscher et al. (2018b) annotate argumentative components of arbitrary lengths with three type of units: *own claim*, *background claim*, and *data* (evidence for a claim) and three directed relations:

[40]Hubert's extension to Cohen's κ (Hubert, 1977) is used.

[41]In the paper a weighted average of $\kappa = 0.3912$ is reported. The averages are weighted considering the probability of the relations between two components, based on the distance between them.

supports, *contradicts*, and *semantically same* (to link together components that communicate the same information). Four annotators were involved in the annotation of the argumentative layer of the corpus, which was conducted in five iterations. They compute the resulting inter-annotator agreements in terms of F_1 scores. For a *relaxed* version of the agreement–in which components only have to match in type and overlap in span–in the last annotation iteration they obtain approximately 0.72 F_1 score for component types and 0.48 for relations.[42] They perform a normalized mutual information (NMI) analysis (Strehl and Ghosh, 2002) of the information shared by the rhetorical and argumentative layers, in which they find correspondences between segments annotated as *own claim* in the argumentative layer and as *approach* or *outcome* in the rhetorical layer, and between segments annotated as *background claim* and as *background* or *challenge* in the argumentative and rhetorical layers, respectively. In a follow-up work, Lauscher et al. (2018a) use the annotated corpus to train a model for the automatic identification of claims and evidence in scientific texts. For token-level classification models trained with the annotated corpus and evaluated in a test set of 12 randomly selected publications, they obtain a macro-averaged F_1 score of 43.8 in the identification of argument components.

Green (2016), in turn, proposes to parse arguments in biomedical texts by means of manually-constructed rules derived from argumentation schemes and implemented in a logic programming language such as Prolog. She includes a set of rules to exemplify the proposed approach. Some of them include domain-specific predicates such as *have-phenotype*, *have-genotype*, and *have-protein* and would require a previous text-mining stage. Other predicates, such as *similar* or *cause*, would exploit domain knowledge contained, for instance, in knowledge bases. The idea of identifying arguments by means of a combination of domain-specific knowledge and inference rules has also been proposed by Saint-Dizier (2018). In this case, linguistic and domain knowledge would be encoded by means of Qualia structures in the *Generative Lexicon* framework (Pustejovsky, 1998). These rule-based approaches have not been implemented and evaluated beyond pilot studies.

[42]The results are presented only graphically so the numbers are approximations.

2.4 Computational analysis of scientific discourse

2.4.1 Argumentative Zoning

One of the most influential initiatives in the annotation and automatic processing of rhetorical components of scientific articles is the *Argumentative Zoning* (AZ) model (Teufel et al., 1999). The AZ scheme includes annotations that characterize *knowledge claims* included in the papers according to whether they are introduced by the authors of the paper (*Own*), other researchers (*Other*) or they are part of accepted background knowledge (*Background*). Other AZ categories are aimed at establishing connections with previous work, and, in that sense, play a role similar to that of citation functions: *Basis* is used to refer to work that the current work is based on, and *Contrast* to refer to problems/weaknesses of previous work that the current work addresses. There is one category, *Aim*, intended to identify the main *knowledge claim* of the paper, and another one, *Textual*, reserved to annotate presentational information (such as the structure of the paper).

Originally developed and tested with a corpus of computational linguistics articles, the AZ scheme was later extended and used to annotate a corpus of 61 chemistry papers (Teufel et al., 2009). The updated scheme, AZ-II, contains 15 categories that result from further specifying the AZ categories *Other*, *Basis*, *Contrast*, and *Own*, adding two new categories to indicate advantages of a new *knowledge claim* (*NovAdv*), and description of limitations of the described proposal and/or future work (*Fut*).

The annotation units in AZ are sentences. Teufel (2010) explains that this was not an obvious decision. She also considered the possibility of doing the annotations at the clause level. She justifies her final choice based on the fact that, while clause-level units could potentially lead to more accurate annotations, *"this effect would be restricted to the rare cases where a sentence does contain more than one move."* Teufel argues that this *"has to be weighted against the much larger number of cases for one move or segment covers the sentence."* (Teufel, 2010, p. 199)

The AZ model was originally aimed at the annotation of full articles. The AZ scheme was developed having in mind automatic summarization and citation indexing (Teufel, 2010). The scheme has, nevertheless, been used in other applications. For instance, in (Feltrim et al., 2006), the AZ model is adapted for the automatic annotation of scientific abstracts in Portuguese (AZPort). The AZPort model is integrated as a module of SciPo,[43] a web-based tool aimed at supporting novice writers of academic texts: given an abstract, the system classifies its sentences by means of AZPort and, based on a set of rules for well-formed rhetorical structures, it provides feedback for potential improvements (e.g., re-ordering the elements of the text or adding missing content). Vargas-Campos and Alva-Manchego (2016), in turn, adapt the AZPort model to Spanish (AZEsp), which is also integrated into a computer-assisted writing tool for computer science dissertations in Spanish (Sci-Esp).

2.4.2 Core Scientific Concepts

The *Core Scientific Concepts* (CoreSC) annotation scheme adopts the view of a scholarly paper as a readable representation of a scientific research process by associating research components to sentences describing them (Liakata et al., 2012). It is, therefore, also a sentence-based annotation scheme. CoreSC contains 11 categories: *Hypothesis*, *Motivation*, *Goal*, *Object*, *Background*, *Method*, *Experiment*, *Model*, *Observation*, *Result*, and *Conclusion*. In turn, some of these categories[44] can be further qualified, giving origin to sub-categories (e.g., *Method-New-Advantage*).

The CoreSC scheme was used originally to annotate 265 papers in physical chemistry and biochemistry. The main application foreseen in its development was to use it to train machine learning models targeted at extractive summarization and *"intelligent querying"* of a repository of research articles (Liakata et al., 2010).

While CoreSC was initially intended to annotate each sentence with a single category, it has then been adapted to allow the annotation of multiple labels when more than one concept is expressed within the same sentence (Ravenscroft et al., 2016). The new scheme was applied to the annotation of the Multi-CoreSC CRA corpus,

[43] `nilc.icmc.usp.br/scipo/`
[44] In particular *Method* and *Object*

which contains 50 research papers on cancer risk assessment (CRA). Even when multiple concept annotations are allowed in the updated scheme, the annotation unit is still the sentence. In a similar line to the arguments of Teufel (2010), the authors justify the decision based on the lack of reliable clause-recognition systems, which, they understand could *introduce noise to the task of automatically identifying CoreSC concepts* ((Ravenscroft et al., 2016, p. 4116)).

Differences and similarities between AZ and CoreSC are studied in depth in (Liakata et al., 2010), where correlations between both annotation schemes are considered. Based on their analysis, the authors conclude that the two schemes are complementary in their approach, and that a combined scheme could improve the quality of automatically generated summaries of the papers, since they could better reflect both the research process outcomes as well as the knowledge claims and author attributions contained in them.

2.4.3 Claim Framework

Blake (2010) proposes the *Claim Framework* for the characterization of *claims* in biomedical articles. In this context, a *claim* captures how the *basis* of a *change* in a relation between *two concepts* is reported.[45] *Claims* are categorized as *explicit claims*, *observations*, *correlations*, *comparisons*, and *implicit claims*. While the annotation is also done at the sentence level, annotators can associate more than one *claim category* to each sentence.

Blake (2010) provides the following example of a sentence with two *explicit claims*, which are marked with square brackets:

> Indeed, [*glycine* **prevented** *Wy-14643-stimulated superoxide production*] by Kupffer cells.
>
> Indeed, glycine prevented [*Wy-14643-stimulated superoxide* **production** by *Kupffer cells*].

In the first case, *glycine* and *Wy-14643-stimulated superoxide production* are the concepts involved in the relation and **prevented** the reported change, while, in the second case, *Kupffer cells* and *Wy-14643-stimulated superoxide* are the concepts involved in the relation and **production** the reported change.

[45] These four elements are referred to *facets* in the framework.

Being *explicit claim* the type most frequently found in the studied domain, the author develops a system to identify this type of *claims* by means of a dependency grammar with semantic and syntactic constrains, which is tested on an annotated corpus of 29 articles. This work is extended in (Park and Blake, 2012) to identify sentences containing *claims* of type *comparison*. In this case, they use a corpus of 122 annotated documents and the classification of the sentences is done by means of a Bayesian Network.

Ahmed et al. (2013) explore the applicability of the Claim Framework to social science literature. They conduct a pilot study in the areas of community informatics (CI) and information and communication technologies for development (ICT4D) and compare how *claims* are reported in these fields with respect to the biomedical domain in which the framework was first tested. They analyze *claims* made in different sections of the papers and find various differences between both disciplines. For instance, they observe that there is a greater proportion of *claim* sentences in the abstract section of biomedical research papers when compared to those found in CI or ICT4D.

2.5 A note on transfer learning

In our experiments we implement *transfer learning* approaches, including *multi-task learning* (Caruana, 1997) and *sequential transfer learning* (pre-fine-tuning parameters of attention-based models on an intermediate related task). With the increased popularity of parameter-rich models, intermediate fine-tuning has become an active research area (Phang et al., 2018). Recent works have focused on trying to clarify when and why it works–including (Pruksachatkun et al., 2020)–as well as on how to select the task(s) to use for intermediate training–including (Park and Caragea, 2020; Poth et al., 2021).

Transfer learning is not one of our research goals. An in-depth review of work in this area is, therefore, beyond the scope of this thesis. Multiple surveys on the topic are available, including (Pan and Yang, 2010; Weiss et al., 2016; Tan et al., 2018; Zhuang et al., 2020), as well as books, including (Azunre, 2021; Yang et al., 2020). In his PhD thesis, Ruder (2019) provides an in-depth survey of the application of transfer learning to natural language processing.

Chapter 3

THE SCIARG CORPUS OF SCIENTIFIC ABSTRACTS

In this chapter we propose SciARG, a new annotation scheme particularly tailored at the identification of argumentative units and relations in scientific abstracts. We apply this scheme to augment, with an argumentation layer, a subset of the *Discourse Dependency TreeBank for Scientific Abstracts* (SciDTB) (Yang and Li, 2018), which contains annotations with discourse relations between at elementary-discourse units in computational linguistics abstracts. We refer to the augmented corpus as SciARG-CL.

This chapter is organized as follows:

- In Section 3.1 we describe the data used in our annotations.
- In Section 3.2 we describe our annotation scheme, including argumentative *unit types* and *relations*, and motivate our decision to consider sentences as annotation units.
- In Section 3.3 we describe the annotation process and show an example of the annotation interface.
- In Section 3.4 we report and analyze the inter-annotator agreement obtained.
- In Section 3.5 we provide corpus statistics and analyze correspondences between the different annotations considered (i.e., types and relations).
- In Section 3.6 we summarize the main contributions of this chapter.

3.1 Data

In this work we enrich the SciDTB corpus[1] with an annotation layer that describes the argumentative structure of scientific abstracts. In preliminary experiments, described in (Accuosto and Saggion, 2019b,a, 2020), we explore the possibility of leveraging existing discourse annotations for the identification of argumentative components and relations in scientific abstracts by annotating 60 SciDTB abstracts with and argumentation-level of annotations. These experiments and its results are summarized in Appendix A. The extended corpus presented in this chapter includes the 225 abstracts available in the Proceedings of the EMNLP 2014 Conference[2].

3.1.1 The SciDTB corpus

SciDTB contains 798 abstracts from the ACL Anthology[3] annotated with discourse relations based on an adaptation–to the scientific domain–of the Rhetorical Structure Theory (RST) framework (Mann et al., 1992). As mentioned, RST provides a set of coherence relations with which adjacent spans in a text can be linked together in a discourse analysis, resulting in a tree structure that covers the whole text. The minimal units that are joined together in RST are called *elementary discourse units* (EDUs). The SciDTB annotations use 17 coarse-grained relation types and 26 fine-grained relations (Table 3.1).

The segmentation of sentences into discourse units in SciDTB was performed in a semi-automated way, with a first automatic segmentation done by means of the SPADE discourse parser (Soricut and Marcu, 2003), which was then manually checked. The resulting EDUs were then labeled in order to construct the discourse dependency trees, which were made available in JSON format.[4] Poly-nary discourse relations in RST are binarized in SciDTB by applying a *"right-heavy"* transformation used in other works that represent discourse structures as dependency trees (Morey et al., 2017; Stede et al., 2016; Li et al., 2014).

[1] Section 3.1.1

[2] 2014 Conference on Empirical Methods in Natural Language Processing, `emnlp2014.org`

[3] `aclanthology.org`

[4] The SciDTB corpus is available at `github.com/PKU-TANGENT/SciDTB`.

Coarse-grained	Fine-grained
ROOT	*ROOT*
Attribution	*Attribution*
Background	*Related, Goal, General*
Cause-effect	*Cause, Result*
Comparison	*Comparison*
Condition	*Condition*
Contrast	*Contrast*
Elaboration	*Addition, Aspect, Process-step, Definition, Enumerate, Example*
Enablement	*Enablement*
Evaluation	*Evaluation*
Explain	*Evidence, Reason*
Joint	*Joint*
Manner-means	*Manner-means*
Progression	*Progression*
Same-unit	*Same-unit*
Summary	*Summary*
Temporal	*Temporal*

Table 3.1: Fine and coarse-grained relations used in the SciDTB corpus

Let us consider the following example from (Zhang and Wang, 2014), included in the SciDTB corpus, in which EDUs are numbered and identified by square brackets. Fig. 3.1 shows the partial discourse tree in SciDTB that includes EDUs 1-4.

> [State-of-art systems for grammar error correction often correct errors]$_1$[based on word sequences or phrases.]$_2$[In this paper, we describe a grammar error correction system]$_3$[which corrects grammatical errors at tree level directly.]$_4$

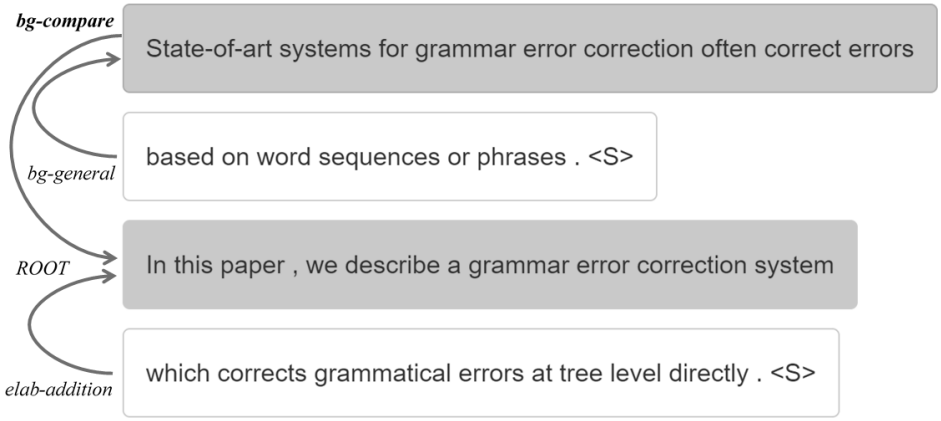


Figure 3.1: Discourse relations in SciDTB.

3.2 The SciARG annotation scheme

One of the main goals of our work is to facilitate argumentation-based analyses of scientific abstracts by identifying the way in which authors structure its contents to persuade potential readers about the relevance and/or validity of their proposals. For this we propose the *SciARG* annotation scheme, which aims at making explicit the underlying argumentative structure of the scientific abstracts by i) identifying the *rhetorical role(s)* of annotation units, as well as ii) *discourse relations* between them–in particular, those relevant from an *argumentative* point of view.

For the proposed scheme we adopt a pragmatic perspective that intends to bridge an existing gap between different annotation levels.

In particular:

- *Discourse-level annotation schemes*, such as those in the RST framework (Mann and Thompson, 1988);
- *Rhetorical-level annotation schemes*, such as Argumentative Zoning (AZ) (Teufel et al., 1999) and the CoreSC schemes (Liakata et al., 2010);
- *Argumentation mining annotation schemes*, such as the one proposed in Peldszus and Stede (2013)).

3.2.1 Annotation level

When analyzing the argumentative structure of texts, a key consideration is to define the boundaries of the argumentative units, which can, in theory, range from a few words to several sentences. In our preliminary studies (Accuosto and Saggion, 2019b,a, 2020), we considered *elementary discourse units* (EDU) as the minimal annotation unit and modeled all the tasks as token-classification problems. Analyzing these annotations we observed that unit boundaries coincided with sentence boundaries 93% of the times. Models trained with such annotations would therefore tend to predict sentences as annotation spans, achieving a high accuracy in the prediction of the boundaries, but not being very useful for the prediction of the more difficult cases, where argumentative spans are shorter than a sentence. In addition, if we expect units to be sentences in a high percentage of the instances, modeling the problem as a token-level classification task is not an optimal solution, as we transform the problem into a more difficult one in cases in which it is not required.

Based on the above considerations, we decide to define the *sentence* as the annotation unit in the more general case, and to tackle as separate problems the classification of *rhetorically/argumentatively complex* sentences–those containing more than one argumentative unit–and the identification of argumentative units within them. This is described in detail in Chapter 5. The decision of whether it is necessary to consider this second step of analysis in an argumentative mining pipeline or not depend on the granularity level of argumentative components needed in a specific downstream application.[5]

The consideration of rhetorical or argumentative units at the sentence level is an approach that has been frequently adopted in argument mining works, including, in the scientific domain (Teufel et al., 1999; Liakata et al., 2012; Kirschner et al., 2015a). Considering annotations at the sentence level not only facilitates the annotation process, it contributes to model the prediction of argumentative types and relations in a way that is more natural in the majority of the cases. This, as we see in Chapter 5, can have an impact on the decision of the classification methods that are better suited for argument mining tasks in each context.

[5]For instance, in the application described in Part II, we use predictions at the sentence level.

3.2.2 Types of units

Argumentation mining initiatives across multiple domains frequently divide argumentative components into two (non-mutually exclusive) types: *claims* and *premises*. The main tasks in this case are i) to identify which spans of texts can be associated to one or both of these types,[6] and ii) to establish links between *premises* and the *claims* that they *support* or *attack*, as seen in Chapter 2. In the case of scientific discourse, however, it is frequent to find that *claims* are not explicitly stated in an argumentative writing style but are instead left implicit or presupposed (Al Khatib et al., 2021).

The description of the work addressed in a given paper, for instance, conveys implicit *claims* in relation to the *relevance* of the problem at stake–which can be supported by stating existing unsolved problems and/or limitations in current solutions (*motivation*), or with respect to the *benefits* of the proposed approach–which can be supported by explaining the improvements obtained with the new approach (*results*). Works such as AZ and CoreSC go in-depth into the identification of the diverse roles[7] that different components of the scientific discourse can play, with an important difference with respect to the general argumentative mining perspective: they put the focus on the characteristics of the constituents and/or their functions, but not on the *structure* of the text (i.e., the *relations* between them).[8]

Several works have been dedicated to the study of the *rhetorical organization* of scientific texts and their parts, including (Swales, 1990) and (Maeda, 1981). Focusing in particular on the papers' abstracts, different authors consider different *preferred* or *most frequent* sequences of *rhetorical moves*, depending on the specific scientific discipline they study.

Dos Santos (1996), for instance, analyzes the textual organization of *applied linguistics* abstracts, Abdollahpour and Gholami (2018) study abstracts included in *medical science* databases, Doró (2013) considers articles published in *English studies* journals, Cross and Oppenheim (2006) conduct a genre analysis of *biol-*

[6] In some works parts of the text that are considered to be uninformative from an argumentative perspective are previously filtered out.

[7] The *knowledge claim* role of sentences, in the case of AZ, and in terms of the *type of information* they convey, in the case of CoreSC.

[8] With the exception of shared identifiers that link instances of the same concept together.

ogy abstracts, and Hartley and Betts (2009) explore common *weaknesses* of *social science* abstracts, and compare structured *vs.* non-structured abstracts from this perspective. (Orasan, 2001), in turn, studies scientific abstracts across six scientific disciplines, considering *lexical*, *syntactic* and *discourse* patterns. Closer to our work, Feltrim et al. (2003) use the AZ scheme for the analysis of writing patterns in *computer science* abstracts in Portuguese, with the ultimate goal of supporting the development of assisted-writing tools. Finally, Dayrell et al. (2012) develop the MAZEA corpus for the identification of rhetorical moves in abstracts in two disciplines: *physical sciences and engineering* and *life and health sciences*, as we see in more detail in Chapter 5.

Even with the variations observed in the different disciplines analyzed, a broad categorization of the most frequent rhetorical moves in scientific abstracts can be considered:

- Contextualize the *research topic*;
- Expose *limitations* in existing solutions;
- Explain new proposed *approaches*;
- Describe the *methodologies* used;
- Summarize the main *results*;
- Draw *conclusions*.

Based on this general categorization of the different types of constituents that can be found in scientific abstracts, we propose a fine-grained annotation scheme that contains ten types of unit (Table 3.2).

Each type can, in turn, be mapped to a coarse-grained category. The use of fine or coarse-grained types depend on the needs of the specific application in which the annotations are to be used.[9]

Units identified with the described types can be seen as playing different argumentative roles–either as *claims* and/or *premises*–when considered in relation to other units. Table 3.3 shows an example for each type of unit extracted from the annotated corpus.

[9] In the context of this work we use the fine-grained types.

Fine-grained type / Description	**Coarse**	**Argumentative**
proposal High level description of the proposed approach/solution	*proposal*	*claim*
proposal-implementation Processes/tools/methods that are part of the proposal	*proposal*	*claim/premise*
observation Data obtained from experiments	*outcomes*	*premise*
result Direct interpretation of observed data	*outcomes*	*claim/premise*
conclusion High-level interpretation/generalization of results	*outcomes*	*claim/premise*
means Secondary methods/processes not part of the proposal	*methods*	*premise*
motivation-problem Known problem/limitation addressed by the proposal	*motivation*	*claim/premise*
motivation-hypothesis New ideas/paths for known problems/limitations	*motivation*	*claim/premise*
motivation-background Known information to support the proposed approach	*motivation*	*premise*
information-additional Additional information (definitions/examples)	*other*	*premise*

Table 3.2: Fine and coarse-grained types of units with most frequent argumentative roles.

proposal
We present a novel approach to improve word alignment for statistical machine translation (SMT).
proposal-implementation
We observe, identify, and detect naturally occurring signals of interestingness in click transitions on the Web between source and target documents, which we collect from commercial Web browser logs.
observation
Our method produces a gain of +1.68 BLEU on NIST OpenMT04 for the phrase-based system, and a gain of +1.28 BLEU on NIST OpenMT06 for the hierarchical phrase-based system.
result
Experimental results show statistically significant improvements of BLEU score in both cases over the baseline systems.
means
We conducted experiments on two standard benchmarks: Chinese PropBank and English PropBank.
conclusion
This transfer learning approach brings a clear performance gain over features based on the traditional bag-of-visual-word approach.
motivation-problem
However, fundamental problems on effectively incorporating the word embedding features within the framework of linear models remain.
motivation-hypothesis
Combining the two tasks can potentially improve the efficiency of the overall pipeline system and reduce error propagation.
motivation-background
Recent work has shown success in using continuous word embeddings learned from unlabeled data as features to improve supervised NLP systems, which is regarded as a simple semi-supervised learning mechanism.
information-additional
The structure of argumentation consists of several components (i.e. claims and premises) that are connected with argumentative relations.

Table 3.3: Examples for each type of unit in SciARG

In order to facilitate the annotation process and reduce the number of decisions that annotators had to make, we instructed them to annotate *all* of the sentences in the abstracts, understanding that specific annotations types could be ignored if they are considered to not be relevant for a particular downstream application.

Linked to the observation made in the pilot annotation experiment–in relation to the frequent coincidence between sentence and argumentative units boundaries–is the fact that most sentences contain only *one* type of unit. This is in line with findings in the MAZEA corpus, for which the authors indicate that *"the vast majority of sentences from English abstracts reflect one single rhetorical move"* (Dayrell et al., 2012, p. 1607). Similar results are obtained with CoreSC's revised schemes that allows multi-label annotations: although in this case sentences in the whole text of the papers are annotated–which are expected to be more complex than sentences in abstracts, *"only 3.25% of sentences in the consensus have more than one annotation"* (Ravenscroft et al., 2016, p. 4117).

Similar observations are made by Stab (2017) with respect to his corpus, which contains a total of 7,116 sentences annotated at clause level: *"There are no sentences that include more than two argument components. [...] In total, there are 583 sentences that include several argument components of which 302 sentences include two argument components of a different type, e.g. a claim followed by a premise. Therefore, 8.2% of all sentences need to be split in order to identify argument components. This shows that classifying sentences as a whole is not sufficient for identifying argument components"* (Stab, 2017, pp. 57-58).

Even when we differ in the way to approach the problem–as we argue that the differences in proportion between the different types of sentences should be taken into consideration–we agree with Stab (2017) in relation to the fact that when multiple rhetorical roles/argumentative components can be identified within a sentence, this can provide significant information to interpret it–for instance, when analyzing the *cogency* of the whole argumentation.

In the first iteration of the annotation process, which is described in Section 3.3, we observed that sentences containing mentions to the *means* by which *results* were obtained where the most frequent cases among those in which more than one type could be identified.[10] In fact, this was corroborated in the final annotations,

[10]The ways in which *results* and *means* are included in the sentences are also syntactically more diverse than for other combination of types–which are mostly introduced by coordinated clauses.

where more than one type was identified in 11% of all the sentences. Of those, 53% contained information both about *results* and the *means* by which they were obtained.[11]

Consider, for instance, the following sentence from the abstract of (Zhang et al., 2014):[12]

> [*Experiments using the IWSLT 2010 dataset* show]$_{means}$ [that *the system achieves BLEU comparable to the state-of-the-art syntactic SMT systems*]$_{result}$

In this case we can identify a complement clause that reports the obtained *results*, while the subject of the sentence indicates the *means* through which those results were obtained.

Based on these observations, and with the goal of reaching a good balance between facilitating the annotation process and making sure that no essential information was missed in the annotations, we adopted the following criteria:[13]

i. We introduce the type *result-means* to annotate sentences where both *results* and the *means* by which they were obtained can be identified.
ii. We allow the annotation of a *secondary type*, in addition to the *main type* of the units.

Annotators were asked to weight the relevance of the different types of information contained in the sentence to make a decision with respect to the selection of *main* and *secondary* types.

It was possible, therefore, for annotators to identify at most three types of units in a sentence –in the case where one of the annotated types was *result-means*. This occurred very rarely: only in 6 out of 1,199 annotated sentences (0.5% of the annotations).

[11] These percentages corresponds to the *consensus annotations* described in Chapter 4.

[12] The square brackets correspond to the discourse-level segmentation.

[13] The guidelines used in the annotation process are available at: `github.com/LaSTUS-TALN-UPF/SciARG/blob/main/Annotation_Guidelines_Arguments_SciDTB.pdf`.

3.2.3 Relations

Our second annotation level includes directed labeled *relations* that link pairs of sentences together, thus forming a directed graph with sentences as its nodes and the relations between them as the edges. In order to gain in uniformity in the annotations, reduce the level of ambiguity and simplify the annotation process, we only consider *trees* as valid annotations (acyclic graphs where each node, with the exception of the *root* node, has one and only one parent).

As explained in Section 3.2, our proposed annotation scheme is intended at exposing the *argumentative structure* of the abstracts. We are interested, in particular, in identifying relations insofar they provide information that is relevant to analyze argumentative dimensions of the texts. We intend our scheme to capture information that can contribute, for instance, to analyze the *clarity* of the abstracts, as well as the *local sufficiency* of the given premises in relation to the–explicit or implicit–claims included in them.[14] This goal delimits the repertoire of relations to consider. It would not be practical, for instance, to include in our scheme the whole set of fine-grained relations considered in different discourse analysis frameworks.

Table 3.4 shows the six types of relations that we include in our scheme. Depending on the particular dimensions of arguments that are of relevance in a downstream application, different subsets of relations can be considered.

Relation	Description of the child node function
support	Provides new supporting information/evidence for the parent
elaboration	Provides additional information relevant to specify/contextualize the parent
by-means	Describes methods through which supporting evidence is obtained
info-required	Provides information essential to understand/contextualize the parent
info-optional	Provides non-essential information (e.g.: examples, definitions)
sequence	Describes a step that comes after the step described by the parent in a process

Table 3.4: Types of relations

[14] In line with the argumentative quality dimensions described in (Wachsmuth et al., 2017a).

Most argument mining initiatives consider the *logical* dimensions of argumentation, largely conveyed by relations of type *support* (or *attack*). We also consider the *support* relation to link *premises* to *claims* or to other *premises* in an argumentation chain. Fig. 3.2 shows three examples of the use of the *support* relation, between nodes (5) (*observation*) and (3) (*result*), between node (4) (*motivation-problem*) and (2) (*proposal*) and between nodes (3) (*result*) and (2) (*proposal*).

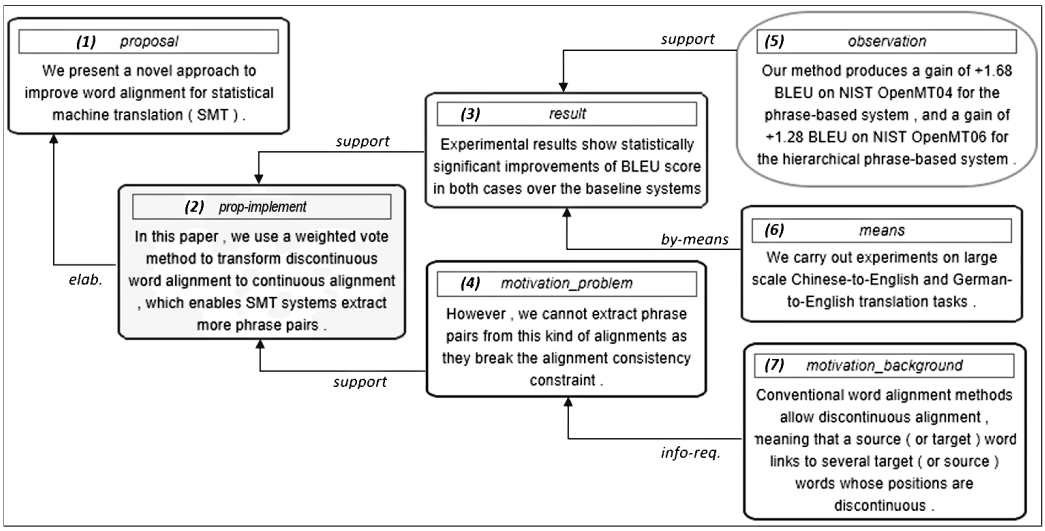


Figure 3.2: Example of argumentative tree. The main unit corresponds to the node with dark background.

In the same line as (Peldszus and Stede, 2013), where they consider *example* as a particular type of *support*, we further specify different types of *support* that are specifically tailored to the analysis of the scientific discourse. Should we consider only pure *support* relations, our annotations would not fully capture, for instance, the link between a *proposal* and its *implementation* details, which can contribute to persuade the reader about the relevance and/or validity of the proposal[15] or between a *result* and the *means* by which it was obtained, supporting the *credibility* and *acceptability* of the presented results.[16] We differentiate them by considering one relation type for each of these cases: *elaboration*[17] and *by-means*, respectively.

[15]In Fig. 3.2, the link between nodes (2) and (1).

[16]In Fig. 3.2, the link between nodes (6) and (3).

[17]Note that, in our scheme, the *elaboration* relation contains but has a broader meaning than the elaboration relation used in discourse analysis frameworks such as RST.

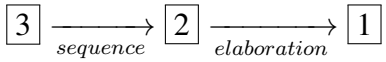
Considering single sentences as annotation units implies that relations have to be introduced to join groups of sentences that, if looked from a higher perspective, could be considered to be part of one argumentative block (i.e., the whole block could potentially be omitted, but no information conveyed by one sentence in the group can be omitted without affecting the coherence of the text, in contrast, for instance, to the *elaboration* relation, where leave nodes could be omitted). Analyzing the types of constructions present in our dataset, we identify two situations: i) groups of sentences that describe steps in a process and, ii) groups of sentences that authors use to build up supporting evidence for–implicit or explicit–claims. We use, respectively, the labels *sequence* and *info-required* for these types of relations.

The *info-required* relation is illustrated by the link between nodes (4) (*motivation-problem*) and (7) (*motivation-background*) in Fig. 3.2: it can be said that the *main* motivation for the proposed approach is given by the sentence that introduces the problem, in node (4), but this could not be clearly understood/contextualized without the background information provided by the sentence in node (7). When looking for supporting evidence for node (2), therefore, we would need to consider not only their direct children but also the chains of sentences below them linked by *info-required* relations.

For an example of the *sequence* relation, consider, for instance, the following sentences from the abstract of (Stab and Gurevych, 2014b):

> [We consider this task in two consecutive steps.]$_1$ [First, we identify the components of arguments using multi-class classification.]$_2$ [Second, we classify a pair of argument components as either support or non-support for identifying the structure of argumentative discourse.]$_3$

Sentences (2) and (3) describe a sequence of steps in a process and are therefore linked together by a *sequence* relation (in this case, the adopted criterion is that the direction of the links goes from the last to the first step in a sequence). In this case, the first step of the sequence–sentence (2)–is finally linked to sentence (1) by means of an *elaboration* relation.

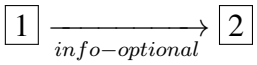


Note that the *sequence* relation is an exception in the sense that it does not convey any information that can contribute to assess argumentative dimensions of the text. In general, we refer to *argumentative functions* or *relations* but these terms should be understood as including also these non-argumentative relations.

As explained in Section 3.2.2, all of the sentences are labeled when applying the SciARG scheme. Information considered as less *essential* is annotated as *information-additional* and linked to its parent node by means of the *info-optional* relation. Consider, for instance, these two sentences from the abstract of (Persing and Ng, 2014):

> [A poll consists of a question and a set of predefined answers from which voters can select.]$_1$ [We present the new problem of vote prediction on comments, which involves determining which of these answers a voter selected given a comment she wrote after voting.]$_2$

Sentence (1) provides a definition that is useful to contextualize the proposal included in sentence (2),[18] but that is not *essential* to understand it. Therefore, a link from sentence (1) to sentence (2) can be established and labeled with the *info-optional* relation.



In contrast to many argument mining works, we do not include *attack* as a type of relation in our scheme. This is simply because we could not identify any real *attack* relation, neither in the annotations made in the pilot experiment nor in the final annotations.[19] Should it be necessary, our scheme could be extended with other types of relations without significant alterations to our experiments and analyses.

[18] Contributing, therefore, to improve the *clarity* of the text.

[19] In (Stab and Gurevych, 2014b) *attack* relations were also omitted in the experiments due to their low frequency, although they were kept in the annotation scheme.

3.2.4 Main unit

In addition to annotating unit types and relations between them, annotators were asked to identify, in each abstract, the unit that, by itself, best describes the most relevant contribution of the work. This corresponds to the concept of *main claim* in other textual genres. We denominate it the *main unit*. In Fig. 3.2, for instance, the *main unit* is (2).

3.3 Annotation process

We refer as SciARG-CL to the corpus obtained by annotating the 225 computational linguistics abstracts from SciDTB described in Section 3.1 with the SciARG scheme.

Three annotators participated in the annotation of the SciARG-CL corpus: two NLP researchers and one computational linguist. The annotation was done in three rounds, in the course of six months. The first two rounds were aimed at training the annotators, clarifying doubts and making the necessary adjustments to the annotation scheme and tool. The annotation was done by means of the GraPAT tool (Graph-based Potsdam Annotation Tool) (Sonntag and Stede, 2014), which was adapted to the specific needs of the task.

The annotation tool was originally tailored to the annotations made in the pilot experiments described in A, for which the annotation units considered were EDUs. This means that, even when the annotations are done at the sentence level in the final scheme, annotators were presented with the list of EDUs contained in the abstract, which were then combined to form the nodes of the argumentative tree, as shown in Fig. 3.3.

More details of the annotation process are provided in the annotation guidelines.[20]

As a result of the process, 225 CL abstracts were annotated, having 30 abstracts annotated by the three annotators to compute inter-annotator agreement (Table 3.5).

[20] Available at `github.com/LaSTUS-TALN-UPF/SciARG/blob/main/Annotation_Guidelines_Arguments_SciDTB.pdf`.

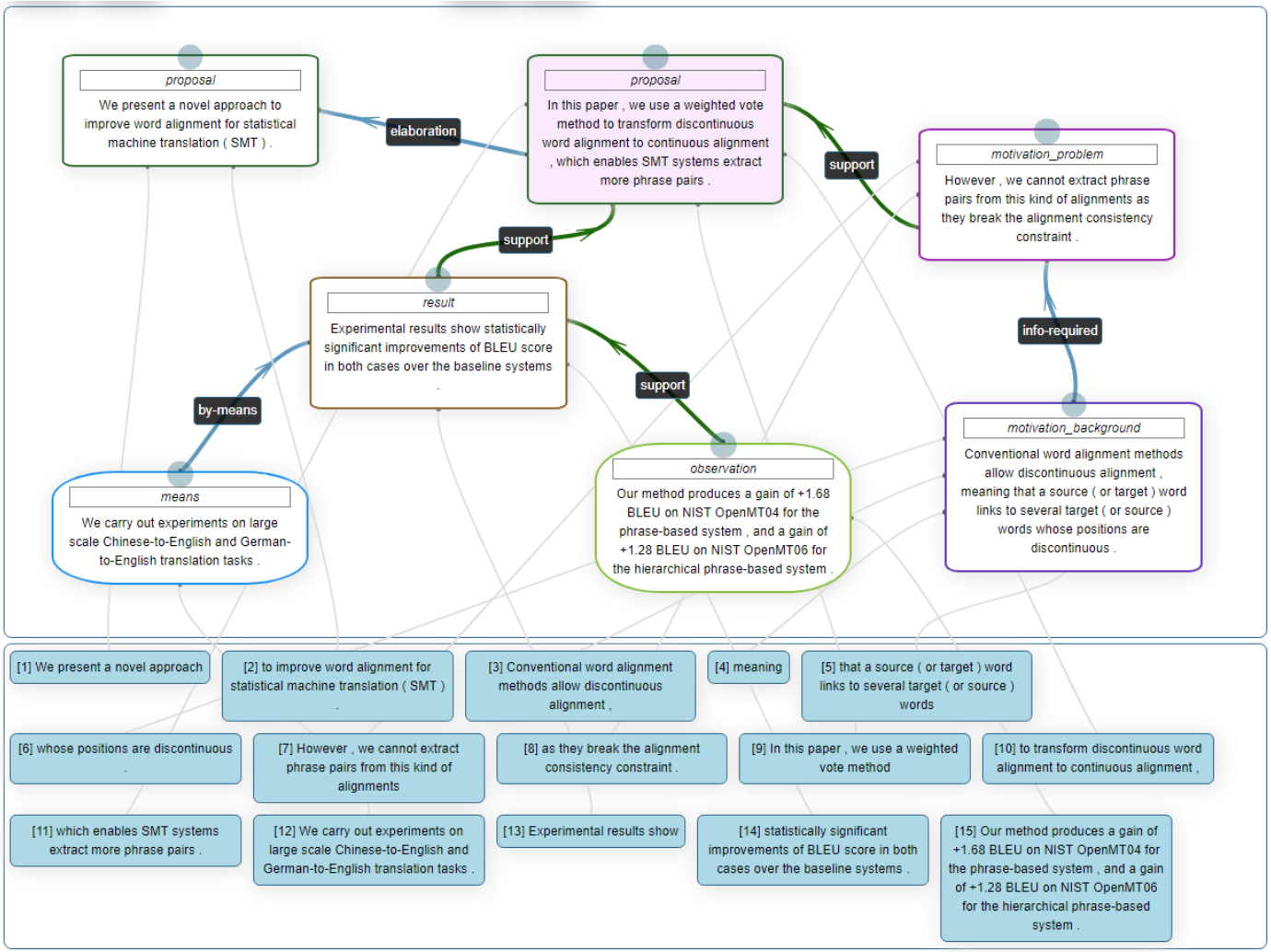


Figure 3.3: Annotation interface based on the GraPAT tool.

3.4 Agreement

In this section we assess the reliability of our annotations by considering inter-annotators' agreement, computed in the set of overlapping annotations.

Annotator	Individual	Overlapping	Total
ann_1	80	30	110
ann_2	77	30	107
ann_3	38	30	68
Total	195	30	225

Table 3.5: Number of individual and overlapping annotations by each annotator in the SciARG-CL corpus.

Table 3.6 shows the agreements obtained in terms of the average of the pair-wise Cohen's κs–with their corresponding standard deviations. In order to compute the agreements we consider pair-wise label matching for four tasks:

- *unit type*: one of the ten labels in Table 3.2;
- *parent position*: absolute position of the parent sentence in the document;
- *relation type*: one of the six labels in Table 3.4;
- *main unit*: two possible values: *main*/*secondary*.

In addition to each task-specific agreement, we report the agreement observed when considering simultaneous exact matches for all the tasks. Substantial agreement is achieved for all tasks (and almost perfect agreement when coarse-grained types are considered).

Cohen's κ coefficient is a standard measure of inter-annotator agreement and, as such, it makes it possible to compare the reliability of different annotation initiatives. It is relevant to note, nevertheless, that in our case the different types of annotations cannot be consider as completely independent from each other–as a decision made when annotating one node of the argumentative structure affects decisions made in other nodes.[21] This presents a limitation when interpreting the significance of Cohen's κ coefficient in these cases.

[21]This problem was already pointed out by Marcu et al. (1999) when evaluating inter-annotator agreement of discourse annotations.

Task	Cohen's κ
Fine-grained unit type	0.77 ($\sigma = 0.004$)
Coarse-grained unit type	0.94 ($\sigma = 0.016$)
Parent position	0.72 ($\sigma = 0.075$)
Relation type	0.79 ($\sigma = 0.026$)
Main unit	0.92 ($\sigma = 0.042$)
All combined	0.59 ($\sigma = 0.055$)

Table 3.6: Agreement in SciARG-CL. Average pairwise of Cohen's κ,

In Table 3.7 we report the average of pair-wise *accuracies* obtained for the different tasks, which provides a better intuition of the degree of agreement between pairs of annotators.

Task	Accuracy
Fine-grained unit type	0.81 ($\sigma = 0.004$)
Coarse-grained unit type	0.96 ($\sigma = 0.010$)
Parent position	0.77 ($\sigma = 0.062$)
Relation type	0.84 ($\sigma = 0.019$)
Main unit	0.97 ($\sigma = 0.013$)
All combined	0.61 ($\sigma = 0.053$)

Table 3.7: Agreement in SciARG-CL. Average of pairwise accuracy.

When considering a pair of annotated documents as labeled trees, the accuracy indicates the percentage number of changes (with respect to the number of nodes) that would be necessary to make in one tree to obtain the other one. It can, therefore, be interpreted as an edit-distance measure that allows to estimate the degree of agreement between two annotators when considering the document as a whole.

For instance, let us consider the two graphs in Fig. 3.4, representing two different annotations for the same three sentences, where the letters represent the type of unit (*P=proposal*, *R=result*, *C=conclusion*, *M=means*) and the subscript represents the position of the sentence in the text (we omit here the *main unit* annotation). In order to transform one graph into the other we would need to change four annotations, as shown in Table 3.8.

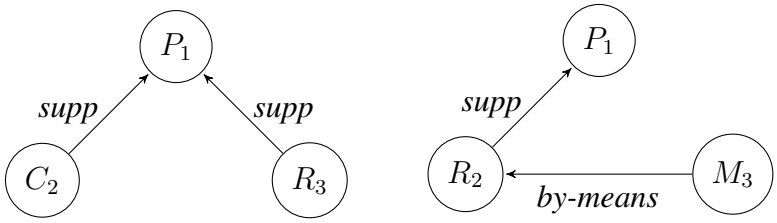


Figure 3.4: Two graphs representing two annotations for units' types, functions, and parents of the same three sentences.

Unit	**Type** ann_1	**Type** ann_2	**Correct**
1	*proposal*	*proposal*	1
2	*conclusion*	*result*	0
3	*result*	*means*	0
Unit	**Relation** ann_1	**Relation** ann_2	**Correct**
1	*none*	*none*	1
2	*support*	*support*	1
3	*support*	*by-means*	0
Unit	**Parent** ann_1	**Parent** ann_2	**Correct**
1	*0*	*0*	1
2	*1*	*1*	1
3	*1*	*2*	0

Table 3.8: Example of the computation of the global accuracy for the graphs shown in Fig. 3.4.

In this example, the accuracy for the unit types, relations and parents are, respectively: $acc_{type} = 0.33$, $acc_{func} = 0.67$ and $acc_{par} = 0.67$, and, when all the labels are considered, the global accuracy is $acc_{all} = 0.56$.

The evaluation of argument annotations is an open issue, as traditional agreement scores might not properly reflect the reliability of the annotations. Different annotations of the same text might reflect distinct interpretations of the authors' intentions and could, therefore, be considered as fully or partially correct. Kirschner et al. (2015a) propose a graph-based measure that captures similarity of meaning in annotations, while Stab et al. (2014) propose to explore, for the evaluation of argument annotations, methods that consider multiple correct possibilities, such as those used in text summarization.

3.5 Corpus analysis

3.5.1 Consensus annotations

For the 30 abstracts with overlapping annotations used to compute inter-annotator agreement (Table 3.5) we generate a *consensus* set of annotations by assigning, to each instance, the majority label. In the infrequent cases[22] in which there is total discrepancy among the three annotators we keep the label assigned by the annotator with the highest average of pair-wise agreement with the other two annotators for the specific type of label.

3.5.2 Corpus statistics

Table 3.9 shows the overall statistics of the SciARG-CL corpus considering the consensus annotations.

Statistics	
Number of abstracts	225
Total number of units	1,199
Avg. #units/abstract	5.3 (σ = 1.7)
Max. #units/abstract	13
Min. #units/abstract	2
Avg. #tokens/unit	24.4 (σ = 9.9)
Max. #tokens/unit	101
Min. #tokens/unit	5
Forward relations	32%
Backward relations	68%

Table 3.9: Statistics of SciARG-CL

The corpus contains 1,199 annotated sentences. The majority of them (46%) are annotated as *proposals* (either *proposal* or *proposal-implementation*), 26% correspond to *outcomes*, being *results* the most frequent type, while fewer abstracts include higher-level *conclusions* and/or explicit data labeled as *observations* (Table

[22]This occurs only in six cases.

3.2). A similar percentage (24%) corresponds to *motivation* units, being the units that provide *background* or state *problems* in existing solutions the overwhelming majority.

Main type	**Number**	**Percentage**
proposal	290	24%
proposal-implementation	260	22%
result	157	13%
result-means	70	6%
conclusion	50	4%
observation	40	3%
means	27	2%
motivation-background	159	13%
motivation-problem	102	9%
motivation-hypothesis	21	2%
information-additional	23	2%
Total	1,199	100%

Table 3.10: Distribution of unit types in SciARG-CL.

It is interesting to observe that, in computational linguistics abstracts, there is a tendency to describe the methods used to obtain the reported results in the same sentence: while 6% of the units of type are labeled as *result-means*, only 2% are considered to be reporting only *means* (methods/procedures/datasets). The decision to include the type *result-means*, therefore, is validated by annotators' usage of this label. As mentioned in Section 3.2.2, excluding this particular case, very few sentences (only 6%) are annotated with a secondary type, which is in line with the observations made in the preliminary annotation experiments, as mentioned in Section 3.2.2.

In Table 3.11, statistics for the annotation of relations show that *support* relations are the most frequent ones, followed by *elaborations*. This is expected if we observe the frequency of the different types and the correspondences between the types of the units and the relations in which they participate.

Relation	**Number**	**Percentage**
support	420	35%
elaboration	355	30%
info-required	118	10%
sequence	31	2%
by-means	28	2%
info-optional	22	2%
root	225	19%
Total	1199	100%

Table 3.11: Distribution of relations in SciARG-CL.

In Fig. 3.5 we can observe that the *root* unit is, in most of the cases, a unit of type *proposal* (in few cases it is a unit of type *proposal-implementation* and, in even fewer cases, of type *conclusion*). Units of type *proposal-implementation* are in general linked by *elaboration* relations to their parent units. An exception is the case in which they are part of the description of a *sequence* of steps. Units with coarse-grained type *outcomes* (*results*, *result-means*, *observation*, *conclusion*) are, in general, linked to their parents by *support* relations. Most of the *info-required* relations involve units of type *motivation-background*. As explained in Section 3.2.2, it is somewhat frequent to find, in computational linguistics abstracts, that the motivation for the described proposal is built-up with a combination of *motivation-background* and *motivation-problem* units in which the *motivation-problem* is the final element in the argumentative chain. Finally, as expected, *means* and *information-additional* units are practically exclusively linked to their parents with *by-means* and *info-optional* relations, respectively.

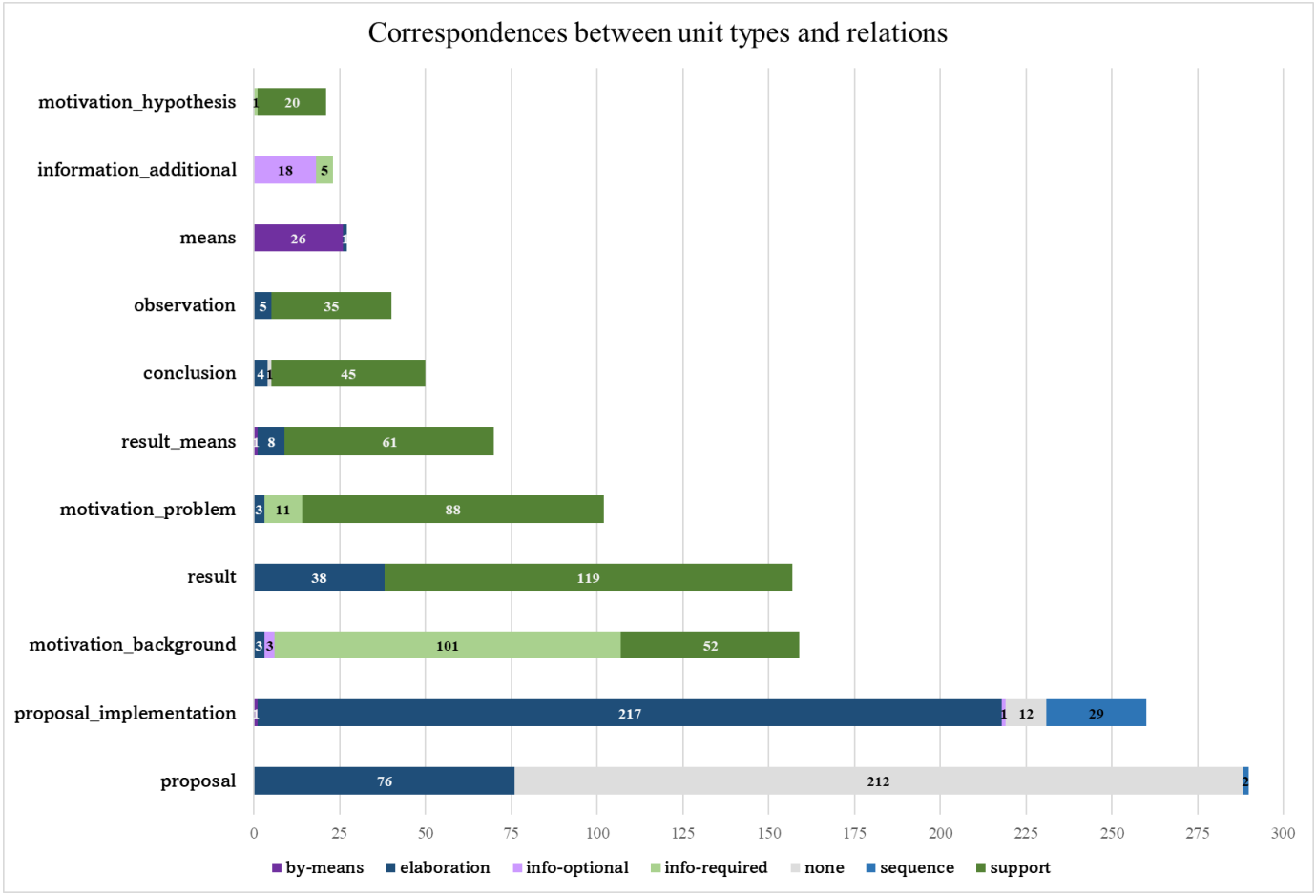


Figure 3.5: Correspondences between unit types and relations in SciARG-CL

In order to measure the level of correspondence between unit types (T) and relation (R) labels, we compute the *normalized mutual information* (NMI) coefficient (Shannon, 1949; Kreer, 1957) for the two sets of labels, obtaining $NMI_{RT} = 0.54$.[23] It is expected that if not only the type of the child node, but also the type of the parent node is considered, the possibilities for labeling the relation would be more limited. We therefore compute also the NMI coefficient between the relation label (R) and the combination of child and parent's types (T_{CP}). In this case we obtain $NMI_{RT_{CP}} = 0.57$, indicating that there is, in fact, overlapping information in the two sets of labels. This means that in computational linguistics abstracts[24] the argumentative function of a sentence (i.e., the relation to its parent node) is correlated with its type (which can be considered as the relation of the sentence with the whole text).

These different levels of analysis is what different annotation schemes capture, depending on whether they focus on the rhetoric role of the sentences, or whether they focus on their argumentative or discourse function when linked to other sentences. As mentioned, one of the goals of this work is to establish links between these two levels of analysis.

Not surprisingly, the distance and direction of the relations are also linked to the types of the units, as shown in Figure 3.6. We can observe that *motivation* units precede the *proposals* that they support, while units describing *implementation* details are more naturally placed after the more general *proposal* has been introduced in the text. This is also the case of units describing *outcomes* such as *results* and *conclusions*.

[23] The normalized coefficient is in the range [0, 1], where 1 means that there is a perfect correlation between both sets.

[24] In Chapter 6 we see that in other disciplines, such as biomedicine, there is more variability.

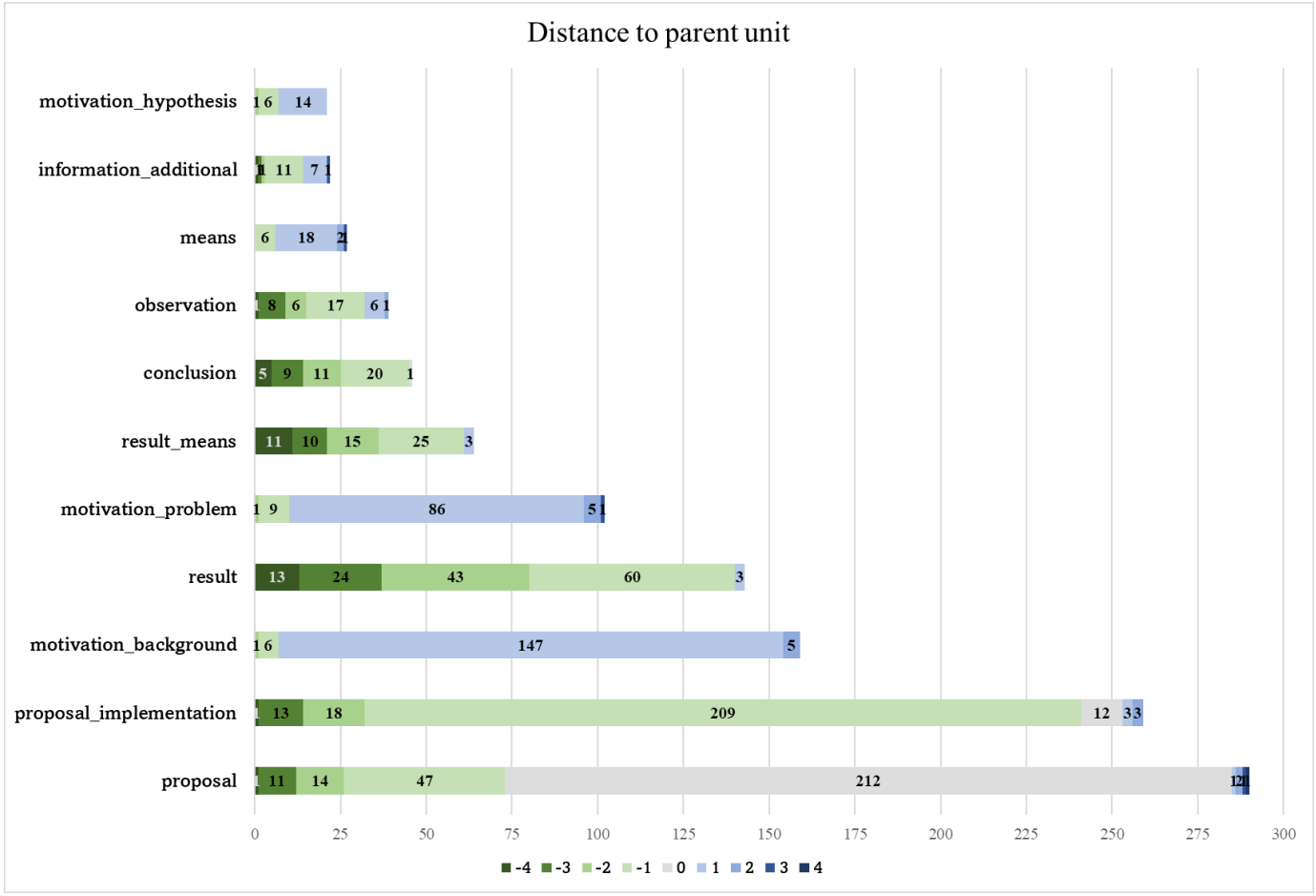


Figure 3.6: Distance to parent by unit type in SciARG-CL

Tables 3.12 and 3.13 show the percentage distances between units and their parents in SciARG-CL. We can observe that the most frequent case is that units are attached to an adjacent unit and that *backward* relations–i.e., the parent unit occurs before in the text–are almost 50% more frequent than *forward* relations.

Distance	Number	Percentage
1	288	29.6%
2	18	1.8%
3	4	0.4%
4	1	0.1%
Total	311	31.9%

Table 3.12: *Forward* relations. Distribution of the distances to parent units.

Distance	Number	Percentage
1	416	42.7%
2	111	11.4%
3	76	7.8%
4	33	3.4%
5	14	1.4%
6	8	0.8%
7	2	0.2%
8	1	0.1%
10	1	0.1%
11	1	0.1%
Total	663	68.1%

Table 3.13: *Backward* relations. Distribution of the distances to parent units.

It is also more frequent that *backward* relations are established at longer distances. While for *forward* relations in very few cases the parent unit is not adjacent to the child unit–and, when this happens, there are at most three units in between, while in the case of *backward* relations this occurs much more frequently.

In the SciARG-CL corpus the longest distance identified involves a relation in which there are 10 units between the child and parent nodes. These cases are exceptional in computational linguistic abstracts.[25]

[25]This case corresponds to the abstract of (Li and Fung, 2014), available at `aclweb.org/anthology/D14-1098.pdf`, which is considerably longer than the average abstract in

3.6 Conclusions

In this chapter we introduced SciARG, our proposed annotation scheme for argumentative units and relations in scientific abstracts, which is informed by previous works in the areas of argument mining, rhetorical analysis of scientific texts, and analysis of discourse structures.

We motivated our decisions to adopt sentences as annotation units, to enable the annotation of sentences with more than one type of unit, and to include the combined type *result-means*.

We described the annotation process in which the proposed scheme was applied to the annotation of 225 computational linguistic abstracts from the EMNLP 2014 Conference–included in the SciDTB corpus, generating the SciARG-CL corpus of scientific abstracts.

We analyzed inter-annotator agreement obtained in SciARG-CL for the four types of annotations included in the scheme (*unit type*, *relation type*, *parent attachment* and *main unit*), observing substantial agreement in terms of both pair-wise Cohen's κ and accuracy, which we propose as a more intuitive measure of the distance between graphs resulting from two different annotations of the same text.

We identified characteristics of the annotations, including the level of information shared between different labels (such as units and relations), and, finally, we analyzed the specificities of the argumentative structures of the annotated computational linguistics abstracts, including the most frequent links between sentences–both in terms of the direction and distance between them.

computational linguistics. In this case, the last sentence of the abstract is linked by a *support* relation to the first one.

Chapter 4

MINING ARGUMENTS IN THE SCIARG-CL CORPUS

In this chapter we describe how we use the SciARG-CL corpus described in Chapter 3 to train and evaluate models aimed at predicting the argumentative structure of scientific abstracts. The chapter is organized as follows:

- In Section 4.1 we describe the four tasks that we propose in order to model the prediction of the argumentative structure of scientific abstracts.
- In Section 4.2 we briefly overview BERT (Devlin et al., 2019), the base architecture that we use for all our models.
- In Section 4.3 we describe the experimental setups used in our experiments, including a description of the loss function used when training all the tasks jointly, in a multi-task setting.
- In Section 4.4 we explain the strategy implemented for selecting models that are then used for evaluation or prediction.
- In Section 4.5 we analyze the results obtained with the experiments described in the previous sections.
- In Section 4.6 we investigate potential benefits obtained by leveraging discourse-level annotations. In particular, we explore whether including a supplementary fine-tuning stage with sentence-level discourse tasks can contribute to improve the performance of our models. We describe the discourse-level

tasks and the models trained with them, as well as the new experimental setups, and compare the results obtained with the ones obtained in Section 4.5. We explore the benefits obtained with the new approach in function of the size of the training sets for the target tasks. For this, we consider, in addition to the results obtained with the full SciARG-CL, results obtained when using only 25% and 50% splits of the corpus.

- In Section 4.7 we propose a set of heuristics to post-process the predictions obtained with the models that are to be used in downstream applications, in order to ensure the well-formedness of the predicted trees. We evaluate the impact that these transformations have in the results obtained for the *parent attachment* and *main unit* tasks (as these are the two tasks that can be affected by changes made to the structure of the graph).
- In Section 4.8 we summarize the main contributions and results of this chapter.

4.1 Tasks

As mentioned in Chapter 3, in order to capture the argumentative structure of a text it is necessary to identify its components and how they are linked to each other. Based on the SciARG-CL annotations, we consider the following set of tasks for the prediction of the argumentative structure of scientific abstracts:

Unit type: Given a sentence, predict its type. The class to predict in this case is one of the eleven fine-grained types described in Chapter 3.[1]

Parent attachment: Given two sentences, predict i) whether they are related and, if that is the case, ii) whether a *forward* or *backward* relation exists between them (i.e., whether the first unit is a child of the second one in the argumentative tree or *vice versa*). We model this task as a three-class classification problem where, given two sentences, the possible classes to predict are *forw*, *back* or *none*, indicating, respectively, that there is a directed relation from the first to the second sentence, from the second to the first sentence, or that the two sentences are not related.

[1] The ten atomic types plus the most frequent combined type, *result-means*.

Relation type: Given a sentence, predict the label of the relation with its parent. The class to predict in this case is one of the six relations described in Chapter 3 plus *none*, for the *root node*.

Main unit: Given a sentence, predict whether it is the *main unit* of the text. The two possible values in this case are *main* or *secondary*.

In Chapter 3 we observe that, in SciARG-CL, there is overlapping information between the types of the units and the relations in which they participate. To assess what this implies in practical terms, we train a decision tree classifier[2] that, given the types of related child and parent nodes, predicts the relation label. We train and evaluate this classifier in a 10-fold cross-validation setting, obtaining a weighted-averaged F_1 score of 0.9110 and a macro-averaged F_1 score of 0.7493. This means that, given the types of two related nodes, we could predict with a high level of accuracy the relation label for the most frequent relations.

We nevertheless consider the prediction of unit types and relation labels as separate tasks because: i) even if there is a high correlation there is not a total match between the two tasks–in particular, for less frequent relations; ii) it is not evident that these results could be directly extrapolated to abstracts in other domains, with a higher level of argumentative complexity (we see this in more detail in Chapter 6), and iii) the two annotation levels convey different types of information: it could be the case that, in a downstream application, we are interested in obtaining only the labels of the relations in the argumentative structure without having to predict the types of the nodes beforehand.

It is also relevant to note that we consider the *parent attachment* and the *relation type* as separate tasks. Another alternative would have been to model them as one single task. If we encoded the type and direction of the relation in the same label, we would have 13 potential labels for this task (two labels for each of the six types of relations, each one for each direction, plus one *none* label when there is no relation between the two sentences). We opt, instead, to split them into two tasks, which makes it possible to train them in jointly in a multi-task setting, but having each one their specialized classifier, each one dealing with a smaller set of potential labels.

[2]We use Weka's implementation of the C4.5 algorithm (Hall et al., 2009)

4.2 Base architecture

In pilot experiments described in Appendix A,[3] we used as base architecture Bi-directional Long Short-Term Memory (BiLSTM) neural networks (Graves and Schmidhuber, 2005)[4] and ELMo (Embeddings from Language Models) (Peters et al., 2018) as contextualized representations of the words. After these preliminary experiments, *transformer* architectures based on the self-attention mechanism were proposed, outperforming recurrent neural network (RNN) architectures such as BiLSTM in several text-classification tasks (Vaswani et al., 2017; Galassi et al., 2020).

In 2019 a new transformer-based language representation model, BERT (Bidirectional Encoder Representations from Transformers), was introduced, obtaining new state-of-the-art results on eleven NLP tasks (Devlin et al., 2019). BERT pre-trained models were made available to the research community[5] and can be easily fine-tuned for specific tasks by adding different output layers on top of the encoders.

BERT models are pre-trained with two training objectives: i) Masked Language Model (MLM), where, given a sentence, some words are randomly masked and the objective is to predict the masked words from their context, and ii) Next Sentence Prediction (NSP), where, given two sentences, the objective is to predict whether the second sentence comes after the first one in a text. BERT is, therefore, particularly well-suited both for *single sentence* classification tasks as well as for modeling relations between *pairs of sentences*.

Figs. 4.1 and 4.2 show how single sentences and pairs of sentences are processed in classification tasks, where E_i represents the input embedding corresponding to token i and T_i represents the learned contextualized representation of token i. `[CLS]` is a special token added in front of the input sentence–or pair of sentences– and `[SEP]` is a special token used to separate tokens from the first and second sentences.

[3] Included in (Accuosto and Saggion, 2019b).

[4] In this case, as we designed the tasks as token-classification problems, we used a conditional random fields (CRF) classifier on top of the BiLSTM network.

[5] https://github.com/google-research/bert

The standard practice when fine-tuning BERT is to take, in the *pooling* layer, the contextualized representation of the `[CLS]`token (`[C]`) and feed it into a *classification* layer that depends on the specific task at stake. A *softmax* function is then applied to the classifier's output in order to obtain the distribution of probabilities for the predicted labels. The length of the input sequences used in each case is a hyper-parameter of the model, being 512 tokens the maximum length supported by BERT. Sequences shorter than the established length are padded with the special token `[PAD]`–which is ignored when performing the attention operation.

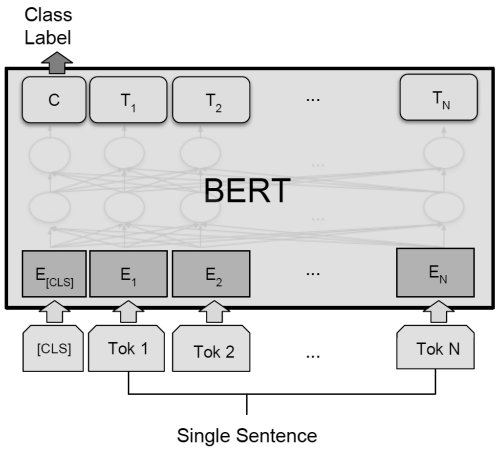


Figure 4.1: Classification of single sentence in BERT. Source: (Devlin et al., 2019).

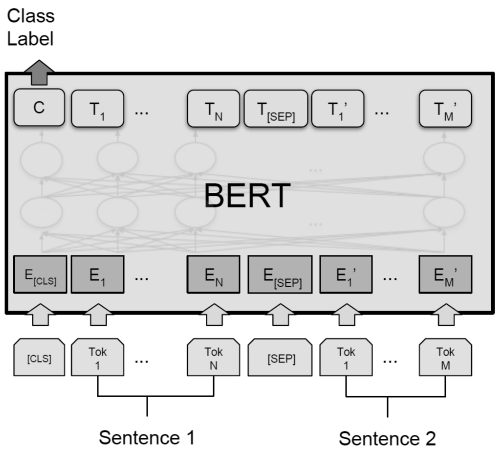


Figure 4.2: Classification of pair of sentences in BERT. Source: (Devlin et al., 2019).

4.3 Experimental setups

For our experiments we use the cased version of AllenAI's SciBERT (Beltagy et al., 2019) as base model, which has shown to improve the performance of several NLP tasks in scientific language. SciBERT is a BERT model pre-trained on a random sample of 1.14M papers indexed by Semantic Scholar[6] in computer science and biomedical disciplines (18% and 82%, respectively). SciBERT is made available[7] with its own vocabulary (SciVocab) (both in cased and uncased versions), to match the training dataset.

We use in all our experiments PyTorch implementations of BERT available as part of HuggingFace's Transformers library (Wolf et al., 2020).

4.3.1 Training parameters

To fine-tune our models we consider the median number of tokens in the input sequences and set the maximum sequence length to its double. Longer sequences are truncated. We found that this provides a good balance between training time and performance. While the general recommendation is to fine-tune BERT for 2 to 4 epochs (Devlin et al., 2019), we observed that more epochs were required to train our tasks. Determining the optimal hyper-parameters for each task, in general, and, the number of training epochs, in particular, is far from trivial. In Section 4.4 we explain the model selection method used throughout our experiments, both to report the obtained results and to choose the model checkpoints used for predictions and/or for additional fine-tuning, as described in Section 4.6.

In order to simplify the experiments and the analysis we do not perform hyper-parameter optimization for each task and setting–with the exception of the number of epochs. For all the experimental settings described we use *Adam* with weight decay (Loshchilov and Hutter, 2018) as optimization algorithm, with a linear warm-up learning rate schedule.[8] We fix the dropout probability in 0.1 for multi-task settings and 0.2 for single-task ones, as in multi-task settings each task functions as a regularization factor with respect to the other tasks and it is not nec-

[6] `semanticscholar.org`

[7] `github.com/allenai/scibert`

[8] We set an initial learning rate of 2e-5 with a warm-up period of 10% of the learning steps.

essary to establish a high-dropout to prevent overfitting, which is more likely to occur in single-task settings (Baxter, 1997). The batch size used is of 16 instances with gradient accumulation of 2 batches,[9] therefore having effective batch sizes of 32 instances when training.

4.3.2 Multi-task experiments

We observe, in Section 4.1, that the tasks considered in our experiments capture different perspectives of the information conveyed by the annotations. In particular, we analyze the association between the prediction of the types of the nodes in the argumentative tree and the prediction of the labels of the relations between them, which is the objective of the *relation type* task. A correspondence can also be established between the *main unit* and *parent attachment* tasks, as the *main unit* is, in 98% of the cases, also the *root* of the argumentative tree. This means that, for the *main unit*, the predicted direction of the relation should, in general, be *none* when paired with all of the other sentences in the abstract. It is therefore natural to wonder whether the training signal of one task can contribute to improve the performance of the others. We explore this possibility by training the tasks jointly, in a multi-task setting, and compare the results to those obtained when training each task independently.

4.3.2.1 Input format for training and evaluation

When training multi-task networks with different types of input data for each task, issues such as the way in which batches are sampled for each task can have a critical impact on the performance of the models (Subramanian et al., 2018).

We propose to simplify this process by unifying the format of the input data and training the four tasks in parallel with the same instances–with one label for each task. We compute a loss function that combines the losses of each individual task and then update the network's parameters once, instead of alternating between task-specific batches.

[9] As a way to deal with the memory limitations in our computing environment.

We model the unified input instances as sequences containing *pairs of sentences*. In the cases in which we only need to predict a label for one sentence (for instance, for the *unit type* task), we predict the label corresponding to the first sentence of the pair. The second sentence can be thought as providing context. We conducted exploratory experiments and observed that this not only does not impair the performance of single-sentence tasks, but actually contributes to slightly improve it, as mentioned below.

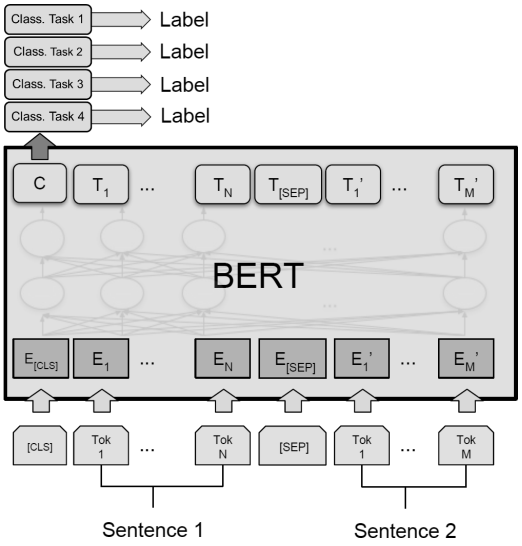


Figure 4.3: Multi-task classification in BERT.

Parent attachment task

Let us suppose that we have an annotated text with 5 sentences, represented by the argumentative graph in Fig. 4.4, where the subscript numbers represent the absolute position of the corresponding sentence in the text.

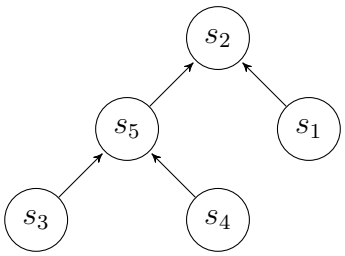


Figure 4.4: Graph representing an annotated text with 5 sentences.

We can represent the graph with the following triplets:

$< s_1, s_2, \textit{forw} >; < s_1, s_3, \textit{none} >; < s_1, s_4, \textit{none} >; < s_1, s_5, \textit{none} >;$
$< s_2, s_3, \textit{none} >; < s_2, s_4, \textit{none} >; < s_2, s_5, \textit{back} >;$
$< s_3, s_4, \textit{none} >; < s_3, s_5, \textit{forw} >;$
$< s_4, s_5, \textit{forw} >$

Training stage

The most probable case, given any two sentences, is that they are not related. This means that there will be many more instances labeled with *none* than with *forw* or *back*. In order to provide the model with more *positive* instances, we sample related pairs twice, once for each direction. Therefore, in addition to the set of instances considered above, we include, for training, also:

$< s_2, s_1, \textit{back} >;$
$< s_5, s_2, \textit{forw} >; < s_5, s_3, \textit{back} >; < s_5, s_4, \textit{back} >$

In order to predict the existence and direction of the relations we have to consider all the possible combinations of sentences in the text.

Evaluation / prediction stage

For evaluation/prediction, we consider each combination of sentences only once, in the order in which the occur in the text:

$< s_1, s_2, ? >; < s_1, s_3, ? >; < s_1, s_4, ? >; < s_1, s_5, ? >;$
$< s_2, s_3, ? >; < s_2, s_4, ? >; < s_2, s_5, ? >;$
$< s_3, s_4, ? >; < s_3, s_5, ? >;$
$< s_4, s_5, ? >$

Single-sentence tasks

Training stage

As mentioned, in order to train all the tasks in parallel with the same input data, we feed the model with the same set of instances, with one label for each task. In the case of single-sentence tasks, the label corresponds to the class of the *first sentence* in the pair.

For instance, let us suppose that in the graph considered in Fig. 4.4, the types of the units are the ones shown in Table 4.1.

Position	Type
s_1	*motivation-background*
s_2	*proposal*
s_3	*observation*
s_4	*means*
s_5	*result*

Table 4.1: Examples for types of nodes in Fig. 4.4

In order to train the *unit type* classifier, for instance, we feed the model with the same instances used for the *parent attachment* task, but in this case the label used to train the classifier corresponds to the *unit type* of the *first sentence* in the pair. In the example being considered, the instances that the *unit type* classifier process, would therefore be:

$<\mathbf{s_1}, s_2, \textit{motiv-b}>$; $<\mathbf{s_1}, s_3, \textit{motiv-b}>$; $<\mathbf{s_1}, s_4, \textit{motiv-b}>$; $<\mathbf{s_1}, s_5, \textit{motiv-b}>$;
$<\mathbf{s_2}, s_1, \textit{prop}>$; $<\mathbf{s_2}, s_3, \textit{prop}>$; $<\mathbf{s_2}, s_4, \textit{prop}>$; $<\mathbf{s_2}, s_5, \textit{prop}>$;
$<\mathbf{s_3}, s_4, \textit{observation}>$; $<\mathbf{s_3}, s_5, \textit{observation}>$;
$<\mathbf{s_4}, s_5, \textit{means}>$
$<\mathbf{s_5}, s_2, \textit{result}>$; $<\mathbf{s_5}, s_3, \textit{result}>$; $<\mathbf{s_5}, s_4, \textit{result}>$

It can be observed that we are not sampling all the instances in a balanced way. In the example, s_1 is seen by the model four times paired with different context sentences in every epoch, while s_4 is seen just once per epoch.

Neither the sampling strategy nor the fact that single-sentence tasks as modeled with pairs of sentences impact negatively on the performance of single-sentence tasks, as described in Section 4.3.3. Based on these considerations, we decide that we can safely adopt the proposed method, with the benefit of having a unified way of modeling the input data for all the tasks, independently of whether the objective is to obtain a label for the pair of sentences or just for the first sentence.

Evaluation / prediction stage

For the prediction of labels in the case of single-sentence tasks (including the predictions done for the evaluation of the model), we consider each sentence only once in the first position of the pair. In the example, we would therefore predict only the classes for the following instances:

$$< \mathbf{s_1}, s_2, ?>; < \mathbf{s_2}, s_3, ?>; < \mathbf{s_3}, s_4, ?>; < \mathbf{s_4}, s_5, ?>; < \mathbf{s_5}, s_4, ?>;$$

This means that each sentence is considered with the next sentence in the text as context, except for the last sentence, where the previous sentence is considered as context.

4.3.2.2 Loss function for multi-task models

When training multi-task models the objective is to minimize the combined losses for all the tasks. We therefore want to minimize:

$$L = \sum_{t=1}^{T} w_t.l_t$$

Where T is the total number of tasks, l_t is the loss for task t and w_t is a weighting factor that indicates how much the loss of task t should contribute to the overall loss. These weights are necessary because if the tasks do not have the same level of difficulty and/or the losses' scales are very different, it can occur that one task dominates the overall loss, impacting negatively on the performance of the other tasks. Determining how to optimally weight each task when computing the overall loss is therefore very important (Gong et al., 2019).

We follow the proposal by Kendall et al. (2018), in which the tasks' weights are learned as parameters of the training process by considering the tasks' *homoscedastic uncertainty*.[10] This means that the higher the uncertainty of a task, the smaller its contribution to the total loss.

[10]Which refers to a level of uncertainty that depends on the task and not on specific inputs. Please see (Kendall et al., 2018) for a more detailed explanation and the formula used in the computation of the weights.

In practice, we consider a set of trainable parameters $\{\eta_1, \ldots, \eta_T\}$ and compute as global loss function:

$$L = \sum_{t=1}^{T} (l_t . e^{-\eta_t} + \eta_t)$$

Where the η_t terms are also added as a regularization factors to prevent the network to set them to arbitrarily high numbers. For all our tasks t, we consider *cross entropy* as the base loss function l_t to optimize.

4.3.2.3 Additional features as special tokens

We observed that the models' overall performances tend to improve when including, as additional tokens, information about the position of the sentences in the abstract, as well as the relative distance and order of the sentences in the input pairs.

This information is codified by means of *special tokens* (such as `[POS_1]`, `[DIST_1]`, `[AFTER]`, `[BEFORE]`) added to BERT's tokenizer and included in the sequence of tokens representing the pair of sentences before it is processed by the model. For instance, let us consider the example graph in Fig. 4.4. After the tokenization process, the input sequence corresponding to the pair $< s_5, s_2 >$ would be:

`[CLS] [AFTER] [POS_5] [DIST_3]` $T^1_{s_5} \ldots T^N_{s_5}$ `[SEP]` $T^1_{s_2} \ldots T^M_{s_2}$ `[SEP]`

Where $T^i_{s_j}$ represents the *ith* token of sentence j. In the example, `[POS_5]` indicates that the first sentence in the pair is in the fifth position (s_5) and `[AFTER]` and `[DIST_3]` indicate that it occurs three positions after the second sentence of the pair (s_2). `[CLS]`, `[SEP]` and `[PAD]` are the standard BERT special tokens, as explained in 4.2.

4.3.3 Single-task experiments

In the case of single-task experiments, we fine-tune BERT and train one linear classifier on top of it for each task independently.

In this case the input data is not shared among the different tasks. It is therefore not necessary to keep a unified format: we could model the *parent attachment* task as a pair classification task, as described in Fig. 4.2, and, for single-sentence tasks–such as the prediction of *unit types*–we could use the single-sequence input format as described in Fig. 4.1, sampling each sentence once per epoch in the training phase.

The problem that arises with this approach is that, if we want to compare the performance of single and multi-task models, it would be difficult to assess whether differences can be attributed to the different architectures or, instead, to the different ways in which the tasks are modeled in each case, and/or to the sampling strategy, which determines how many times each instance is seen by the model.

We run several cross-validation tests with the training data in order to evaluate whether modeling single-sentence tasks with pairs of sentences and/or implementing the sampling strategy used for the *parent attachment* task–where sentences are weighted in function of their position in the texts and the number of relations in which they participate–could impact negatively on the performance of single-sentence tasks. In particular, we explored:

- Training and evaluating single-sentence tasks with the standard single-segment encoding strategy described in Fig. 4.1;
- Training and evaluating single-sentence tasks with pairs of sentences, in the same way that we do in the case of the *parent attachment* task, but with different sampling strategies:
  - Sampling each sentence only once in the first position of the pair;[11]
  - Sampling each sentence in the first position of the pair with every other sentence in the text in the second position.

Based on these exploratory experiments, we found that:

- Modeling single-sentence tasks as pairs–where the second sentence is considered as context–does not impact negatively on the performance of single-sentence tasks. In fact, it contributes to slightly improve their performance (between 0.01 and 0.03 F_1 score points in average);

[11] With the next sentence in the text as the second element in the pair, as we do for evaluation.

- There are no significant differences, in terms of performance, between the different sampling strategies considered for pairs of sentences.

Both observations would require additional investigation to be explained in detail. Part of the explanation for the first finding might lie in the fact that BERT is designed to clearly distinguish between the first and second segments when the input is modeled as a pair, so information in the second segment can be taken advantage of, should it provide relevant information, and it can be ignored if does not. Training for several epochs also seem to contribute to smooth any differences that might arise due to differences in the number of times each sentence is sampled.

4.4 Model selection

As mentioned in Section 4.3.1, we fix all the training hyper-parameters with the exception of the number of *epochs* in which we train the model. In this section we describe the method used to select the *model checkpoints* to be used for evaluation and in downstream applications.

A popular method to determine the value of a parameter in data-driven models–e.g., the number of clusters in a clustering algorithm–is to plot the improvement of the model as a function of the parameter to find. In our case, the parameter to optimize is the number of *epochs*, and the improvement of the model is measured in terms of the *loss* in the training set. Fig. 4.5 shows, as an example, the graph obtained from plotting the normalized training loss as a function of the number of epochs when fine-tuning SciBERT with the *parent attachment* task in a single-task setting.

In general, in machine learning, the training loss is expected to continue descending as we continue training our model, but from a given point on the models to start to overfit the training data, losing generalization power. A standard way to heuristically determine a number of epochs in which the model has had time to learn the task and, at the same time, keeps generalization power, is to pick the value for which an *elbow* is observed in the plotted loss as a function of the epoch (Satopaa et al., 2011).

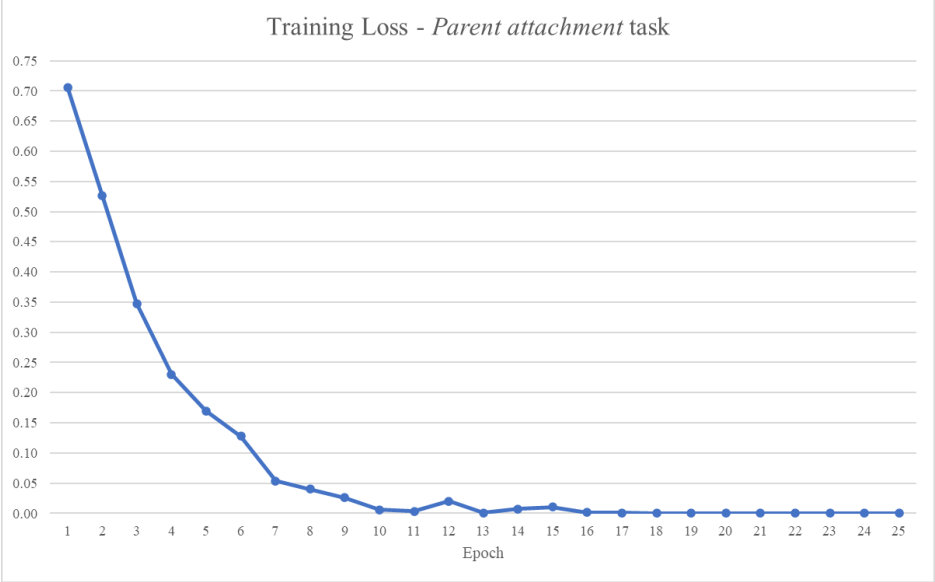


Figure 4.5: Normalized training loss by epoch when fine-tuning SciBERT for the *parent attachment* task.

We use the *kneed* Python package[12] to find the value of *elbow epochs* by: i) computing a function f to fit the data points, ii) rotating f to obtain a function f' such that the $f'(x_1) = f'(x_n) = 0$, being $x_1$and x_n the first and the last values for the x axis (in our case, n=25), and iii) finding the value k such that $f'(x_k)$ is a maximum. The point k is the one for which the elbow is obtained.

The difficulty with this method is that, in many cases, it is not trivial to determine one single elbow, as can be observed in the graph (the elbow could be considered to be 7, but also 8, 9 or 10). Depending on the method used to find the interpolating function f, different elbows might be obtained. The *knee* package implements two different methods: i) fitting a spline to the input x and y data,[13] and ii) fitting a polynomial function.[14]

Fig. 4.6 shows the f' functions obtained for both interpolation methods[15] and the two candidate epochs selected for the *parent attachment* task: epochs 7 and 9 for

[12] `kneed.readthedocs.io`

[13] By means of SciPy's *interp1d* function. `docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html`

[14] By means of NumPy's *polyfit* function. `numpy.org/doc/stable/reference/generated/numpy.polyfit.html`

[15] We use a cubic polynomial.

spline and polynomial interpolations, respectively. We consider both elbow candidates and return the rounded-up median between them. We therefore consider, in the example, epoch 8 as the elbow of the loss function.

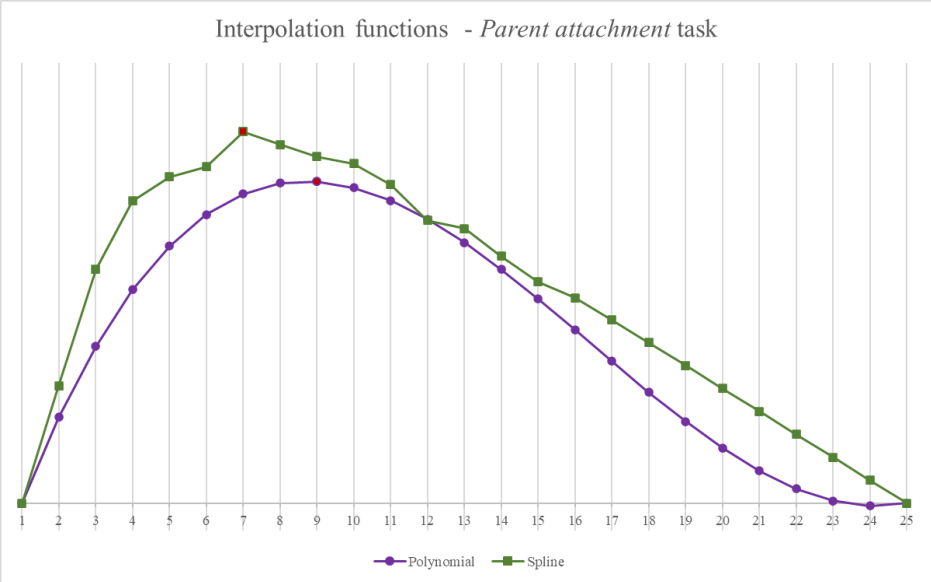


Figure 4.6: Interpolation methods used to find the elbow for the *parent attachment* task.

4.5 Results and analysis

In this section we report the results obtained for the four tasks described in Section 4.1 when trained both in single and multi-task learning settings. We perform the evaluations considering, as validation set, the *consensus annotations* described in Chapter 3, using the rest of the annotations for training. Table 4.2 shows the split of instances intro training/validation sets.

Set	**Abstracts**	**Instances**
Training	195	3431
Validation	30	151

Table 4.2: Number of training/validation documents and instances (pairs of sentences).

For the *unit type* and *relation type* tasks we use weighted-averaged F_1 scores as metric, as we want to consider the contribution of each label to the results in proportion to their frequency.

For the evaluation of the *parent attachment* task, instead, we use macro-averaged F_1 scores, which are more sensitive to mis-classifications in minority classes. If we were to use micro-averaged or weighted-averaged scores in these cases we would obtain misleading high average F_1 scores, given the large proportion of *none* labels which are correctly classified, even if the model performs poorly in the classification of *forw* and *back* classes, which are the ones we are most interested in. Similarly, for the evaluation of the *main unit* task, we macro-average the F_1 scores as we are particularly interested in evaluating how well the models perform for the minority class–since most of the sentences will be correctly classified as not being the *main unit*.

As mentioned in Section 4.4, selecting a single checkpoint model obtained in a particular epoch is necessary if we want to use the trained model for prediction, but it is not a trivial task. In fact, no method can ensure that the best possible model will be selected to predict the labels of unseen instances, so the information obtained from evaluating a single model checkpoint can be limited. In fact, comparing results obtained by any model selection method evaluates not only how well an architecture performs with unseen data but also the model selection method itself. In our case, let us suppose that the *best* checkpoint for a particular model is obtained in epoch *e*. If our elbow-based model selection method picks up instead epoch *e-1*–or *e+1*, for instance, this could have a significant impact on the results obtained when evaluating the model, potentially leading us to wrong conclusions when assessing how it performs in the validation set. As Ding et al. (2018) state, *"model selection, no matter how it is done, is exploratory in nature and cannot be confirmatory"*.

We are interested in assessing how the different architectures perform, trying to isolate as much as possible the potential errors introduced by the model selection method. Therefore, instead of just looking at the performance of the models in *one epoch*, we consider the average metrics obtained by a set of *five checkpoints* that include the epoch selected by the elbow method described in 4.4 (*elbow epoch*), two epochs before it and two epochs after it. The idea is that we compare the average performance of five *likely good-performing models* in the validation set, knowing that the resulting values will, in general, be below the maximum score.

Table 4.3 shows the results obtained for the four SciARG tasks trained in single and multi-task settings and evaluated in the validation set. In addition to the average scores, we include, for F_1, the confidence intervals obtained from the standard deviations. The results underlined are those statistically significant when considering the confidence intervals.

We observe that, for all the tasks, the F_1 scores tend to improve when the tasks are trained jointly in a multi-task setting–most significantly in the prediction of the types of units.

Single-task	**Average**	**Elbow**	**Epochs**	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{CI_{F_1}}$
Unit type	Weight.	7	5-9	0.8060	0.7748	0.7821	±0.0167
Relation type	Weight.	6	4-8	0.7991	0.7854	0.7832	±0.0204
Main unit	Macro	7	5-9	0.9023	0.9035	0.9026	±0.0066
Parent attachment	Macro	9	7-11	0.8042	0.8293	0.8150	±0.0207
Multi-task	**Average**	**Elbow**	**Epochs**	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{CI_{F_1}}$
Unit type	Weight.	7	5-9	0.8230	0.8026	**0.8005**	±0.0100
Relation type	Weight.	7	5-9	0.8084	0.7907	**0.7910**	±0.0129
Main unit	Macro	7	5-9	0.9287	0.9376	**0.9323**	±0.0109
Parent attachment	Macro	9	7-11	0.8202	0.8458	**0.8316**	±0.0125

Table 4.3: Results of fine-tuning SciBERT in SciARG tasks in single and multi-task settings. Average of models in epochs $[elbow-2, elbow+2]$ with 95% confidence intervals.

Fig. 4.7 provides a broader picture, as it shows not only the performance of the models in the vicinity of the *elbow epoch*, but how the prediction of the four SciARG tasks evolve when fine-tuning SciBERT in single and multi-task settings as a function of the number of epochs.

We can observe that, in fact, all of the tasks tend to perform better in a multi-task setting when given enough time–even if slightly for the *parent attachment* task and moderately for the *units* and *relations types*.

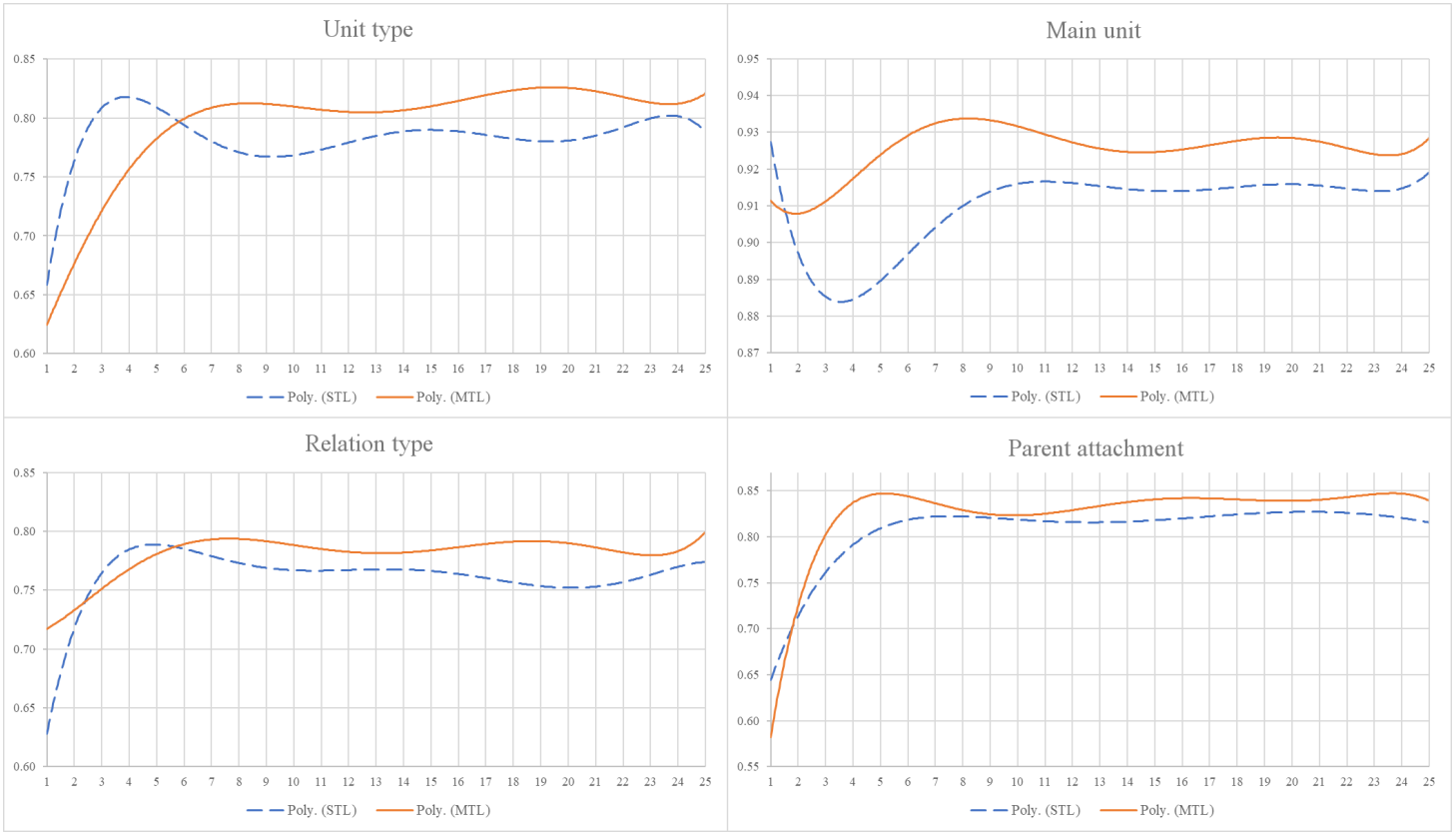


Figure 4.7: Polynomial trendlines of F_1 scores in consensus annotations set in function of the number of epochs. Comparison of SciBERT fine-tuned in single and multi-task learning settings.

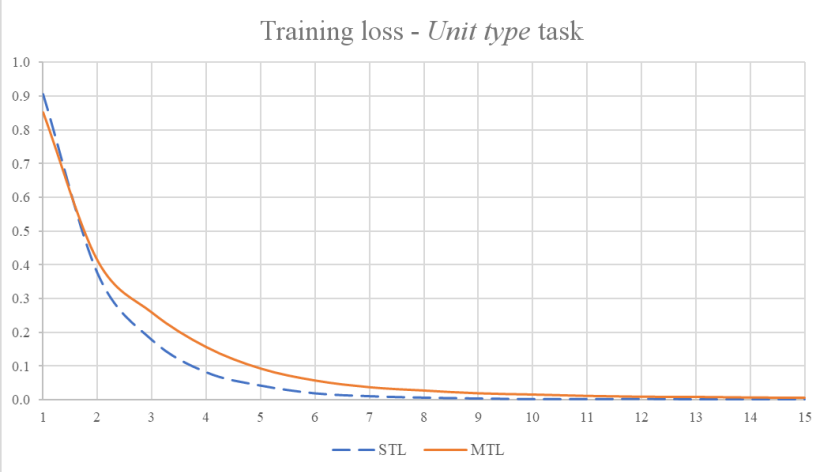


Figure 4.8: Training loss in function of the number of epochs for the *unit type* task in single and multi-task settings. Values are scaled between 0 and 1 for comparison.

It is not unusual to find that tasks require more training time in multi-task settings to surpass the performance of the tasks when trained independently.[16] Fig 4.8 shows, as a way of example, how the training losses of the *unit type* task evolve as a function of the number of epochs in single and multi-task settings.

We can see that the loss decreases more gradually in the multi-task settings than when the task is trained independently. In general, it is less probable for one particular task to overfit the training set in a multi-task setting, as the other tasks play regularization a role (Caruana, 1997). However, an interesting observation in the case of BERT is that, against intuition–and despite the large number of parameters contained in a BERT model–the fine-tuning procedure is robust to overfitting even for tasks trained in single-task settings when trained for longer periods (than the number of epochs originally recommended) and with an appropriately small learning rate (Hao et al., 2019; Mosbach et al., 2021). This can also be observed for all of the graphics included in Fig. 4.7, where we do not see single-task models dropping significantly in performance with more training epochs–in the case of the *main unit*, the initial performance is high and it drops sharply after a few training epochs, but it gradually increases again with more training time until it gets stabilized.

[16] Yet, of course, nothing ensures that, as rule-of-thumb, picking a large enough number of epochs for a task fine-tuned in a multi-task setting will yield better performance: it can be the case that a single-task setting just performs better–either in the short or long run.

In summary, if we accept that our results in the set of *consensus annotations* are representative of how the models perform in general with unseen instances, we can expect them to yield acceptably good predictions for the four SciARG tasks, both when trained independently and jointly in a multi-task setting. The prediction of the four tasks, nevertheless, is shown to benefit from the training signals of the other tasks–even if, in some cases, more training time is needed for the transferring to take place. Additional advantages of our multi-task approach are that i) it takes considerably less time to train–as the same data is used to train the four classifiers in parallel, and ii) instead of having to deal with four different models, we can use a single model to predict the four classes at once.

4.6 Leveraging discourse-level relations

The possibility of leveraging existing discourse-annotated corpora for the identification of arguments, its components, and the relations between them is a relevant research topic in argument mining, which has been addressed in works such as (Cabrio et al., 2013; Stab et al., 2014; Peldszus and Stede, 2016), as mentioned in Chapter 2.

In our exploratory experiments (Appendix A) we observed that the prediction of argumentative units and relations in a small set of abstracts improved with a sequential transfer learning approach in which we used, to train the argument mining models, word representations[17] pre-trained with discourse parsing tasks. In this section we further explore the possibility of transferring knowledge learned from discourse-level annotations to improve the prediction of the argumentative structure of scientific abstracts.

It has been observed that the performance of models based on encoders pre-trained with unsupervised language-modeling tasks–such as BERT–can improve when an intermediate stage of training on a supplementary supervised task is applied before fine-tuning on the target task (Phang et al., 2018; Pruksachatkun et al., 2020; Gururangan et al., 2020).

[17] In this case, ELMo embeddings.

In particular, the gain in performance tends to be more significant when there is a small number of annotations available for the target task. This approach is known as STILT (Supplementary Training on Intermediate Labeled-data Tasks).

In this section we explore the potential benefits of pre-fine-tuning AllenAI's SciBERT model with discourse-level tasks before fine-tuning it with our four target tasks (Section 4.1).

4.6.1 SciDTB tasks for intermediate fine-tuning

The SciDTB corpus (Yang and Li, 2018) contains 798 abstracts segmented into sentences which are, in turn, segmented into elementary discourse units (EDUs). Each EDU is attached to one parent EDU by means of a discourse relation,[18] with the exception of the *root* EDU. Each abstract can therefore be represented as a discourse dependency tree, as described in Chapter 3.

As shown in Table 4.4, the 798 unique abstracts were splitted by SciDTB authors into *training*, *validation* and *test* sets containing 492, 154 and 152 documents, respectively. Of the 492 abstracts in the training set, 200 were annotated by two annotators and 51 by three annotators, adding up to a total of 743 annotated abstracts used for training.[19] We therefore consider a total of 1,049 annotated abstracts for our experiments with the SciDTB corpus.

For the experiments described in this section we focus on SciDTB information at the *sentence level*. We define the *parent-child* relation between sentences in SciDTB as follows. Given two sentences s_c, s_p, we consider that the sentence s_c is a child of s_p if s_c contains an EDU e_c, and s_p contains an EDU e_p, such that there is an outgoing discourse relation from e_c to e_p.

There is only one *root* EDU in a discourse tree, which means that there is also one *root* sentence–a parent-less sentence–in each abstract (the sentence containing the *root* EDU).[20]

[18]The list of relations used in SciDTB is included in Chapter 3.

[19]Some abstracts in the test and validation sets were also annotated more than once in order to compute inter-annotator agreement, but the authors of the corpus produced harmonized annotations for these two subsets, which are the annotations that we are using in our experiments. They refer to these harmonized sets as *gold* validation/test sets.

[20]For *non-root* sentences (sentences without a *root* EDU), there should be at least one EDU

Based on RST's compositionality principle, we can safely assume that sentences can be predicted to have *one* parent–i.e., one outgoing discourse relation. This is, in fact, the case in SciDTB annotations.

Set	Abstracts	Sentences
Training	743	3,975
Validation	154	819
Training + Validation	897	4,794
Test	152	819
Total	1,049	5,613

Table 4.4: Distribution of number of abstracts and sentences in training and harmonized (*gold*) validation and test sets in SciDTB.

We consider the following tasks:

> **Discourse relation**: Given a sentence, predict the label of the discourse relation with its parent sentence–or $ROOT$ for the *root* sentence. The class to predict is one of the 26 fine-grained relations in Table 3.1, in Chapter 3.
>
> **Parent attachment**: Given a pair of sentences, predict whether i) there is a relation between them and, if that is the case, ii) whether it is a *forward* or *backward* relation. Analogously as we do for SciARG, we model this as a three-class classification task where the classes to predict are *forw*, *back* or *none*, when there is no relation.

For our STILT experiments we first fine-tune SciBERT with SciDTB's sentence-level tasks and, then, further fine-tune the resulting model for the four target SciARG tasks. The intermediate fine-tuning step could be done in different ways: i) to fine-tune each of the two SciDTB tasks independently, thus obtaining two different models and then use them to fine-tune each of the SciARG tasks, ii) to fine-tune SciDTB tasks sequentially (adding not one but two pre-fine-tuning stages),[21] and iii) training both SciDTB tasks jointly in a multi-task setting, thus obtaining a single model which we can then use to continue fine-tuning the SciARG tasks. Options i) and ii) have some drawbacks: in addition to the increased training time, we would be duplicating the number of results, introducing more complexity in

whose parent is in another sentence. Otherwise, a cycle would be produced, which would not be compatible with SciDTB annotations.

[21] In this case we should consider doing the two pre-fine-tuning in both orders.

the analysis.[22] We therefore opt to train both tasks jointly in a multi-task setting. The model selected to be further fine-tuned with our target tasks is obtained by applying the elbow-based model selection strategy used for SciARG's tasks.

4.6.2 SciDTB models

In this section we report the results obtained when training and evaluating the *discourse relation* and *parent attachment* SciDTB tasks with the training / validation / test split of the dataset proposed by the authors of the corpus (Table 4.4).

We are not using these models for anything other than to get an idea of how well we can expect SciBERT encoders to perform after a fine-tuning stage with SciDTB annotations. In our STILT experiments, described in Section 4.6.3, we do not use the original SciDTB split but instead exclude from the training set the SciARG-CL *consensus annotations* set that we use to analyze the performance of the models.

As we fix the parameters with the same values used for SciARG-CL and implement the same model selection method, we do not use the SciDTB validation set for hyper-parameter tuning but, instead, use the union of the training and validation sets for training (4,794 sentences) and then evaluate the selected checkpoint in the provided test set (819 sentences). In the same way as we do in SciARG-CL, we report the average metrics of the five models obtained in the *elbow epoch*, two epochs before and two epochs after it (Table 4.5.)

Task	**Average**	**Elbow**	**Epochs**	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{CI_{F_1}}$
Discourse relation	Weight.	7	5-9	0.6036	0.5897	0.5851	$\pm$0.0102
Parent attachment	Macro	8	6-10	0.7040	0.6850	0.6898	$\pm$0.0177

Table 4.5: Results of fine-tuning SciBERT with the two SciDTB tasks in multi-task setting. Average of models in epochs $[elbow-2, elbow+2]$ with F_1 confidence intervals.

The SciDTB tasks have moderate performances–in particular, the *discourse relation* task. This is expected if we consider that the *discourse relation* task involves the prediction of a large number of classes.

[22] Although it would be relevant to analyze the effect of each of the SciDTB tasks disagreggated.

4.6.3 STILT experiments with SciDTB and SciARG

In this section we explore the potential benefits of implementing a two-stage fine-tuning strategy, first with a large set of SciDTB annotations and then with our smaller-sized SciARG-CL corpus. We are, in particular, interested in exploring to what extent transferring parameters learned with SciDTB relations can contribute to improve the prediction of the *relations* in SciARG-CL (both in terms of their types and attachment to parents).

We therefore compare the performances of the models obtained when fine-tuning SciBERT directly with the SciARG tasks to those obtained when including an intermediate fine-tuning stage with SciDTB annotations.

4.6.3.1 Experimental setup

The experimental setup and evaluation strategy implemented in these experiments are identical to those described in Sections 4.3 and 4.5.

For the STILT experiments we first fine-tune SciBERT in a multi-task setting for the two SciDTB tasks. As base model–to continue fine-tuning the SciARG tasks– we consider the checkpoint obtained in the *elbow epoch* for the SciDTB's *parent attachment* task (as it is not only the better-performing task according to the results in Table 4.5, but it is also the potentially most informative task to fine-tune SciARG).

We use all the SciDTB dataset for fine-tuning, excluding the abstracts that are also in SciARG's validation set. We exclude these sentences to make sure that none of the base models being compared has been previously exposed to instances used for evaluation.

4.6.3.2 Results and analysis

In Tables 4.6 and 4.7 we consider the results obtained when the target tasks are trained in single and multi-task settings, respectively. The tables compare the results obtained when using as base model SciBERT with no previous fine-tuning to the results obtained by applying the STILT approach with an intermediate fine-tuning stage with SciDTB tasks.

Single-task setting

SciBERT model	Average	Epochs	P	R	F_1	CI_{F_1}
Unit type	Weighted	5-9	0.8060	0.7748	0.7821	±0.0167
Relation type	Weighted	4-8	0.7991	0.7854	0.7832	±0.0204
Main unit	Macro	5-9	0.9023	0.9035	0.9026	±0.0066
Parent attachment	Macro	7-11	0.8042	0.8293	0.8150	±0.0207
STILT w/SciDTB	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	4-8	0.8155	0.7960	**0.7977**	±0.0106
Relation type	Weighted	4-8	0.8113	0.8026	**0.8033**	±0.0198
Main unit	Macro	5-9	0.9135	0.9093	**0.9100**	±0.0077
Parent attachment	Macro	6-10	0.8175	0.8386	**0.8273**	±0.0052

Table 4.6: Comparison of results fine-tuning SciBERT directly vs. STILT fine-tuning w/SciDTB models. Both in single-task setting. Average of models in epochs $[elbow - 2, elbow + 2]$ with F_1 confidence intervals.

Multi-task setting

SciBERT model	Average	Epochs	P	R	F_1	CI_{F_1}
Unit type	Weighted	5-9	0.8230	0.8026	0.8005	±0.0100
Relation type	Weighted	5-9	0.8084	0.7907	0.7910	±0.0129
Main unit	Macro	5-9	0.9287	0.9376	**0.9323**	±0.0109
Parent attachment	Macro	7-11	0.8202	0.8458	0.8316	±0.0125
STILT w/SciDTB	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	5-9	0.8138	0.8013	**0.8023**	±0.0072
Relation type	Weighted	4-8	0.8474	0.8292	**0.8295**	±0.0169
Main unit	Macro	5-9	0.9572	0.9117	0.9322	±0.0083
Parent attachment	Macro	6-10	0.8272	0.8458	**0.8343**	±0.0094

Table 4.7: Comparison of results fine-tuning SciBERT directly vs. STILT fine-tuning w/SciDTB models. Both in multi-task setting. Average of models in epochs $[elbow - 2, elbow + 2]$ with F_1 confidence intervals.

Analogously as in the previous experiments, we consider the five models around the *elbow epoch* and average the results obtained with their predictions, in order to minimize potential errors introduced both by the model selection method and/or by artifacts resulting from the fact that we are using a rather small set for validation.

If we observe the average results obtained with the five models containing the *elbow epoch*, we can see that the performances tend to improve with an intermediate fine-tuning stage with SciDTB annotations. Nevertheless, the benefits obtained with an intermediate fine-tuning stage with the whole training set are, overall, moderate. As mentioned in Section 4.5, the performance of the four SciARG tasks is already good when fine-tuning SciBERT directly–in particular for the multi-task setting, which would indicate that the SciARG annotations included in the training set provide enough information for the models to extract the necessary information from them. The task the tends to benefit more from the STILT approach is the prediction of the *relation type*–in particular, in a multi-task setting. This could be in part due to the relational knowledge contained in the SciDTB base model and, in the case of the multi-task setting, could respond to a cumulative effect: as the prediction of the types of the units and the relations between them improve, it is expected that this information contributes to better determine the label of the relations, considering the mutual information existing between these tasks, as seen in Chapter 3.

4.6.3.3 STILT experiments with SciARG subsets

The STILT approach has proven particularly useful with small-sized datasets, where the training set does not provide enough information to directly fine-tune the target tasks (Phang et al., 2018). Therefore, in addition to consider the results obtained with models trained with the whole SciARG-CL training set, we are interested in investigating the impact of the STILT training strategy with a limited number of training instances. To do this, we analyze the results obtained with SciARG-CL models fine-tuned using only 50% and 25% of the training data.

Experimental setup

We split the training instances randomly into two and four sets–for the 50% and 25% experiments, respectively. We train models with each of the splits and then average the results obtained when evaluating the models with SciARG-CL validation set.[23] In order to select the *elbow epochs* we also consider the average of the training losses obtained in each epoch for all the splits.

[23] Averaging the results of four models ensures more stable results than if we were to train a single model with few training instances.

Results - ***Models trained with 50% splits of SciARG training set***

Single-task settings

SciBERT model	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	5-9	0.7701	0.7318	0.7367	±0.0068
Relation type	Weighted	5-9	0.7742	0.7437	0.7461	±0.0186
Main unit	Macro	5-9	0.8647	0.8873	0.8733	±0.0177
Parent attachment	Macro	6-10	0.7070	0.7467	0.7125	±0.0260
STILT w/SciDTB	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	4-8	0.7995	0.7642	**0.7665**	±0.0161
Relation type	Weighted	5-9	0.8135	0.7947	**0.7905**	±0.0150
Main unit	Macro	4-8	0.9097	0.9018	**0.9032**	±0.0160
Parent attachment	Macro	6-10	0.7520	0.8039	**0.7715**	±0.0133

Table 4.8: Results obtained with with two splits of 50% of SciARG's training set when fine-tuning SciBERT directly vs. STILT fine-tuning w/SciDTB models. Both in single-task setting. Average of models in epochs $[elbow - 2, elbow + 2]$.

Multi-task setting

SciBERT model	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	4-8	0.7478	0.7576	**0.7422**	±0.0254
Relation type	Weighted	4-8	0.7685	0.7788	0.7632	±0.0155
Main unit	Macro	3-7	0.8976	0.9194	0.9073	±0.0099
Parent attachment	Macro	6-10	0.7462	0.8087	0.7694	±0.0091
STILT w/SciDTB	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	4-8	0.7367	0.7530	0.7350	±0.0141
Relation type	Weighted	3-7	0.8285	0.8345	**0.8256**	±0.0189
Main unit	Macro	3-7	0.9238	0.9155	**0.9190**	±0.0069
Parent attachment	Macro	4-8	0.7914	0.8368	**0.8063**	±0.0150

Table 4.9: Results obtained with with two splits of 50% of SciARG's training set when fine-tuning SciBERT directly vs. STILT fine-tuning w/SciDTB models. Both in multi-task setting. Average of models in epochs $[elbow - 2, elbow + 2]$.

Results - ***Models trained with 25% splits of SciARG training set***

Single-task settings

SciBERT model	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	7-11	0.7422	0.7133	**0.7093**	±0.0138
Relation type	Weighted	6-10	0.6617	0.6692	0.6517	±0.0180
Main unit	Macro	7-11	0.8977	0.8891	**0.8885**	±0.0170
Parent attachment	Macro	7-11	0.6026	0.6301	0.5944	±0.0275
STILT w/SciDTB	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	5-9	0.7329	0.7162	0.7075	±0.0208
Relation type	Weighted	5-9	0.7271	0.7175	**0.7067**	±0.0251
Main unit	Macro	6-10	0.9109	0.8708	0.8836	±0.0100
Parent attachment	Macro	6-10	0.6905	0.7341	**0.7003**	±0.0149

Table 4.10: Results obtained with with four splits of 25% of SciARG's training set when fine-tuning SciBERT directly vs. STILT fine-tuning w/SciDTB models. Both in single-task setting. Average of models in epochs $[elbow - 2, elbow + 2]$.

Multi-task setting

SciBERT model	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	6-10	0.6486	0.6838	0.6468	±0.0155
Relation type	Weighted	5-9	0.6575	0.6884	0.6605	±0.0201
Main unit	Macro	3-7	0.8776	0.9042	0.8857	±0.0122
Parent attachment	Macro	9-13	0.5994	0.5942	0.5888	±0.0222
STILT w/SciDTB	**Average**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weighted	4-8	0.6830	0.7152	**0.6835**	±0.0185
Relation type	Weighted	4-8	0.7695	0.7762	**0.7562**	±0.0162
Main unit	Macro	2-6	0.9264	0.9218	**0.9217**	±0.0082
Parent attachment	Macro	7-11	0.7397	0.7826	**0.7555**	±0.0129

Table 4.11: Results obtained with four splits of 25% of SciARG's training set when fine-tuning SciBERT directly vs. STILT fine-tuning w/SciDTB models. Both in multi-task setting. Average of models in epochs $[elbow - 2, elbow + 2]$.

Summary of the results with 50% and 25% of the training data

Adding an intermediate fine-tuning stage with SciDTB relations' tasks significantly improves the prediction of SciARG-CL relations, both in terms of their types and attachment.

The relative gain in performance becomes more evident as the size of the training data for the target tasks is reduced. In fact, if we look at the F_1 scores obtained for *relation type* and *parent attachment* tasks–in particular, in multi-task settings, we observe that the performance achieved with the SciDTB fine-tuned models is comparable to the performance obtained with the whole training set when no previous fine-tuning is performed. For ease of comparison, in Table 4.12 we include only the scores obtained for these two tasks in multi-task settings at the respective *elbow epochs* with the percentage differences to the results obtained when fine-tuning the SciBERT model directly with the whole training set.[24]

Base model	**Task**	**Training data**	F_1	Δ
SciBERT	Relation type	100%	0.7910	
STILT w/SciDTB	Relation type	100%	0.8295	+5%
STILT w/SciDTB	Relation type	50%	0.8256	+4%
STILT w/SciDTB	Relation type	25%	0.7562	-4%
SciBERT	Parent attachment	100%	0.8316	
STILT w/SciDTB	Parent attachment	100%	0.8343	+3%
STILT w/SciDTB	Parent attachment	50%	0.8063	-3%
STILT w/SciDTB	Parent attachment	25%	0.7555	-9%

Table 4.12: Comparison of results obtained when directly fine-tuning SciBERT with 100% of the training set to those obtained with an intermediate fine-tuning stage with SciDTB with 50 and 25% of the training data. All results obtained in a multi-task setting.

If we consider the F_1 scores obtained with 50% of the training data with the SciDTB fine-tuned models, we observe a decrease in performance of only 3% with respect to the models obtained when fine-tuning SciBERT directly with 100% of the training set for the *parent attachment* task and, more surprisingly, a gain in performance of 4% for the *relation type* task.

[24] With the limitations already pointed out of looking at one single point for evaluation.

For the prediction of the *unit type* and the *main unit* tasks, we also observe substantial improvements with models trained in a multi-task setting with 25% of the training set, while there are no significant differences between STILT training or direct fine-tuning for these two tasks when trained independently. The opposite occurs with the models trained with 50% of the training data: while both tasks improve significantly in single-task settings, they perform similarly when trained jointly, in a multi-task setting (with a moderate improvement for the *main unit* task). These results trigger relevant questions with respect to how the transferring of knowledge occurs with intermediate fine-tuning stages and multi-task settings, and how these two transfer learning methods interact with small-sized training sets and for highly related tasks.

In Chapter 5 we apply a similar approach but leveraging information contained in a corpus annotated with rhetorical types similar to the AZ scheme, where we focus on the analysis of potential benefits of the STILT approach for the prediction of *unit type* labels.

4.6.4 Final model used for predictions

From Tables 4.6, 4.7 we observe that, when averages are considered, the models trained in a multi-task setting with an intermediate fine-tuning stage tend to perform similarly or better for all the tasks than when the tasks are trained independently and/or without an intermediary fine-tuning stage. We therefore adopt this training strategy for the models to be used for predictions in downstream tasks.

In this case, we need to pick just one checkpoint, so we take the model obtained at the *elbow epoch* when plotting the training loss in function of the number of epochs. In this case, the model in trained with the whole SciARG-CL corpus.

In Part II of the thesis we use features derived from predictions obtained with this model in a downstream task, which allows us to indirectly assess the utility of the model in a practical application.

4.7 Heuristics for well-formedness of predicted trees

When using the trained models to predict the argumentative structure of an unannotated abstract we consider, for the prediction of the relations, all the possible combinations of sentences in the order in which they appear in the text. As each decision is made locally, this could lead to structures that are not necessarily trees. Another possible violation of the *well-formedness* of the the argumentative tree can occur when more than one of the nodes are predicted as *main unit* or, conversely, when there is none. In this section we propose a set of post-processing steps in order to reduce significantly the possibility of obtaining non-valid trees.

These heuristics take advantage of the fact that, in the overwhelming majority of the cases in SciARG-CL gold annotations, the *main unit* corresponds to the *root* of the argumentative tree. The high accuracy in the prediction of the *main unit*, in turn, greatly facilitates the identification of the *root* node and the correction of potential ill-formedness in predicted trees.

I. Nodes with more than one parent

a) The most frequent situation of a node n_i being predicted as the child of more than one parent is when n_i is attached both to a node n_j and to one or more nodes n_{kx} such that n_{kx} are ancestors of n_j. In fact, the most frequent case is that n_i is attached to two nodes, n_j and n_k, where n_k is the parent of n_j, and therefore a grandparent of n_i. Other cases are rare. In this case, we remove the relations between n_i and the ancestors n_{kx}. An example is shown in Fig. 4.9.

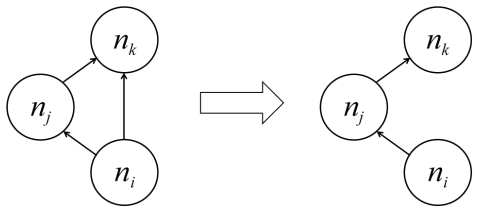


Figure 4.9: Example: Remove attachment to ancestor.

b) If a node n_i is attached to more than one node n_{p_1},... n_{p_m} as parents–where the subscripts indicate the position of the sentences in the text–and we are not in the situation described in ***a)***, we take as parent of n_i the node n_p that corresponds to the closest sentence in the text to n_i:

$$p = p_j, \text{ such that } |i - p_j| = \min_{k=1,...,m} |i - p_k|$$

If there are two nodes $n_{p_{j1}}$, $n_{p_{j2}}$ at the same distance: $|i - p_{j1}|$ = $|i - p_{j2}|$ (necessarily, one before and one after n_i), we consider as parent the node n_p that occurs before n_i, since background relations are more frequent in the annotations, as seen in Chapter 3. Therefore, we consider $p = p_j$ such that $p_j = \min\{p_{j1}, p_{j2}\}$.

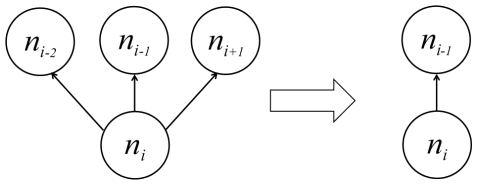


Figure 4.10: Example: Keep closest node as parent.

II. Cycles

After the application of step I, nodes have at most one parent. Therefore, for a cycle to be formed, there should be a sub-graph without a *root* node. In practice, cycles very rarely occur in the predicted graphs.

If a cycle is detected, we consider the nodes included in the sub-graph S containing the cycle.

a) If a node n_k in S is labeled as *main unit* we make n_k a *root* node–by removing any relation originating from it.

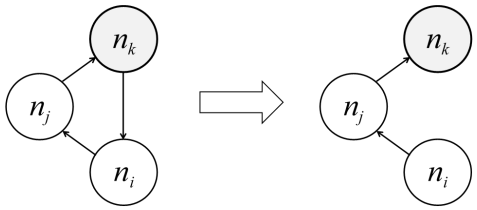


Figure 4.11: Example: Consider a node labeled as *main unit* as *root node*.

b) If no node in S is labeled as *main unit*, we consider the subset S' of nodes with the largest number of children. From S', we consider the node n that occurs before in the text and make it the *root* node of S.

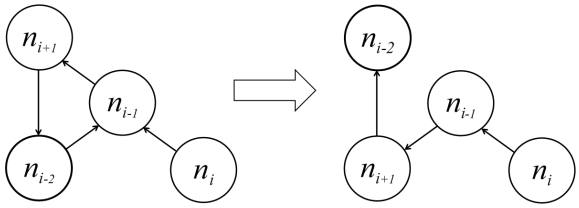


Figure 4.12: Example: Consider the node that occurs before as *root node*.

III. Main unit

a) If *no nodes* are labeled as *main unit* in the predicted graph:

After the application of steps I and II we know that there is at least one *root* node in the graph. We label the *root* node(s) as *main unit*. Note that it is possible that, in this step, we label more than one node as *main unit* (if there are unconnected trees). This is fixed in the next step.

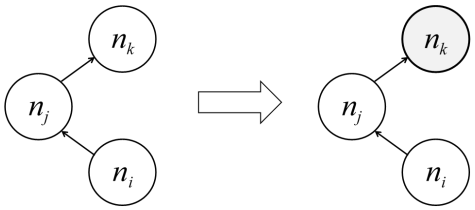


Figure 4.13: Example: Consider the *root node* as *main unit*.

b) If there is *more than one* node labeled as *main unit* in the predicted graph:

We consider as *main unit* the node with the largest number of children. If two or more nodes have the same number of children, we consider as *main unit* the one that occurs first in the text.

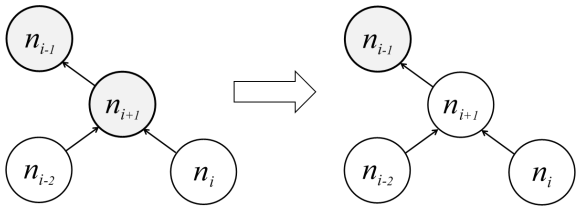


Figure 4.14: Example: Consider the node occurring before as *main unit*.

IV. Orphan nodes / trees

In a well-formed tree, there should only be one node without a parent, the *root* node.

Let us consider the subset R of *root* candidates (parent-less nodes). If one node r in R is labeled as *main unit* we consider it to be the *root* of the argumentative tree (from the previous step, we know that we will have at most one node labeled as *main unit*).

For the remaining nodes n_j in $R - \{r\}$, we attach n_j as child of a node n_k such that n_k is a non-leaf node at the minimum distance from n_j in the text.

If there are two non-leaf nodes n_{k1}, n_{k2} at the same distance from n_j–one after and the other one before, we consider the one that has the largest number of children. If they have the same number of children, we consider the one that occurs before in the text.

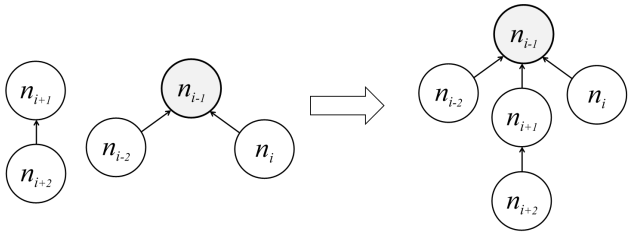


Figure 4.15: Example: Attaching an orphan tree to its parent.

4.7.1 Evaluation of *structural tasks* after post-processing

An analysis of the post-processed predictions shows that the proposed heuristics are effective to ensure the well-formedness of the predicted argumentative structures. For all the abstracts in the validation set we obtain single trees with one predicted *main unit*. In this section we explore to what extent the modifications introduced impact on the performance of the *main unit* and *parent attachment* tasks.

Main unit

Tables 4.13 and 4.14 show the scores obtained for the *main unit* task after applying the post-processing transformations to the predictions obtained with the best models described in Section 4.6.3.

Label	P	R	F_1	**Number**
main	0.9615	0.8333	0.8929	30
secondary	0.9600	0.9917	0.9756	121
Macro avg.	0.9608	0.9125	0.9342	151

Table 4.13: Original F_1 scores obtained with predicted labels for *main unit*.

Label	P	R	F_1	**Number**
main	0.9333	0.9333	0.9333	30
secondary	0.9835	0.9835	0.9835	121
Macro avg.	0.9584	0.9584	0.9584	151

Table 4.14: F_1 scores obtained with post-processed labels for *main unit*.

We observe that for the *main unit* task, the F_1 score improves both for *main* and *secondary* labels, resulting in an overall gain of 0.0242 for the *macro-averaged* F_1 score. This is as a consequence, in particular, of a gain of 0.1 points in the *recall* of the *main* label (with a smaller reduction of 0.282 points in *precision*), and a more modest improvement, of 0.0235 points, in the *precision* of the *secondary* label (with a smaller reduction of 0.082 points in recall).

Parent attachment

Tables 4.15 and 4.16 show the evaluation of the original and fixed predictions for the *parent attachment* task.

Label	P	R	F_1	Number
back	0.8205	0.7191	0.7665	89
forw	0.7692	0.9375	0.8451	32
none	0.8938	0.9099	0.9018	222
Macro avg.	0.8278	0.8555	0.8378	343

Table 4.15: Original F_1 scores obtained with predicted labels for *parent attachment*.

Label	P	R	F_1	Number
back	0.8072	0.7528	0.7791	89
forw	0.7368	0.8750	0.8000	32
none	0.9009	0.9009	0.9009	222
Macro avg.	0.8150	0.8429	0.8267	343

Table 4.16: F_1 scores obtained with post-processed labels for *parent attachment*.

For the *parent attachment* task, in turn, we observe an improvement of 0.0367 in the *recall* of the *back* label (with a smaller decrease in *precision*), while the changes for the majority class *none* are not significant.

Overall, the *macro-averaged* F_1 score for the post-processed labels is 0.0111 points below the original predictions. This is due, in part, to a decrease in the *recall* of *forw* labels, which can be explained by the priority given to the assignment of *back* labels in the post-processing stage. As the number of *forw* relations is of less than 10% in the consensus annotations, this does not have a very significant impact on the overall scores.

In summary, we observe that the simple heuristics implemented in the post-processing stage, aimed at ensuring the well-formedness of the predicted argumentative structure of the abstracts, contribute, as a side effect, to improve the prediction of the *main unit* of the trees. As a consequence of the changes operated in the links between the nodes–in order to avoid cycles or nodes with multiple parents–they improve the prediction of *back* relations, while have a slightly negative impact on the prediction of the less frequent *forw* labels.

4.8 Conclusions

In this chapter we addressed the prediction of the argumentative structure of computational linguistics abstracts–in particular, those included in the SciARG-CL corpus described in Chapter 3, focusing on argumentative types and relations at the sentence level.

We described our approach to model the identification of the abstracts' argumentative structure by means of four related tasks, aimed at predicting the types of the components (*unit type* task), the existence and direction of the links between them (*parent attachment* task), the type of the relations (*relation type* task), and the identification of the *main unit* of the abstract, the one that conveys the most relevant information.

We briefly introduced BERT (Devlin et al., 2019), the base architecture used in our experiments, and described our experimental setup–for single and multi-task settings. We motivated our decision to train all the tasks with a unified input format and sampling scheme, which facilitates the implementation of the multi-task settings and enables a fair comparison between the different assessed architectures. We described the loss function used in the multi-task experiments, in which the weights of the different tasks are trainable parameters that reflect the tasks' difficulty.

On the goals of this chapter was to explore the possibility of leveraging existing discourse-level annotations for the identification of the argumentative structure of scientific abstracts. We proposed to do this by introducing an intermediate training phase with sentence-level discourse tasks. We used the annotations available in the SciDTB corpus (Yang and Li, 2018) to fine-tune, in a multi-task setting, the SciBERT encoder (Beltagy et al., 2019) with *parent attachment* and *discourse relation type* tasks at the sentence level.

We commented on the difficulties involved in performing model selection and described the method we implemented to do it, which is based on plotting the loss function as a function of the number of training epochs and identifying candidate *elbows* in the graph. We trained the models with the different architectures and training strategies considered, and evaluated their predictions. In order o minimize potential errors introduced by the model selection method–in case the se-

lected epoch does not produce a particularly well-performing model–and/or by artifacts resulting from the fact that we are using a rather small set for validation, we proposed to compare the different settings by averaging the results obtained with five model checkpoints that include the *elbow-epoch* model as well as the models obtained two epochs before and two epochs after it during training.

We compared the performance obtained in single and multi-task setting with models trained with and without including an intermediate fine-tuning stage. In order to assess the impact of the supplementary training stage with different-sized training sets, we considered models obtained when using the whole SciARG-CL training set and also 50% and 25% splits of it.

We observed that the performance of the models trained with the STILT strategy performed, in general better, than those in which the base encoder–SciBERT–is directly fine-tuned with the target tasks. We found this to be the case, in particular, for tasks involving the prediction of *relations*, confirming our initial hypothesis with respect to the potential of leveraging discourse relations to improve the prediction of the argumentative structure of abstracts. We also observed that the gains in performance increases significantly as the size of the training set decreases.

As a result of our experiments, we also observed that SciARG tasks tend to perform better when trained jointly in a multi-task setting. This training strategy also presents additional benefits, including a reduced fine-tuning time, and the fact that we obtain a single encoder as a result of the process, thus facilitating the joint prediction of labels for the four sub-tasks. These considerations led us to decide to use this setting to train the models that are to be used in downstream applications (for instance, in Chapter 9).

Finally, we proposed a set of heuristics to secure the well-formedness of the predicted argumentative structures, based on a series of transformations that take into account the predictions obtained for the *main unit* task, as well as the most frequent structures observed in the gold annotations. We evaluated how these transformations impact on the performance of the *main unit* and *parent attachment* tasks in the validation set, observing that, in fact, they contribute to improve the prediction of the *main unit* and the *backward* relations, while there is a slight decrease of performance in *forward* relations.

Chapter 5

IDENTIFYING INTRA-SENTENCE ARGUMENTATIVE UNITS AND RELATIONS

In Chapter 3 we take into consideration antecedents in the identification of the rhetorical organization of scientific texts–such as (Teufel et al., 1999; Liakata et al., 2012; Kirschner et al., 2015a), as well as our own findings–including the statistics obtained from the annotation of the SciARG-CL corpus–to motivate our decision to address the identification of the rhetorical/argumentative[1] roles of sentences and intra-sentence segments as separate tasks, each one with its own challenges and ways of better approaching them.

In Chapter 4 we focus on the identification of the global argumentative structure of computational linguistics abstracts, taking the sentence as argumentative unit. In this chapter we focus on the identification of rhetorical/argumentative units within sentences, their types, and the relations between them.

[1] In most of this chapter we use the terms *rhetorical* and *argumentative* interchangeably when we refer to the functions of units and when we refer to the complexity of sentences. The same applies to the terms *rhetorical move* and *unit type* or *argumentative type*. In general–although not always–when we refer to MAZEA units we use the term *rhetorical move*, which is the term used in the original paper and related literature, and when we refer to SciARG-CL we keep the terminology used in other chapters of the thesis and refer to *unit type*.

Our decision to divide the two tasks brings with it the need to distinguish the cases in which one or the other is to be addressed: one of the first goals, then, is to identify the cases in which a sentence contains a level of rhetorical complexity–not necessarily syntactic (Lu et al., 2020)–that makes it necessary to further investigate its potentially multiple argumentative functions. Here again, different levels of analysis can considered, depending on the needs arising from different applications: if we are interested in studying, for instance, the *soundness* dimension of abstracts, we can accept that a text that includes *observations* (i.e: *hard* evidence) is argumentatively *stronger* than another one that only provides a high-level interpretation of the *results* without further evidence about how they were obtained. For this application, therefore, it might not be necessary to identify the specific boundaries of each type of unit and to make the relation explicit; it can be enough to know whether these different types of information are present or not. Other applications–such as abstractive summarization, fact extraction, etc.–might require to identify precisely the spans of text in which each type of information is communicated, and even other applications–including a fine-grained analysis of the argumentative structure of texts–might require to identify how these different parts are logically connected. In this Chapter we address these different levels of analysis.

In the same way in which, in Chapter 4 we investigate the application of a STILT-learning approach (Phang et al., 2018) to leverage sentence-level discourse information, in this chapter we continue exploring potential benefits obtained by means of including intermediate supplementary tasks in the training process. We focus, in particular, on the possibility of exploiting existing rhetorical-level annotations–similar to those used in AZ (Teufel et al., 1999) or CoreSC (Liakata et al., 2012)–available in the MAZEA corpus of scientific abstracts (Dayrell et al., 2012).

This chapter is organized as follows:

- In Section 5.1 we describe the MAZEA corpus.
- In Section 5.2 we propose a set of sentence-level tasks aimed at identifying the presence of different types of units within a sentence, as well as its *rhetorical/argumentative complexity*–in terms of the number of units of different types that it contains. We train and evaluate models for these tasks both in MAZEA and SciARG-CL.

We consider different strategies to determine the complexity of a sentence: i) to predict it directly by means of a sequence classifier, ii) to consider the number of the predicted type of units contained in it, and iii) a combination of both.

- In Section 5.3 we address the identification of intra-sentence unit boundaries and their labels. We focus, in particular, on the prediction of rhetorical/argumentative types of intra-sentence units–both in MAZEA and SciARG–and, in the last part of the section, we briefly address the prediction of intra-sentence relations in SciARG. The core of this section is dedicated to assess potential benefits obtained by implementing classification pipelines in which, given a sentence, we first predict its rhetorical/argumentative complexity, and then, depending on whether the sentence contains one or more than one unit, we process it by means of sequence-level or token-level classifiers, respectively. The sentence complexity is determined by the classification methods analyzed in Section 5.2. For sentences containing more than one unit, we compare the results obtained in two scenarios to predict the boundaries of its components and their types: in the first scenario we predict the boundaries and labels jointly, by means of a token classifier, while in the second scenario we first predict the boundaries by means of a token classifier and, in a subsequent step, we predict the labels of the identified units by means of sequence classifier. We evaluate all the proposed experimental settings in MAZEA and SciARG-CL.

5.1 The MAZEA corpus

The MAZEA (Multi-label Argumentative Zoning for English Abstracts) corpus (Dayrell et al., 2012) includes 1,335 scientific abstracts from two different disciplines: 645 abstracts from physical sciences, computing and engineering (PE) venues and 690 from life and health sciences (LH) venues. Each sentence in MAZEA can be segmented in any number of units, and each one is assigned a label corresponding to one of six rhetorical moves described in Table 5.1.

Rhetorical move	Description
background	The context of the study, including any reference to previous work on the topic, relevance of the topic and main motivations behind the study.
gap	Any indication that the researched topic has not been explored, that little is known about it, or that previous attempts to overcome a given problem or issue have not been successful.
purpose	The intended aims of the paper or hypotheses put forward.
method	The methodological procedures adopted as well as the description of the data/materials used in the study. Specifications of the structure of the paper.
result	Main findings or, in some cases, indication that the findings will be described or discussed; discussion or interpretation of the findings, which includes any hypothesis raised on the basis of the findings presented in the paper.
conclusion	General conclusion of the paper; subjective opinion about the results; suggestions and recommendations for future work.

Table 5.1: Rhetorical moves used in the MAZEA corpus.

It can be observed that, even if the semantics of some of MAZEA *rhetorical moves* and SciARG's fine and coarse-grained *types of units*[2] overlap, there is not an exact correspondence between them. For instance, there is no distinction in MAZEA between evidence obtained from observed data and its interpretation: in both cases the units are labeled as *result* in MAZEA, while they would be categorized, respectively, as *observation* or *result* in SciARG. There are also no distinct moves in MAZEA to distinguish between procedures that are part of the contributions of the described work (which, in SciARG we would label as *proposal-implementation*) from those that are simply used in the process and which could be replaced by others without significant impact on the overall proposal (in SciARG these units would be labeled as *means*).

Even when MAZEA is specifically aimed at analyzing the occurrence of multiple rhetorical moves within sentences in scientific abstracts, most of the sentences are annotated with a single label. Multi-label sentences account for 16.5% of all LH sentences and for 11.3% of all PE sentences. Coincidentally, in SciARG's ab-

[2]Described in Chapter 3.

stracts also 11.3% of sentences are assigned more than one type. We can hypothesize that, even with the differences in the granularity levels between MAZEA and SciARG's labels, results of experiments with MAZEA[3] can be a good approximation of what we can expect the results of the same experiments to be in SciARG.

The authors of the MAZEA corpus use it to train and evaluate multi-label classifiers aimed at identifying rhetorical moves contained in a sentence, independently of their order and boundaries. They use a combination of lexical, positional and grammatical features (52 features in total) that they use to evaluate two multi-label classification algorithms that work on top of single-label classifiers: i) a *classifier chain* (Read et al., 2009), which combines binary classifiers for each individual label in a chain structure, and ii) the *random k-label sets* (RAKEL) algorithm (Tsoumakas and Vlahavas, 2007), which predicts combinations of single labels (by taking into account label correlations). As single-label base classifiers they use Weka's (Hall et al., 2009) implementations of support vector machines (SMO) and of the C4.5 decision-tree algorithm (J48).

To evaluate the performance of their classifiers they consider multi-label example-based accuracy (A_e), which can be defined as:

$$A_e = \sum_{i=1}^{n} \frac{|Y_i \cap \hat{Y_i}|}{|Y_i \cup \hat{Y_i}|}$$

Where n is the number of instances, and Y_i and $\hat{Y_i}$ are the sets of real and predicted labels, respectively, for the *ith* instance. In order to compute the overall accuracy, A_e, the accuracy across instances is averaged. For instance, if a sentence s contains units of two different moves, $Y_s = \{m_1, m_2\}$, and a classifier predicts $\hat{Y_s} = \{m_2, m_3\}$, the example-based accuracy for sentence s would be $\frac{|\{m_2\}|}{|\{m_1,m_2,m_3\}|} = \frac{1}{3} = 0.33$.

The authors of the MAZEA paper use half of each subset (LH, PE) to extract features[4] and the other half to train and evaluate the classifiers.[5] The best performances are achieved for the chain classifier as multi-label algorithm on top

[3] In particular, the PE subset.

[4] They extract *formulaic expressions*–recurrent combinations of words–which they use as features in the classifiers.

[5] It is not evident from the paper whether they use a cross-validation setting or training/validation subsets.

of a support vector machine classifier for single-label predictions: the highest example-based accuracies obtained are of $A_e = 0.56$ and $A_e = 0.69$ for the PE and LH subsets, respectively.

5.2 Sentence-level tasks

In this section we describe a set of tasks aimed at identifying the presence of different *types of units* / *rhetorical moves* within sentences in MAZEA and in SciARG corpora–in sub-sections 5.2.1 and 5.2.2.

We address three sentence-level interrelated tasks:

i. Prediction of all occurring rhetorical moves (i.e: for each move, predict whether it occurs in the sentence);
ii. Prediction of a single–the first occurring–rhetorical move; and
iii. Identification of whether a sentence is *rhetorically complex* (i.e., contains more than one rhetorical move) or not.

The goals of carrying out these experiments in MAZEA are two-fold: on one hand, we are interested in exploring the prediction of MAZEA annotations as intermediate tasks to train SciARG's models in a STILT-learning approach; on the other hand, implementing the prediction of intra-sentence units and their types in MAZEA allows us to evaluate our proposals on a larger and more diverse dataset, before validating these approaches in SciARG.

5.2.1 Sentence-level experiments with MAZEA

In this section we describe the experiments carried out with the MAZEA annotations for the prediction and evaluation of the three mentioned sentence-level tasks.

5.2.1.1 Prediction of all rhetorical moves in MAZEA sentences

We model the task of predicting the rhetorical moves that occur in MAZEA sentences by means of a *multi-label classifier* that combines the predictions obtained by *six move-specific binary classifiers*.

Given a sentence s and a move-specific classifier C_m, the prediction obtained for the sentence, $C_m(s)$, is either *true* or *false*, indicating whether s contains the move m or not (being m one of the moves described in Table 5.1).

The prediction of all the rhetorical moves occurring in a sentence s ($\hat{M}_s$), is the set of moves m for which the corresponding move-specific classifier C_m returns *true*:

$$\hat{M}_s = \{m \mid C_m(s) = true\}$$

Experimental setup

For each type of move we train a linear classifier on top of a BERT encoder that takes as input the contextualized representation of the [CLS] token, in the same way as described in Chapter 4.

On of the main drawbacks observed when using single-label binary classifiers to model multi-label tasks is that knowledge about relations between the different labels is ignored. For instance, in our case, the prediction of the occurrence of one *rhetorical move* in a sentence is strongly linked to the prediction of the remaining types of moves. In order to capture existing relations between the different labels we propose a multi-task architecture in which the parameters of a BERT model are shared among all the tasks.

The experimental setup is generally equivalent to the one described for multi-task classifiers in Chapter 4. The only difference is that we model the task as single-segment classification, instead of classification of pairs of sentences.[6]

We use AllenAI's SciBERT model (Beltagy et al., 2019) as the base BERT model to fine-tune and we fix the training hyper-parameters with the values used for the sentence-level classification tasks in Chapter 4, leaving the number of epochs as the only parameter to be determined. Analogously as we do in other classification tasks, we consider the training losses for each task and select the model checkpoints by means of the elbow method described in Chapter 4.

In contrast to the evaluation strategy in Chapter 4, where we use the set of consensus annotations as validation set, in this chapter we train and evaluate the models

[6]To conduct additional experiments to assess the effect of providing context by means of including a second sentence as we do in Chapter 4 seems to unnecessarily complicate the task in this case.

in *five-fold cross-validation* settings.[7] The generation of the training/validation folds is stratified with respect to whether the sentence contains one or more *rhetorical moves*. The main reason to do cross-validation in this case is that we are interested in using the same experimental setup for MAZEA and SciARG. The number of sentences with more than one type in SciARG's validation set is too small to be considered as representative of the predictions that we would obtain with unseen sentences in general in computational linguistics abstracts. This, in turn, modifies the way in which the results are reported, as explained below.

The sequence of steps in this series of experiments is described in Algorithm 1, which can be summarized as:

1. Split the dataset into five folds containing a training / validation sets;
2. Fine-tune the base model–in a multi-task setting–with each of the five cross-validation training sets, obtaining six classifiers for each training set–one for each move;[8]
3. Use the fine-tuned models to classify all the instances of the corresponding cross-validation validation set as containing (*true*) or not containing (*false*) the considered move;

Algorithm 1 Fine-tuning the six rhetorical-move classifiers and obtaining predictions for the whole dataset in a five-fold cross validation setting

$cvFolds \leftarrow$ STRATIFIEDSPLIT($mazeaAnnotations$, 5)
for $f = 1$ to 5 **do**
  $(train_f, validation_f) \leftarrow$ GETTRAINVALFOLD($cvFolds$, f)
  $allEpochsModels_f \leftarrow$ MULTITASKFINETUNE($scibert$, $train_f$)
  $predictions_f \leftarrow \emptyset$
  for $move \in \{background, gap, purpose, method, result, conclusion\}$ **do**
    $modelMove_f \leftarrow$ MODELSELECTION($allEpochsModels_f$, $move$)
    $predsMove_f \leftarrow$ PREDICTMOVE($modelMove_f$, $validation_f$)
    $predictions_f \leftarrow predictions_f \cup predsMove_f$
  end for
end for

[7] Unlike the original paper, where part of the corpus was used to compute features, we can use the whole dataset–in a cross-validation setting–for training and validation.

[8] The model checkpoints are obtained by applying the elbow method to the training loss.

In this way, we obtain, for each sentence of the dataset and for each move, a binary prediction indicating whether the move occurs in the sentence or not.

Note that, in this case, we obtain predictions for the models obtained in the *elbow epochs* for the training set in each fold. The difference with the approach followed in Chapter 4 it is that, in this case, for each task we use the whole dataset for validation.[9] In Chapter 4 we deal with a small validation set, so we consider the averaged performance of five checkpoints around the elbow epoch to reduce the possibility of picking by chance an exceptionally good–or bad–performing model. The added complexity introduced by evaluating five models for each fold does not seem to be justified in this case, as it would unnecessarily complicate the interpretation of the results.

Results and analysis

In the same line as the authors of the MAZEA paper, we consider example-based metrics to be more suitable than label-based metrics for assessing the performance of classifiers in the context of a multi-label task.

In addition to the example-based *accuracy* defined in Section 5.1, we are interested in analyzing the balance between example-based *precision* and *recall* scores for *single-move* and *multiple-move* sentences.

We compute example-based precision (P_e) and recall (R_e) as:

$$P_e = \sum_{i=1}^{n} \frac{|Y_i \cap \hat{Y_i}|}{|\hat{Y_i}|}$$

$$R_e = \sum_{i=1}^{n} \frac{|Y_i \cap \hat{Y_i}|}{|Y_i|}$$

Where n is the number of instances, and Y_i and $\hat{Y_i}$ are the sets of real and predicted labels, respectively, for the *ith* instance.

As in their label-based metrics counterparts, the example-based precision (P_e) indicates to what extent the labels predicted for a particular sentence s_i are correct, while the example-based recall (R_e) indicates whether the labels annotated in s_i are in fact predicted.

[9] Obtained with five different cross-validation-trained models.

Based on them we can compute the example-based F_1 score ($F1_e$) in the standard way, as the harmonic mean of P_e and R_e:

$$F1_e = 2 \times \frac{P_e \times R_e}{P_e + R_e}$$

One of the goals of this chapter is to assess how the different algorithms perform in the subsets of sentences that contain *one* and *more than one* rhetorical moves to determine which tasks would benefit from processing these two subsets as separate tasks. We refer to these subsets as *single-move* and *multiple-move* sentences, respectively.

In turn, we refer to algorithms that do not discriminate between single and multiple-move sentences as *rhetorical-complexity-agnostic* algorithms, while algorithms that are provided with this information are referred to as *rhetorical-complexity-aware* algorithms. The multi-label classification algorithm described in this section does not handle explicit information about the rhetorical complexity of the sentences. It is, therefore, an example of a rhetorical-complexity-agnostic algorithm.

Table 5.2, shows the performance of the multi-label rhetorical-move classification algorithm for single-move and multiple-move sentences. In Tables 5.3 and 5.4 we show the same results disaggregated by MAZEA discipline: LH and PE, respectively.

Both disciplines (LH+PE)	$\boldsymbol{A_e}$	$\boldsymbol{P_e}$	$\boldsymbol{R_e}$	$\boldsymbol{F1_e}$
Single-move sentences	0.8092	0.8272	0.8486	0.8223
Multiple-move sentences	0.7372	0.9441	0.7462	0.8085
All sentences	0.7991	0.8437	0.8343	0.8203

Table 5.2: Prediction of all rhetorical moves in MAZEA. Example-based metrics, evaluated in five-fold cross-validation.

LH abstracts	A_e	P_e	R_e	$F1_e$
Single-move sentences	0.8764	0.8909	0.9107	0.8878
Multiple-move sentences	0.7717	0.9616	0.7774	0.8358
All sentences	0.8594	0.9025	0.8891	0.8794

Table 5.3: Prediction of all rhetorical moves for the LH subset in MAZEA. Example-based metrics, evaluated in five-fold cross-validation.

PE abstracts	A_e	P_e	R_e	$F1_e$
Single-move sentences	0.7056	0.7277	0.7529	0.7213
Multiple-move sentences	0.6476	0.8984	0.6650	0.7377
All sentences	0.6997	0.7456	0.7439	0.7230

Table 5.4: Prediction of all rhetorical moves for the PE subset in MAZEA. Example-based metrics, evaluated in five-fold cross-validation.

The example-based accuracies obtained in the MAZEA original paper for feature-based classifiers when all sentences are considered[10] are of $A_e = 0.69$ and $A_e = 0.56$ for the PE and LH subsets, respectively. Our results cannot be directly compared to theirs because we use 80% of the corpus for training and 20% for evaluation in each of the five-fold cross validation splits, while, as mentioned, in the original paper 50% of the corpus is used for training and validation. We observe, nevertheless, that in line with the original results, the performance–both for single and multiple-move sentences–in the PE subset are significantly lower than those obtained in the LH subset. This can be due to a number of reasons, including the smaller size of the PE subset, the lower level of agreement obtained for the PE annotations in MAZEA (0.652 and 0.535 κ scores for LH and the PE, respectively),[11] and, particularly, the differences in the percentages of multiple-move sentences in both subsets. These differences in performance in MAZEA subsets is observed for all the experiments–and provides some hints about what we can expect to find when we conduct the experiments in SciARG-CL.[12]

[10] They do not discriminate between sentences containing one or more than one move.

[11] The MAZEA authors use Siegel and Castellan's κ (Siegel and Castellan Jr, 1988) to measure inter-annotator agreement.

[12] As PE includes *computation* among other engineering disciplines.

We do not report discipline-disaggregated results in all the cases for brevity sake, as in this chapter we are more interested in comparing differences in performance between single and multiple-move sentences, rather than in the different disciplines.

Not surprisingly, the combined classifiers perform significantly worse in terms of *recall*[13] for multiple-move sentences. This confirms our intuition in the sense that, if the prediction of multiple rhetorical moves in a sentence is important for a downstream application in which the predictions are to be used, it is not enough to look at the overall performance but, instead, it is necessary to identify *rhetorically-complex* sentences and implement methods specifically targeted at identifying the multiple moves occurring in them. In the following sections we consider these problems.

Modeling the multi-label classification problem by means of multiple binary classifiers does not ensure that every sentence is labeled with at least one rhetorical move. In fact, in the case of our classifiers, there are 210 instances (almost 2% of all instances) that are not assigned any rhetorical move. This does not seem to have a significant impact in terms of the overall accuracy, but can be a significant problem when using the prediction in downstream applications.

5.2.1.2 Prediction of first rhetorical move in MAZEA sentences

In order to deal with cases in which, given a sentence, the six move-specific binary classifiers predict *false*–i.e., the combined multi-label algorithm fails to predict a sentence as containing at least *one* rhetorical move–we train a single-label classifier RM that, given a sentence s, directly predicts a label $RM(s) = \hat{m}_s$ that corresponds one of the six moves described in Table 5.1:

$$RM(s) = \hat{m}_s \in \{background, gap, purpose, method, result, conclusion\}$$

While SciARG's annotators were asked to assign one *main type* to each sentence–based on the perceived relevance of its argumentative/discursive function(s)–this distinction is not made in the case of MAZEA annotations. We therefore consider, for this task, the prediction of the *first* rhetorical move of the sentence.

[13] And, conversely, higher in terms of *precision*.

Experimental setup

We implement the same BERT-based architecture and parameters used in previous experiments in single-task settings, leaving the number of epochs as the sole parameter to optimize, which we do based on the training loss, and we also use SciBERT as the base model to fine-tune.

We train and evaluate the models in a five-fold cross-validation setting, analogously as we do for the binary move-classifiers described in Section 5.2.1.1. In this case we use the label to be predicted (the first move in the sentence) to generate the stratified training/test splits.

Results and analysis

Table 5.5 shows the results obtained when predicting the first rhetorical move of each sentence in the MAZEA corpus by means of the single-label classifier. We use example-based evaluation metrics for this task so we can compare its performance to the multi-label predictions obtained with the combination of the six move-specific classifiers (Table 5.5).

Both disciplines (LH+PE)	A_e	P_e	R_e	$F1_e$
Single-move sentences	0.8383	0.8383	0.8383	0.8383
Multiple-move sentences	0.4635	0.9350	0.4635	0.6194
All sentences	0.7858	0.8519	0.7858	0.8076

Table 5.5: Prediction of first rhetorical move in MAZEA. Example-based metrics, evaluated in five-fold cross-validation.

As expected, the classifier that predicts a single rhetorical move performs better–in terms of example-based metrics–than the combination of the move-specific binary classifiers for *single-move* sentences ($A_e = 0.8383$ *vs.* $A_e = 0.8092$).

For completeness' sake we include the results for multiple-move sentences in the table but, as expected, it is not a suitable alternative for this subset, even if this fact is not evident just by looking at the disaggregated accuracies (where the differences are $A_e = 0.7991$ *vs.* $A_e = 0.7858$) due to the difference in size of the two subsets.[14]

[14]Note that for *single-move* sentences, $A_e = P_e = R_e$ because as we are predicting a single

5.2.1.3 *Union classifier* for MAZEA rhetorical moves

In this section we investigate the results obtained by combining the predicted rhetorical moves obtained with the single and multi-label methods described in Sections 5.2.1.2 and 5.2.1.1, respectively, which solves the problem of sentences not being assigned any rhetorical move.

Experimental setup

For each sentence s we consider the set of predicted rhetorical moves $\hat{M}_s$ obtained by the combined six move-specific classifiers, and the rhetorical move $\hat{m}_s$ predicted by the single-label classifier. We take their union as the final set of predictions:

$$\hat{M}_s^U = \hat{M}_s \cup \{\hat{m}_s\}$$

Results and analysis

We compute example-based metrics with the set of predictions obtained by the union of the single and multi-label classifiers. In Table 5.6 we report the results of the predictions obtained by the three methods to simplify their comparison. Significant gains are obtained in terms of example-based *accuracy* and F_1 when we add the predicted label obtained by the first-move classifier to the moves obtained from the combined six move-specific classifiers.

The *union classifier* produces a gain of 0.0424 *recall* points when considering the subset of multiple-move sentences, with a loss of only 0.007 *precision* points in this subset.[15]

In the case of the proposed *union classifier*, even when there is an expected loss of 0.0163 points in *precision* for single-move sentences with respect to the first-move classifier, this does not determine the overall outcome. It is noteworthy to consider that this occurs in spite of the difference in size of both subsets: there are 9,056 instances of single-move sentences *vs.* 1,476 sentences with more than one move.

---

label $\hat{m}_i$, for each instance *i* we get that either $M_i = \hat{M}_i$ (i.e., $\hat{m}_i = m_i$) or $M_i \cap \hat{M}_i = \emptyset$ (i.e., $\hat{m}_i \neq m_i$)

[15]This is relevant because, as we see in subsequent sections, when implementing methods to process the two subsets of sentences separately, it is less problematic to consider *single-move* sentences as potentially containing multiple rhetorical moves than the other way around.

All-moves classifier	A_e	P_e	R_e	$F1_e$
Single-move sentences	0.8092	0.8272	0.8486	0.8223
Multiple-move sentences	0.7372	**0.9441**	0.7462	0.8085
All sentences	0.7991	0.8437	0.8343	0.8203
First-move classifier	A_e	P_e	R_e	$F1_e$
Single-move sentences	**0.8383**	**0.8383**	0.8383	0.8383
Multiple-move sentences	0.4635	0.9350	0.4635	0.6194
All sentences	0.7858	0.8519	0.7858	0.8076
Union classifier	A_e	P_e	R_e	$F1_e$
Single-move sentences	0.8220	0.8220	**0.9098**	**0.8508**
Multiple-move sentences	**0.7677**	0.9368	**0.7886**	**0.8356**
All sentences	**0.8144**	0.8381	0.8928	**0.8487**

Table 5.6: Comparison of results obtained in MAZEA when using single and multi-label classifiers and a classifier that takes the union of predictions made by both.

Based on these observations, we conclude that the *union classifier* is the best overall option to identify rhetorical moves in the MAZEA corpus when the rhetorical complexity of the sentences is not known (i.e: when considering rhetorical-complexity-agnostic methods).

5.2.1.4 Prediction of *rhetorical complexity* of sentences in MAZEA

As mentioned, we are interested in being able to distinguish between single-move and multiple-move sentences. In this section we address the prediction of the *rhetorical complexity* of a sentence in terms of whether it contains *one* or *more than one* rhetorical moves.[16]

[16]We use the term *rhetorically complex* in this section to refer to sentences containing more than one move.

Experimental setup

We model the prediction of a sentence *rhetorically complexity* by means of a single-task sentence-classification architecture based on SciBERT with a binary linear classifier on top, that takes as input the representation of the [CLS] token and returns one_move or two_move^+ as possible labels, to indicate, respectively, that the sentence contains *one* or *two-or-more* rhetorical moves. Given a sentence s and the *rhetorical-complexity classifier* RC, therefore:

$$RC(s) = \hat{rc}_s \in \{one\_move, two\_move^+\}$$

We train and evaluate the models in a five-fold cross-validation setting and fix the hyper-parameters to the same values used to train other single-task models. The cross-validation training/test sets are generated so they are stratified with respect to the sentences' rhetorical complexity. The selection of the model is done, as in the other experiments, by means of the elbow method applied to the training loss as a function of the number of epochs.

Results and analysis

In Table 5.7 we report the performance of the *rhetorical complexity classifier*.

In addition, we compare it to the performance obtained by a classification method that considers the number of rhetorical moves predicted by the *union classifier* described in the previous section.

We define the predicted rhetorical complexity $\hat{rc}_s^U$ based on the number of predicted moves by the *union classifier* as:

$$\hat{rc}_s^U = \begin{cases} one\_move & \text{if } |\hat{M}_s^U| = 1 \\ two\_moves^+ & \text{otherwise} \end{cases}$$

Where $\hat{M}_s^U$ is the set of moves predicted by the rhetorical-moves *union classifier* for sentence s.

In this case we address a single-label classification task, so we evaluate it by means of label-based *precision*, *recall* and F_1 scores (by class and *macro-avg*).

#Rhet. moves w/union classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
one_move	**0.9394**	0.8074	0.8684
two_moves^+	0.3654	**0.6802**	0.4754
Macro-avg.	0.6524	**0.7438**	0.6719
Rhetorical complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
one_move	0.9246	**0.9596**	**0.9417**
two_moves^+	0.6772	0.5200	**0.5883**
Macro-avg.	**0.8009**	0.7398	**0.7650**

Table 5.7: Prediction of *sentence complexity* in MAZEA with, evaluated in five-fold cross-validation.

The *rhetorical-complexity classifier* performs, as expected, better in terms of the F_1 score. For both classification methods, the *recall/accuracy*[17] obtained for multiple-move sentences (sentences that belong to the two_moves^+ class) is considerably lower when compared to single-move ones, but this difference is more evident in the *rhetorical-complexity classifier*: while only 4% of single-move sentences are mis-classified as two_moves^+, mis-classification in the other direction, two_moves^+ sentences classified as one_move ones, occurs 48% of the times. The *union classifier* for rhetorical moves yields the best *recall* for multiple-move sentences: in this case, the number of multiple-move sentences mis-classified falls to 31.98%.[18]

Ensemble classifier for rhetorical complexity

In this section we explore whether it is feasible to combine both rhetorical-complexity classifications methods considered (i.e., directly predicting the rhetorical complexity of sentences, or alternatively, deducing it from the number of predicted rhetorical moves).

We are interested in, particular, in evaluating whether this contributes to improve the *recall* of multiple-move sentences, thus reducing the number of *rhetorically complex* sentences that are erroneously considered as containing only one move.

[17]Note that in the case of single-label tasks the class-specific label-based *accuracy* equals the class *recall*.

[18]Of course, the counterpart to this is that we have a decrease in terms of *precision*: 19.26% of *single-move* sentences are mis-classified as having more than one rhetorical move.

As seen in Table 5.7, this is the tendency of the *rhetorical-complexity classifier* due to the unbalance between the two sets of sentences.

We define the prediction of the ensemble-based rhetorical-complexity, $\hat{rc}_s^E$, as:

$$\hat{rc}_s^E = \begin{cases} one\_move & \text{if } |\hat{M}_s^U| = 1 \text{ and } \hat{rc}_s = one\_move \\ two\_moves^+ & \text{otherwise} \end{cases}$$

Where $RC(s) = \hat{rc}_s$ is the class predicted by the *rhetorical-complexity classifier* for sentence s, and $\hat{M}_s^U$ is the set of moves predicted by the *union classifier* for rhetorical moves–which, as seen is the best performing algorithm to obtain all the rhetorical moves present in a sentence.

We consider the new predicted class ($\hat{rc}_s^e$) to be one_move (i.e., s is a single-move sentence) only if both methods[19] predict this label.

Rhet-compl. ensemble	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{\Delta^{uc}}$	$\boldsymbol{\Delta^{rc}}$
one_move	0.9508	0.7920	0.8641	-1.54%	-16.76%
two_moves^+	0.3697	0.7486	0.4950	+6.26%	+22.86%
Macro-avg.	0.6603	0.7703	0.6796		

Table 5.8: Prediction of *sentence complexity* in MAZEA obtained by combining the predictions of the *rhetorical-move* and *rhetorical-complexity* classifiers.

Table 5.8 shows a significant gain in *recall* for multiple-move sentences with the proposed ensemble method. The table shows the differences in percentage of mis-classified instances with respect to the rhetorical-moves *union classifier* (Δ^{uc}) and the *rhetorical-complexity classifier* (Δ^{rc}): the percentage of two_moves^+ instances that are mis-classified is reduced from the original 48% obtained with the *rhetorical-complexity classifier* ($R = 0.5200$) to 25.14% ($R = 0.7486$), and it also improves *recall* for two_moves^+ sentences obtained with the rhetorical-moves *union classifier*, which mis-classifies 31.98% of two_moves^+ instances ($R = 0.6802$), 6.26% more instances than the ensemble method. The counter-part is, naturally, an increased mis-classification of single-move sentences, although lower in percentage.

[19] i.e., Counting the number of predicted moves and directly predicting the rhetorical complexity label.

5.2.2 Sentence-level experiments with SciARG-CL

In this section we apply the proposed methods to identify rhetorical-moves and the rhetorical complexity of sentences in MAZEA annotations to predict unit types in SciARG-CL in the cases in which, in addition to the *main type*, a *secondary* type was annotated.

We evaluate the results obtained from fine-tuning SciBERT directly with SciARG tasks, and also we investigate whether a STILT-learning approach, in which we add a pre-fine-tuning stage with MAZEA annotations, can contribute to improve the prediction of SciARG types, in the same line as the experiments in which SciDTB discourse-level annotations are leveraged to improve the prediction of SciARG relations.

5.2.2.1 Intra-sentence annotations in SciARG-CL

SciARG-CL[20] contains 136 sentences in which annotators identified more than one argumentative unit. This includes 70 sentences annotated with the type *result-means*. In eight cases sentences were assigned the type *result-means* and another–main or secondary–type, effectively assigning three different types.

The final segmentation and labeling of the intra-sentence units in SciARG-CL was done by the author of this work taking into consideration the discussions on the identification of multiple units within sentences held with annotators in the process of refining the annotation criteria, which is reflected in the annotation guidelines.[21] In the case of computational linguistics abstracts, given a sentence with two types of units it is, in general, straightforward to identify which part of the sentence corresponds to which type, based on the definitions of the types of units included in the guidelines (and described in Table 3.2). In cases of potential ambiguity in the determination of unit boundaries, we considered the boundaries of the elementary discourse units included in SciDTB annotation layer, under the assumption that an argumentative unit contains one or more elementary discourse units.

[20] We consider the consensus annotations in these experiments.

[21] Available at `github.com/LaSTUS-TALN-UPF/SciARG/blob/main/Annotation_Guidelines_Arguments_SciDTB.pdf`

As a general rule, we did not divide contiguous segments of text with the same argumentative type within a sentence, as this would blur the frontiers between our argumentation-oriented annotations and more general discourse parsing annotations, such as those included in SciDTB. The only contemplated exception to this is where there is a visual separation of elements in the sentence (for instance, in an enumeration with numbered items). In the annotated texts this was observed only in two cases. We did consider the possibility of identifying discontinuous segments of the same type in a sentence, but, again, this was find to occur very rarely: in our final annotation there are only 13 sentences containing discontinuous units. As a result of the this process, 295 intra-sentence units were finally identified (only 23 units more than what we would get if all of the sentences contained only two units).

Units corresponding to the type annotated as *main type* were considered as *root* units within the sentence. In practically all of the cases, this determines the direction of the relations.[22] The assignment of the types of the relations (the argumentative function of the units) was done considering the table of *most frequent relations*, included in the annotation guidelines. This implies, for instance, that in the cases in which sentences were labeled as *result-means*, the unit corresponding to the *result* is considered as the parent, while the segment corresponding to the *means* is considered as child and linked to the parent with a *by-means* relation.

5.2.2.2 Prediction of all types of units in SciARG-CL sentences

Analogously as we do for MAZEA rhetorical moves, we train type-specific classifiers to predict whether a given fine-grained type of unit is present in a sentence in SciARG-CL. We jointly train ten binary classifiers: one for each of SciARG's fine-grained atomic types included in Table 3.2 in Chapter 3. Even when in the current SciARG annotations each sentence can include at most three atomic types,[23] we do not force this constraint in the classifier.[24]

[22] As there is, in general, only one unit of each type and very few sentences include more than two units.

[23] In case one of the types is *result-means*.

[24] In order to keep the generality of the method.

Experimental setup

We train ten SciARG-CL type classifiers in a multi-task setting with the same architecture and hyper-parameters that we use for MAZEA rhetorical moves classifiers. We also train and evaluate each classifier in a five-fold cross-validation setting.

A minor difference with respect to MAZEA experiments is that in the case of SciARG-CL the cross-validation splits are not stratified with respect to the class. Instead, we manually split similarly-sized training/test splits so all the sentences of an abstract are included in the same sets. The same training/validations splits in all the experiments, including the prediction of relations, which are established between units both within and across sentences, and require all the potentially linked sentences to be in the same training or validation set. Using the same training sets across all the tasks facilitates identifying which set of sentences are used to train a model that is then used to predict labels of another sentence. This is important, in particular, for the implementation of the pipelines described in Section 5.3, where we need to make sure that none of the models that are used in any of the steps of the pipelines have seen a sentence that is used for evaluation.

For the STILT-based experiments we pre-fine-tune SciBERT to predict the six rhetorical types in MAZEA with the same multi-task architecture described in Section 5.2.1.1, using the whole MAZEA dataset. We select the checkpoint to be used for further fine-tuning by means of the training loss elbow, as in the other experiments.

Results and analysis

In Table 5.9 we compare the results obtained for the multi-label classification based on the ten type-specific classifiers when fine-tuning SciBERT directly, and when including an intermediate fine-tuning stage with MAZEA rhetorical moves. We use the same example-based metrics described in Section 5.2.1. The models pre-fine-tuned with MAZEA produce slightly better results–in terms of example-based accuracy–than directly fine-tuning SciBERT with SciARG tasks. In particular, this is the case for sentences containing more than one type of unit (*multiple-type sentences*). The non-STILT approach produces a slightly better F_1 score for sentences labeled only with one type of unit (*single-type sentences*), due to a higher *recall* in this subset.

Results in MAZEA and SciARG-CL are not directly comparable beyond a qualitative analysis. Having this in mind, we can observe that the example-based metrics obtained for SciARG-CL are lower than those obtained for MAZEA when we consider both domains (Table 5.2).

While in MAZEA the example-based accuracy obtained with the multi-label classifier for the whole set of sentences is $A_e = 0.7991$, in SciARG, with the STILT-trained models, we obtain $A_e = 0.7121$. This is expected considering that in SciARG we have a significantly smaller dataset (1,199 instances vs. 10,532 in MAZEA) and that we have ten potential fine-grained types in SciARG while in MAZEA there are only six rhetorical moves.

If we compare the accuracies obtained for the PE subset in MAZEA, which includes documents in disciplines more similar to those in SciARG-CL, we observe that the differences are reduced (Table 5.4). In fact, the overall example-based accuracy for SciARG-CL ($A_e = 0.7121$) is higher than the one obtained in the PE subset ($A_e = 0.6997$), although when we focus particularly on multiple-move/type sentences, better accuracies are still obtained in MAZEA's PE subset ($A_e = 0.6476$ in MAZEA's PE vs. $A_e = 0.5766$ in SciARG-CL).

SciBERT model	A_e	P_e	R_e	$F1_e$
Single-type sentences	0.7269	0.7696	**0.7582**	**0.7372**
Multiple-type sentences	0.5650	0.8043	0.5956	0.6525
All sentences	0.7085	0.7736	**0.7398**	0.7276
STILT w/MAZEA all-moves classifiers	A_e	P_e	R_e	$F1_e$
Single-type sentences	**0.7294**	**0.7808**	0.7460	0.7349
Multiple-type sentences	**0.5766**	**0.8647**	**0.5980**	**0.6772**
All sentences	**0.7121**	**0.7907**	0.7292	**0.7283**

Table 5.9: Combined predictions for the ten SciARG unit types. Comparison between results obtained when fine-tuning SciBERT with a STILT-approach with an intermediate fine-tuning stage with MAZEA rhetorical moves.

5.2.2.3 Prediction of *main type* in SciARG-CL sentences

Following the same strategy applied for the prediction of MAZEA rhetorical moves, in this section we train a single-label classifier to predict the *main type* in SciARG annotations.

The labels to predict are the fine-grained types described in Chapter 3. We follow the same experimental design described for the prediction of the first rhetorical move in MAZEA, in Section 5.2.1.2. The only difference is that, as in the rest of the experiments in this section, we compare the results obtained with and without a STILT-learning approach. In this case, the intermediate supplementary task considered is the prediction of MAZEA first rhetorical move.

Experimental setup

We implement for this task the same architecture that we use throughout the rest of the experiments: the [CLS]token representation obtained with a BERT-based encoder is fed into a single-label linear classifier that predicts one of the 11 SciARG fine-grained types: the ten atomic SciARG types plus the combined type *result-means*.

We use the same five training/validation folds used throughout all the SciARG-CL experiments, as described in Section 5.2.2.2. For each fold we train models fine-tuning SciBERT and, for the STILT-learning experiments, we consider a model checkpoint trained with the whole MAZEA corpus and selected by means of the elbow method applied to the training loss. All the training hyper-parameters are fixed as in the rest of the experiments.

Results and analysis

We observe (Table 5.10) that the models pre-fine-tuned with MAZEA annotations perform, in all cases, better than when fine-tuning SciBERT directly. It is relevant to note that the most significant gain is obtained for the *recall* score of multiple-type sentences. We can reasonably expect this classifier to contribute to the identification of multiple types when its predictions are combined with the predictions obtained by means of the multi-label classifier described in Section 5.2.2.2.

Note that, in the case of SciARG, the combined type *result-means* is one of the potential predicted labels. This means that, in fact, this classifier can predict two types for this specific case. For the example-based evaluation metrics, sentences labeled as *result-means* are considered to include both *result* and *means* atomic types.

SciBERT model	A_e	P_e	R_e	$F1_e$
Single-type sentences	0.7399	0.7399	0.7498	0.7432
Multiple-type sentences	0.4988	0.7978	0.5025	0.5990
All sentences	0.7125	0.7465	0.7217	0.7268
STILT w/MAZEA first-move task	A_e	P_e	R_e	$F1_e$
Single-type sentences	**0.7498**	**0.7498**	**0.7592**	**0.7529**
Multiple-type sentences	**0.5294**	**0.8088**	**0.5368**	**0.6257**
All sentences	**0.7248**	**0.7565**	**0.7339**	**0.7385**

Table 5.10: SciARG *main type* task predictions. Comparison between results obtained when fine-tuning SciBERT with a STILT-approach with an intermediate fine-tuning stage with MAZEA's first rhetorical move.

5.2.2.4 *Union classifier* for SciARG-CL unit types

In the same way as we do in Section 5.2.1.3 for MAZEA rhetorical moves, in this section we consider a classification method for SciARG types of units that takes into account the predictions obtained both by the combined ten type-specific classifiers as well as the main-type single-label classifier.

Experimental setup

The implementation of the *union classifier* is the same as the one described in Section 5.2.1.3 for MAZEA: for each sentence *s* we consider the union of the type $\hat{t_s}$ predicted by the main-type classifier described in Section 5.2.2.3 and the combined output of the jointly trained type-specific classifiers described in Section 5.2.2.2 ($\hat{T_s}$):

$$\hat{T_s^U} = \hat{T_s} \cup \{\hat{t_s}\}$$

We compare, again, the results obtained when directly fine-tuning SciBERT, and when using the corresponding MAZEA tasks for intermediate pre-fine-tuning.[25]

[25]For SciARG's multi-label classifier we use the encoder obtained by jointly fine-tuning MAZEA's six move-specific classifiers, while for the single-label classifier we use the encoder obtained by fine-tuning MAZEA's *first-move* classifier.

Results and analysis

Table 5.11 shows that the *union classifier* based on single and multi-label classifiers pre-trained with MAZEA tasks improves the prediction of SciARG's types of units both for single-type and multiple-type sentences, with a more marked difference in the case of the latter.

The most significant gain is observed in terms of *recall* for both subsets of sentences. While there is an expected decrease in terms of *precision* with respect to the multi-label classifier considered alone, it is not so significant and, therefore, best accuracies and F_1 scores are obtained in all cases.

As expected, considering that this was the case for the individually-considered classifiers, the STILT-based *union classifier* also perform also better than the models trained without an intermediate fine-tuning stage.

	SciBERT model				**STILT w/MAZEA tasks**			
All-types classifier	A_e	P_e	R_e	$F1_e$	A_e	P_e	R_e	$F1_e$
Single-move sent.	0.7269	0.7696	0.7582	0.7372	0.7294	**0.7808**	0.7460	0.7349
Multiple-move sent.	0.5650	0.8043	0.5956	0.6525	0.5766	**0.8647**	0.5980	0.6772
All sentences	0.7085	0.7736	0.7398	0.7276	0.7121	**0.7907**	0.7292	0.7283
Main-type classifier	A_e	P_e	R_e	$F1_e$	A_e	P_e	R_e	$F1_e$
Single-move sent.	0.7399	0.7399	0.7498	0.7432	0.7498	0.7498	0.7592	0.7529
Multiple-move sent.	0.4988	0.7978	0.5025	0.5990	0.5294	0.8088	0.5368	0.6257
All sentences	0.7125	0.7465	0.7217	0.7268	0.7248	0.7565	0.7339	0.7385
Union classifier	A_e	P_e	R_e	$F1_e$	A_e	P_e	R_e	$F1_e$
Single-move sent.	0.7344	0.7344	0.8137	0.7600	**0.7535**	0.7535	**0.8175**	**0.7744**
Multiple-move sent.	0.6158	0.7984	0.6838	0.7105	**0.6446**	0.8370	**0.6875**	**0.7314**
All sentences	0.7209	0.7417	0.7990	0.7543	**0.7412**	0.7630	**0.8028**	**0.7695**

Table 5.11: Comparison of results obtained in SciARG when using single and multi-label classifiers and an *ensemble* classifier that considers the predictions made by both.

5.2.2.5 Prediction of *argumentative complexity* of SciARG-CL sentences

In this section we address the identification of sentences in SciARG as *argumentatively complex*, considering whether they contain *one* or *more than* one atomic types.

Experimental setup

We implement two classification methods analogous to the ones used to determine the *rhetorical complexity* of MAZEA sentences:

i. A binary *argumentative-complexity classifier* AC that returns whether a sentence s contains *one* or *more than one* type:

$$AC(s) = \hat{ac}_s \in \{one\_type, two\_types^+\};$$

ii. A argumentative complexity of a sentence s based on the number of types $\hat{T}_s^U$ obtained by means of the *union classifier* for SciARG types of units:

$$\hat{ac}_s^U = \begin{cases} one\_type & \text{if } |\hat{T}_s^U| = 1 \\ two\_types^+ & \text{otherwise} \end{cases}$$

We also consider an *argumentative-complexity ensemble classifier*, which combines both classification methods, analogous to the ensemble classifier implemented to predict the rhetorical complexity of sentences in MAZEA, described in Section 5.2.1.4. We therefore define the ensemble-based argumentative-complexity prediction, $\hat{ac}_s^E$, as:

$$\hat{ac}_s^E = \begin{cases} one\_type & \text{if } |\hat{T}_s^U| = 1 \text{ and } \hat{ac}_s = one\_type \\ two\_types^+ & \text{otherwise} \end{cases}$$

Where $\hat{ac}_s$ is the class predicted by the *argumentative-complexity classifier* AC for sentence s, and $\hat{T}_s^U$ is the set of types predicted by the *union classifier* for SciARG types.

As in the previous experiments in this section, the performances of STILT-trained and directly fine-tuned models are compared, and for the intermediate fine-tuning tasks, the equivalent tasks in MAZEA are considered.

Results and analysis

Table 5.12 shows the results for the *sentence complexity* task obtained with all the considered classification schemes. Analogously to what we observe in the case of MAZEA, the classifiers specifically trained to predict the sentence complexity in SciARG-CL perform better in terms of label-based F_1 scores (both for each

class and the average) than the other considered alternatives. In this case the STILT and non-STILT trained models perform, overall, similarly, yet the non-STILT classifier yields a better *recall* score for multiple-type sentences, which is reflected in the final F_1 score.

	SciBERT model			**STILT w/MAZEA tasks**		
Types w/union classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
one_type	0.9459	0.8053	0.8699	0.9485	0.8495	0.8963
two_types^+	0.2959	0.6397	0.4047	0.3522	0.6397	0.4543
Macro-average	0.6209	0.7225	0.6373	0.6504	0.7446	0.6753
Arg. complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
one_type	0.9356	0.9567	0.9460	0.9310	**0.9643**	**0.9473**
two_types^+	0.5893	0.4853	**0.5323**	**0.6122**	0.4412	0.5128
Macro-average	0.7624	0.7210	**0.7392**	**0.7716**	0.7027	0.7301
Arg. complexity ensemble	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
one_type	0.9497	0.7808	0.8570	**0.9599**	0.8325	0.8917
two_types^+	0.2831	0.6765	0.3991	0.3574	**0.7279**	0.4794
Macro-average	0.6164	0.7286	0.6281	0.6586	**0.7802**	0.6856

Table 5.12: Prediction of argumentative complexity in SciARG-CL, evaluated in five-fold cross-validation. Best F_1 scores in bold, best *precision* and *recall* in italics.

If we focus on the *recall* (or *class-accuracy*) for multiple-type sentences, we observe, as is also the case in MAZEA, that the ensemble classifier obtained by combining both classification methods performs significantly better than the other options and, in particular, than the *argumentative-complexity classifier* alone.

The combination of the STILT-trained classifiers perform significantly better for this metric than the one obtained with the non-STILT models ($R = 0.7279$ vs. $R = 0.6765$), which represents over 5% improvement in accuracy for this class.

5.3 Intra-sentence tasks

In this section we address the identification of intra-sentence rhetorical moves in MAZEA and argumentative unit types and relations in SciARG-CL. We propose two alternative methods:

1. Token-level classifiers that jointly predict unit boundaries and task-specific labels (*move/unit type*, *relation type*, *parent attachment*) for all the sentences, without taking into account whether they contain one or more than one unit.
2. Classification pipelines that include two steps:
    i. Classification of sentences according to the number of moves/units that they contain (one or more than one);
    ii. Identification of units by means of separate classifiers for sentences containing one or more than one unit.[26]

We compare the results obtained with both methods for sentences containing one and more than one unit, and assess our initial hypothesis with respect to the benefits obtained by considering the level of rhetorical/argumentative complexity of sentences in order to determine the best way to process them.

5.3.1 Token-level base architecture

All the token-level tasks described in this section share the same basic architecture: a BERT-based encoder with a linear classifier on top. The only significant difference with respect to the sequence-level architecture described in Chapter 4, is that the linear classifier predicts labels for the encodings of all the input tokens, instead of considering only the pooled representation of the sequence (the representation of the `[CLS]`token). The predictions obtained for BERT special tokens (`[CLS]`, `[SEP]`, `[PAD]`) are ignored when computing the loss (as in the rest of the experiments, we use *cross-entropy* as loss function).

[26]Based on the idea that, while it is necessary to include boundary-detection as part of the classification task for sentences containing more than one unit, this is not necessary for sentences which we know contain only one unit type.

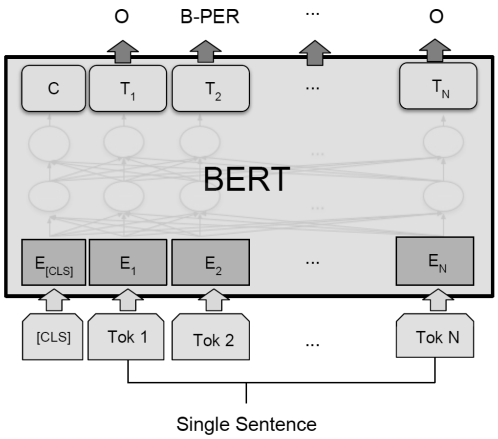


Figure 5.1: Token-level classification in BERT. Source: (Devlin et al., 2019).

5.3.1.1 Input format and tokenization

For all the token-based tasks, the unit spans are indicated in the input text by means of opening and closing tags, with or without indicating the span type, depending on the task being considered (e.g., unit type). For instance, for the identification of argumentative unit boundaries without any other information, the tags used are (`<adu>`, `</adu>`). These tags are assigned new special tokens (`[ADU]`, `[/ADU]`), which are added to the tokenizer.[27]

After the tokenization process, the corresponding labels are assigned to each token using the BIO-tagging scheme. In this process the added tokens indicating the span boundaries/types are removed.

For instance, consider the following sentence from (Landwehr et al., 2014), including the types of the units:[28]

> **<prop-imp>** We empirically study the model for biometric reader identification using eye-tracking data collected from 20 individuals **</prop-imp>** **<obs>** and observe that the model distinguishes between 20 readers with an accuracy of up to 98% . **</obs>**

[27] Otherwise, they would be considered as text and splitted as: `'[','ADU',']'`.
[28] In the example the sentence contains two units: *proposal-implementation* and *observation*.

A particular consideration in the case of BERT is that it uses the WordPiece subword tokenization algorithm to deal with infrequent words (Schuster and Nakajima, 2012).

Token	Label	Token	Label
[CLS]	-	and	B-OBS
We	B-PROPIMP	observe	I-OBS
empirical	I-PROPIMP	that	I-OBS
##ly	I-PROPIMP	the	I-OBS
study	I-PROPIMP	model	I-OBS
the	I-PROPIMP	distinguishes	I-OBS
model	I-PROPIMP	between	I-OBS
for	I-PROPIMP	20	I-OBS
bio	I-PROPIMP	readers	I-OBS
##metric	I-PROPIMP	...	...
reader	I-PROPIMP	98	I-OBS
identification	I-PROPIMP	%	I-OBS
using	I-PROPIMP	.	I-OBS
eye	I-PROPIMP	[SEP]	-
-	I-PROPIMP	[PAD]	-
tracking	I-PROPIMP	[PAD]	-
...	...	...	...
individuals	I-PROPIMP	[PAD]	-

Table 5.13: Example of tokenization and label assignment.

In the example that we are considering, the words *empirically* and *biometric* are splitted into the tokens `['empirical', '##ly']` and `['bio', '##metric']`, respectively. The hyphenated compound word *eye-tracking* is also splitted into two tokens: `['eye', 'tracking']`. We have to take this fact into consideration in the label-assignment stage for training and evaluation, so we provide the model with the correct list of labels to be predicted.

After the tokenization and label-assignment processes, we obtain the list of labeled tokens shown in Table 5.13.

5.3.2 Token-level experiments with MAZEA

In this section we describe the implementation of the two methods proposed in order to determine intra-sentence units and their rhetorical functions in the MAZEA corpus:

- *Rhetorical-complexity-agnostic* token-level classifiers, trained without distinguishing between single and multiple-move sentences (Section 5.3.2.3);
- *Rhetorical-complexity-aware* pipelines, which first classify sentences based on the rhetorical complexity and then predicts the boundaries and types of the moves that it contains by means of classifiers specialized for each type of sentence (Section 5.3.4).

5.3.2.1 General settings for MAZEA token-level experiments

As in previously experiments, we use a five-fold cross-validation setting, which allows us to obtain predictions for all the sentences in the dataset for evaluation.

In all the experiments the classifiers are trained by fine-tuning AllenAI's SciBERT encoder and the model checkpoints used to obtain the predictions are selected by means of the elbow method considering the training losses in each fold, as in previous experiments.

The same cross-validation training/test splits are used for all the experiments described in this section. The folds are generated so they are stratified with respect to: i) the sentence complexity (i.e., whether it contains one or more rhetorical moves) and, ii) the type of the first occurring move.

In sake of simplicity, we often refer in this section to *classifier* in singular. This should be understood as the combined predictions obtained with the five classifiers trained with the respective cross-validation training sets.

The sequence labeling evaluation for the token-level tasks is done by means of the *seqeval* framework.[29] We consider label-based metrics obtained with *strict* mode evaluations with the *IOB2* labeling scheme. In *strict* mode, both the boundaries and the labels should match between a real and a predicted unit to be considered as a *true positive*.

[29] `github.com/chakki-works/seqeval`

5.3.2.2 Token-level prediction of move boundaries in MAZEA

Move-boundaries classifier trained with all sentences

Experimental setup

We implement token-level classifiers for the prediction of the boundaries of rhetorical moves with the described general architecture and settings.

In this case we have, in theory, three possible labels: $L = \{\texttt{B},\texttt{I},\texttt{O}\}$ to represent a token where a new move begins, a token inside a move, and a token that is not included in any move, respectively.[30]

We first train and evaluate the classifiers using all the sentences in the dataset in a five-fold cross-validation setting as described above.

As all the sentences are used for training/validation without previously considering whether the sentences contain one or more than one rhetorical move, we refer to this classifier as *rhetorical-complexity-agnostic classifier*.

Results and analysis

Table 5.14 shows the results obtained for the whole dataset and considering the disaggregated results in the subsets of single and multiple-move sentences.

Complexity-agnostic classifier - trained with all sentences			
Move boundaries	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-move sentences	0.8997	0.9482	0.9233
Multiple-move sentences	0.5716	0.4336	0.4931
All sentences	0.8357	0.8186	0.8271

Table 5.14: Token-level prediction of rhetorical move boundaries in MAZEA, disaggregated by type of sentence. Weighted-averaged metrics at unit level with strict boundary matching.

[30] In practice, both in MAZEA and in SciARG, all tokens are included in some unit. The label *O* is therefore not assigned to any token.

The complexity-agnostic token-level classifier obtains a good performance in the prediction of boundaries for single-move sentences, as the majority of the predicted move boundaries coincide, in fact, with sentence boundaries. This, in turn, implies that the performance of this classifier for multiple-move sentences is substantially lower. Due to the difference in size of the two subsets, this fact is not evident when the aggregated set of sentences is considered.

As mentioned, in this section we are interested in exploring the possibility of discriminating between the way in which sentences are processed depending on their rhetorical complexity. If we can predict that a sentence contains only one move there is no need to determine its boundaries.

We wonder, therefore, how the results for multiple-move sentences would change if we only used this set of sentences to train and evaluate the models.

Move-boundaries classifier trained with multiple-move sentences

Experimental setup

In this experiment we consider the results obtained when training and evaluating move-boundaries classifiers in five-fold cross-validation using only multiple-move sentences, being this the only difference with respect to the previous experimental setup.

Results and analysis

Classifier trained with multiple-move sentences			
Move-boundaries	P	R	F_1
Multiple-move sentences	0.6846	0.7300	0.7066

Table 5.15: Token-level prediction of rhetorical-move boundaries in MAZEA for multiple-move sentences. Weighted-averaged metrics at unit level with strict boundary matching.

We observe in Table 5.15 that, as expected, the predictions obtained by classifiers specifically trained with sentences containing more than one rhetorical move perform significantly better for this type of sentences than those obtained by classifiers trained with the full dataset.

The F_1 score obtained in this case (0.7066), indicates the theoretical upper-bound to which we could aspire for this subset if we could use this classifier for multiple-move sentences. Of course, this would require being able to exactly differentiate sentences based on their rhetorical complexity.

The theoretical F_1 upper-bound for unit boundaries in the case of sentences that contain only one rhetorical move is $F_1 = 1$ as, if we know that a sentence contains only one unit, we can trivially determine its boundaries.

5.3.2.3 Joint token-level prediction of rhetorical move types and boundaries in MAZEA

In this section we analyze the results obtained when predicting both the boundaries and labels[31] of rhetorical moves in MAZEA sentences. The set of potential labels is therefore:

L = {`B-BACKGROUND`, `I-BACKGROUND`, `B-GAP`, `I-GAP`, `B-PURPOSE`, `I-PURPOSE`, `B-METHOD`, `I-METHOD`, `B-RESULT`, `I-RESULT`, `B-CONCLUSION`, `I-CONCLUSION`, `O`}[32]

Experimental setup

We implement token-based classifiers analogous to the ones described in Section 5.3.2.2, where the only difference is the set of labels considered.

We evaluate two scenarios:

- Training and evaluating classifiers with all the sentences;
- Training and evaluating classifiers only considering multiple-move sentences.

As in the previous section, we refer to the first classifier, as *rhetorical-complexity agnostic classifier*.

The second classifier is the one that we would use as a second step in a pipeline if we are able to determine before the rhetorical complexity of the sentences.

[31] One of the six rhetorical moves included in Table 5.1

[32] In practice, the label `O` is not used as all the words are included in some span. This applies both to MAZEA and SciARG-CL.

Results and analysis

Tables 5.16 and 5.17 show the results obtained when training a classifier with all the sentences and with multiple-move sentences, respectively.

In this case we report weighted-averaged metrics, which consider the relative relevance of each type of rhetorical move according to their frequency.

Complexity-agnostic classifier - trained with all sentences			
Move boundaries+types	P	R	F_1
Single-move sentences	0.6840	0.7173	0.6993
Multiple-move sentences	0.5573	0.4166	0.4756
All sentences	0.6580	0.6415	0.6490

Table 5.16: Token-level prediction of rhetorical move boundaries and types in MAZEA. Weighted-averaged metrics at unit level with strict matching of boundaries and types.

Classifier trained with multiple-move sentences			
Move boundaries+types	P	R	F_1
Multiple-move sentences	0.6422	0.6122	0.6254

Table 5.17: Token-level prediction of rhetorical move boundaries and types in MAZEA. Classifier trained/evaluated with multiple-move sentences. Weighted-averaged metrics at unit level with strict matching of boundaries and types.

In the same line of reasoning as in the previous experiment, the results obtained for the classifier trained and evaluated with multiple-move sentences provides the theoretical upper-bound for this subset in a classification pipeline that could perfectly distinguish single and multi-move sentences.

5.3.3 Sequence-level prediction of rhetorical moves in MAZEA

We are interested in comparing the results obtained by means of the token-level classifier described in Section 5.3.2.3, where the rhetorical moves and their boundaries are jointly predicted, with results obtained by a pipeline in which we predict the boundaries and the types of the moves in two subsequent steps.

In this section we therefore train a single-label sequence classifier that predicts a segment's type of move assuming that we already know its boundaries.

Experimental setup

We use, for the classification of intra-sentence segments, the same sequence-label classification architecture and parameters used to predict rhetorical moves at the sentence-level, described in Section 5.2.1.2.

The only difference is that we now include, as special tokens, the position of the segment within the sentence–in addition to the position of the sentence in the abstract, which we also include when training the sentence-level models.

The classifiers are trained and evaluated only with moves from multiple-move sentences (i.e., single-move sentences are excluded from the training and validation sets).

Results and analysis

We evaluate here the results obtained in a five-fold cross-validation setting when considering the gold segmentation of multiple-move sentences into rhetorical units. In Section 5.3.4 we consider a pipeline in which we evaluate the classification of automatically segmented sequences for sentences predicted to be rhetorically complex.

We compute label-based scores for each label and report their weighted average.

Classifier trained with multiple-move sentences			
Move types (sequence-level)	P	R	F_1
Macro-avg.	0.9166	0.9163	0.9164

Table 5.18: Sequence-level prediction of intra-sentence rhetorical move types in MAZEA. Classifier trained/evaluated with multiple-move sentences, evaluated in five-fold cross-validation.

We observe a high performance in the predictions when weighted-averaged metrics are considered. This can be explained due to the fact that only three types of rhetorical moves (*method*, *result*, and *purpose*) are included in the vast majority of multiple-move sentences (89%), with sequences of type *method* being the

most frequent ones (41.4% of all the units) and the ones for which the highest scores are obtained ($F_1 = 0.9510$). In turn, units of type *conclusion* are the least frequent ones (1.4% of all the units) and, as expected, the F_1 score for this class is significantly lower ($F_1 = 0.4468$)–but this does not impact significantly in the weighted-averaged scores.

5.3.4 Rhetorical-complexity-aware pipelines for the prediction of rhetorical moves in MAZEA

In this section we compare the results obtained when predicting rhetorical-move boundaries and types by means of a *rhetorical-complexity agnostic classifier*, as described in Section 5.3.2.3, to results yielded by sequential *rhetorical-complexity-aware pipelines* in which we first predict whether a sentence is *rhetorically complex* or not. Given a sentence s:

I. We predict its rhetorical complexity ($\hat{rc}_s$) by means of the classification methods described in Section 5.2.1.4;

II. If s is predicted to contain a single rhetorical move ($\hat{rc}_s = one\_move$), we use the first-move sentence-level classifier described in Section 5.2.1.2 to predict its type. In this case, there is no need to predict the boundaries–as the move covers the whole sentence–and we can directly label all the tokens with the predicted type.[33]

III. If s is predicted as being a multiple-move sentence ($\hat{rc}_s = two\_moves^+$), we predict the boundaries and types of each move contained in s.
For this we implement two approaches:

   i. We jointly predict move boundaries and types in s by means of a token-level classifier as described in 5.2.1.2, and

   ii. We predict move boundaries and types in s in two sequential steps:

      1) We predict move boundaries in s by means of a token-level boundary classifier, as described in Section 5.3.2.2, obtaining intra-sentence segments $s_1, \ldots, s_n$, and

      2) We predict the move types of the segments $s_1, \ldots, s_n$ by means of a sequence-level classifier as described in Section 5.3.3.

[33] We need the token-level classification for comparison between the different approaches.

In the report of the results, below, we refer to the pipeline in which the boundaries and types of multiple-move sentences are predicted jointly (III.i) as *two-step pipeline*, while the pipeline in which we first predict the move boundaries and then the types (III.ii) is referred to as *three-step pipeline*.

Experimental setup

The model architectures used in this section, as well as the method to obtain model checkpoints used for making the predictions–based on the training losses–are the same as the ones described in the previous sections.

Determining whether a sentence s should be considered as rhetorically complex or not is the first step of the pipelines described in this section. In Section 5.2.1.4 we examine different methods to predict the *rhetorical complexity* of a sentence:

i. We train and evaluate a binary *rhetorical-complexity classifier* that directly predicts whether the sentence contains more than one move or not ($\hat{rc}_s$), and
ii. We implement an *ensemble rhetorical-complexity classifier* that returns a prediction $\hat{rc}_s^E$ considering both the predicted *rhetorical complexity* obtained by the binary classifier ($\hat{rc}_s$), and also the number of rhetorical moves predicted to occur in the sentence ($|\hat{M}_s^U|$).

We compare the results obtained when using both types of classifiers in first step of the pipelines.

We observe in Section 5.2.1.4 that, while the simple binary classifier ($\hat{rc}_s$) performs better overall, a much higher *recall* is obtained for rhetorically-complex sentences with the ensemble classifier (Table 5.8).

This means that, when using the prediction obtained by the *ensemble rhetorical-complexity classifier* ($\hat{rc}_s^E$) in the first step of the pipelines, significantly more sentences are to classified as two_moves^+, and, therefore passed to the prediction of their intra-sentence move boundaries and types in subsequent steps.

Conversely, when using directly the prediction of the binary *rhetorical-complexity classifier* ($\hat{rc}_s$), more sentences are being classified as one_move and, therefore, processed by the sentence-level first-move classifier in the following step.

Results and analysis

In Table 5.19 we compare the results obtained with the different configurations considered. We also reproduce the results obtained with the *rhetorical-complexity agnostic classifier* (Table 5.16) for ease of comparison.

No pipeline. Joint prediction of move bound+type for all sentences			
Complexity-agnostic classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-move sentences	0.6840	0.7173	0.6993
Multiple-move sentences	0.5573	0.4166	0.4756
All sentences	0.6580	0.6415	0.6490

Two-step pipeline (III.i). Joint pred. of move bound+type for two_moves^+ sentences			
W/Basic rhetorical-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-move sentences	**0.7939**	**0.8178**	**0.8046**
Multiple-move sentences	0.5133	0.3721	0.4298
All sentences	**0.7379**	**0.7057**	**0.7206**
W/Ensemble rhetorical-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-move sentences	0.7068	0.7736	0.7361
Multiple-move sentences	**0.5934**	0.4954	0.5384
All sentences	0.6824	0.7036	0.6914

Three-step pipeline (III.ii). Sequential pred. of move bound→type for two_moves^+ sentences			
W/Basic rhetorical-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-move sentences	0.7795	0.8139	0.7954
Multiple-move sentences	0.4900	0.3789	0.4258
All sentences	0.7191	0.7044	0.7109
W/Ensemble rhetorical-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-move sentences	0.6450	0.7486	0.6902
Multiple-move sentences	0.5726	**0.5199**	**0.5438**
All sentences	0.6265	0.6910	0.6561

Table 5.19: Comparison of token-level prediction of rhetorical unit boundaries and types in MAZEA with and without rhetorical-complexity-aware pipelines. Weighted-averaged metrics at unit level with strict matching of boundaries and types.

Processing single and multiple-move sentences into two independent sub-tasks, as done with the proposed pipelines, improves the overall results with respect to a classifier that predicts unit boundaries and types for the whole set of sentences without considering their complexity ($F_1 = 0.7206$ *vs.* $F_1 = 0.6490$).

In most scenarios, this is a consequence of the better performance of the pipeline classifiers for single-move sentences which, as seen, constitute the vast majority of sentences–in the particular case of MAZEA but also in many other datasets, including SciARG. This confirms our hypothesis in the sense that applying token-level classification indiscriminately is a sub-optimal solution when most of the rhetorical moves cover whole sentences.

Implementing a rhetorical-complexity aware pipeline can also improve significantly the prediction of intra-sentence units for multiple-move sentences. In this case, though, we need to make sure to use, in the first step of the pipeline, a classifier that is more sensitive to the mis-classification of multiple-move sentences, as is the case of the proposed *ensemble rhetorical-complexity classifier*.

Table 5.19 shows that there is no significant difference between the results obtained with *two-step* or the *three-step* pipelines for multiple-move sentences in terms of their F_1 scores. This is in line with our previous observation in the sense that the main difficulty in this task relies on the prediction of the unit boundaries. Once they are known, the classification of the unit types can be done with a high level of reliability (5.18).

We can see that the performances obtained for multiple-move sentences improve significantly with classifiers specifically trained for this type of sentences over the performance obtained with a *complexity-agnostic* classifier. Yet there is a considerable space for improvement to reach the theoretical $F_1 = 0.6254$ that would be obtained should all multiple-move sentences were correctly classified in the first step of the pipeline (Table 5.17).

As mentioned, the prediction of boundaries and types of moves in single-move sentences improve considerably when these sentences are processed by sentence-level classifiers, as with the considered pipelines. Of course, the overall performance for this type of sentences depends, as in the case of multiple-move sentences, on their being correctly identified in the first step of the process.

It is natural, then, that the pipelines using the *basic rhetorical-complexity classifier*–that yield a higher *recall* for this type of sentences–produce better results for this subset.

Finally, it is relevant to notice that the two-step pipeline with the *ensemble rhetorical-complexity classifier* improves the prediction of both single and multiple-move sentences with respect to the *complexity-agnostic classifier*, which makes this method the best candidate if a balance between the performance of both types of sentences is important.

5.3.5 Token-level experiments with SciARG

In this section we report the results of the identification of intra-sentence units and their labels in the SciARG corpus. We first consider the prediction of types of units and analyze the results obtained for SciARG of experiments analogous to the ones conducted with MAZEA for the prediction of rhetorical moves in sentences with different levels of complexity.

In the case of SciARG, and in line with experiments conducted in previous sections, we compare the results obtained when fine-tuning the SciBERT encoder directly to those obtained when including an intermediate fine-tuning considering the corresponding MAZEA tasks.

As we do in the case of MAZEA, we consider the results obtained with *complexity-agnostic* classifiers, which do not distinguish between sentences according to their predicted *argumentative complexity*, and compare them to pipelines in which sentences are first classified as containing one or more than one unit (*argumentatively simple* or *argumentatively complex*, respectively) so specially trained classifiers are used in subsequent steps, depending on the sentence complexity.

5.3.5.1 Joint token-level prediction of types and boundaries in SciARG-CL

Experimental setup

Analogously as we do with MAZEA for rhetorical moves, we train token-based classifiers for the joint prediction of SciARG's unit boundaries and types.

As in previous experiments, we use SciARG's fine-grained types described in Table 3.2 in Chapter 3. The experimental setup is the same as described in in Section 5.3.2.3, in which we train classifiers in a five-fold cross-validation scheme and select the model checkpoint to obtain the predictions by means of the elbow method applied to the training loss.

As we do in the case of MAZEA, we consider two scenarios:

i. A *complexity-agnostic classifier*, trained and evaluated with all the sentences;
ii. A classifier trained and evaluated only with multiple-type sentences.

Results and analysis

Unit boundaries+types - Complexity-agnostic class.			
SciBERT model	P	R	F_1
Single-type sentences	**0.6559**	**0.6642**	**0.6480**
Multiple-type sentences	0.2033	0.0959	0.1298
All sentences	**0.5964**	0.5417	**0.5563**
STILT w/MAZEA	P	R	F_1
Single-type sentences	0.6339	0.6548	0.6291
Multiple-type sentences	**0.2724**	**0.1644**	**0.2006**
All sentences	0.5775	**0.5491**	0.5479

Table 5.20: Token-level prediction of argumentative unit boundaries and types in SciARG with and without an intermediate fine-tuning stage with MAZEA's tasks. Weighted-averaged metrics at unit level with strict matching of boundaries and types.

We observe in Table 5.20 that, as expected, when the unit types and boundaries are predicted jointly without considering whether the sentences contain one or more units, the performance within the subset of multiple-type sentences is very low.

When the BERT model is pre-fine-tuned with MAZEA annotations, even if there is an improvement in terms of the F_1 score in this subset of over 50%, the classifier still performs poorly. In particular, in terms of the *recall* in the set of multiple-type sentences.

Unit boundaries+types - Class. w/multiple-type sent.			
SciBERT model	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Multiple-type sentences	0.2258	0.1336	0.1613
STILT w/MAZEA	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Multiple-type sentences	**0.4005**	**0.1986**	**0.2166**

Table 5.21: Token-level prediction of rhetorical unit boundaries and types in SciARG. Classifiers trained/evaluated with *argumentatively complex* sentences with and without considering an intermediate fine-tuning stage with MAZEA's task. Weighted-averaged metrics at unit level with strict matching of boundaries and types.

In turn, the improvement in the detection of intra-sentence units obtained by the transfer of learned parameters from MAZEA to SciARG, which makes the model more sensitive to the presence of multiple units within sentences, has a negative impact on *precision* in the subset of single-type sentences. This is, of course, not so relevant since, if we are able to determine that a sentence contains only one unit, we would not use a token-level classifier to determine its type.

When we train and evaluate the model only considering sentences containing more than one unit–which determines the theoretical limit that we could obtain with a perfect *sentence complexity* classifier) for this type of sentences–we observe that the performance improves with respect to the models trained with the whole dataset. Nevertheless, the best result obtained–obtained when pre-fine-tuning SciBERT with MAZEA annotations–is still low (Table 5.21).

Even if they are not directly comparable, it is important to note that, in the case of MAZEA we obtain an F_1 score of 0.6254 for sentences with more than one rhetorical type for this experiment (Table 5.17). The difference in performance in both cases is not surprising if we consider the different sizes of both datasets: while MAZEA includes 10,532 sentences, of which 1,476 (14%) contain more than one rhetorical type, we have 1,199 sentences in SciARG, of which 136 (11%) have been annotated with more than one type. In addition, while there are six possible labels in MAZEA for rhetorical moves, in SciARG we are predicting ten fine-grained types.

This is relevant because the results obtained for this task in MAZEA make it feasible to consider the implementation of the described two-step pipeline in which we first determine whether a given sentence is rhetorically complex and, in a second step, we directly predict the boundaries and types of the units that it contains. In fact, in the case of MAZEA, this pipeline performs competitively when compared to a three-step pipeline in which the boundaries of the units and their types are predicted in two successive steps (Table 5.19).

In the case of SciARG-CL, however, this does not seem like a feasible possibility. We will, therefore, only evaluate the results obtained when implementing a three-step pipeline–where the prediction of unit boundaries and types are conducted in two successive steps.

5.3.5.2 Token-level prediction of unit boundaries in SciARG-CL

As we do in the case of rhetorical moves in MAZEA (Section 5.3.2.2), we train models to predict the boundaries of intra-sentence units, which is an essential step in the proposed complexity-aware pipeline.

Experimental setup

We use the same architecture and methodology implemented for the token-level prediction of unit boundaries in MAZEA, as described in Section 5.3.2.2. In this case, based on the previous evidence obtained, and for simplicity sake, we only train and evaluate the models with multiple-type sentences.

As in the other experiments with SciARG, we consider the results obtained with and without a STILT-learning approach. The difference, in this case, is that the labels to predict in MAZEA and SciARG are the same, which allows us to not only take advantage of the encoder weights obtained in the preliminary fine-tuning stage, but also of the weights learned by the linear classifier.

Results and analysis

We observe in Table 5.22 that a significant gain is obtained for the token-level prediction of unit boundaries in SciARG-CL when leveraging MAZEA rhetorical-move boundary annotations for pre-training the model.

Unit boundaries - Class. w/multiple-type sent.			
SciBERT model	P	R	F_1
Multiple-type sentences	0.4656	0.3938	0.4267
STILT w/MAZEA	P	R	F_1
Multiple-type sentences	**0.6082**	**0.6062**	**0.6072**

Table 5.22: Token-level prediction of unit boundaries in SciARG. Weighted-averaged metrics at unit level with strict boundary matching.

5.3.6 Sequence-level prediction of intra-sentence unit types in SciARG-CL

Experimental setup

For the prediction of the argumentative types of pre-segmented units in SciARG-CL we use the same sequence classification architecture used in MAZEA and described in Section 5.3.3.

As in the other experiments conducted in SciARG-CL, we compare the results obtained with an without a supplementary fine-tuning stage. In this case we use MAZEA annotations of rhetorical moves, considering the subset of multiple-move sentences.

Results and analysis

Table 5.23 shows the results obtained for the classification of intra-sentence unit types. We compute the scores obtained for each of the ten SciARG atomic types and report the macro-averaged scores.

Unit boundaries - Class. w/multiple-type sent.			
SciBERT model	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Macro-avg.	0.6332	**0.6942**	0.6576
STILT w/MAZEA	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Macro-avg.	**0.6748**	0.6873	**0.6780**

Table 5.23: Sequence-level prediction of intra-sentence unit types in SciARG. Classifier trained/evaluated with *argumentatively complex* sentences, evaluated in five-fold cross-validation.

In Table 5.23 we observe a gain in performance when pre-fine-tuning SciBERT with MAZEA in terms of the macro-averaged precision and $F1$ scores, while both models perform almost equivalently in terms of *recall*, with a minor advantage of the non-STILT model. When looking more in detail at the disaggregated results by class we observe that this is due to the fact that the STILT-trained model does a better job at discriminating classes with fewer instances, while the non-STILT model tends to favor the majority classes–in particular, *result*, *means*, and *proposal-implementation* that, together, constitute 76% of all the intra-sentence units.

5.3.7 Argumentative-complexity-aware pipelines for the prediction of argumentative units in SciARG-CL

In Section 5.3.5.1 we explore the joint prediction of boundaries and types both when training and evaluating a token-level classifier with all the sentences, independently of their argumentative complexity, which we refer to as *argumentative-complexity-agnostic classifier*, and we also train and evaluate a classifier that is specifically targeted at sentences containing two or more units, observing that, in the case of SciARG the results obtained when jointly predicting unit types and boundaries for multiple-type sentences are too low to make it a feasible alternative for a two-step pipeline as the one considered for MAZEA in Section 5.3.4.

In this section we compare the results obtained for the identification of argumentative unit boundaries and their types in SciARG by means of the *argumentative-*

complexity agnostic classifier with the results obtained when implementing a three-step pipeline analogous to the one described for MAZEA, where unit boundaries and types are predicted in two subsequent steps for sentences that are previously identified as containing more than one type.

When a sentence is classified as containing only one unit by the argumentative-complexity classifier in the first step of the pipeline, its type, parent attachment and type of relation are predicted with a sentence-level classifier implemented as described in Chapter 4. In this case, the models are trained in a five-fold cross-validation setting–with the same training/test splits used for all the SciARG-CL experiments in this chapter–so we can obtain predictions for the whole dataset. The classifiers are trained in a multi-task setting with pairs of sentences, and considering sentence-level relations in SciDTB as intermediate task, as described in Chapter 4.

In summary, given a sentence s:

I. We predict its argumentative complexity ($\hat{ac}_s$) by means of the classification methods described in Section 5.2.2.5.

II. If s is predicted to contain a single unit ($\hat{ac}_s = one\_type$), we predict its type–as well as parent attachment and relation type–by means of a multi-task classifier implemented as in Chapter 4.

III. If s is predicted as containing more than one unit ($\hat{ac}_s = two\_types^+$), we:

   i. Predict the boundaries of each unit contained in s: $s_1, \ldots, s_n$.

   ii. Predict the type of each intra-sentence unit $s_1, \ldots, s_n$ by means of a sequence-level classifier as described in Section 5.3.6.

Table 5.24 shows the results obtained for the described pipeline when using in the first step, to determine the argumentative complexity of the sentences, i) the predictions obtained by a *basic argumentative complexity classifier* ($\hat{ac}_s$) described in Section 5.2.2.5, or ii) the predictions obtained by means of the *ensemble argumentative-complexity classifier* ($\hat{ac}_s^E$) described in the same section.

In order to facilitate the comparison, we include also the results of the joint boundary and type predictions obtained with the token-level *argumentative-agnostic classifier* described Section 5.3.5.1

No pipeline. Joint prediction of unit bound+type for *all sentences*			
Complexity-agnostic classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-type sentences	0.6339	0.6548	0.6291
Multiple-type sentences	0.2724	0.1644	0.2006
All sentences	0.5775	0.5491	0.5479

Three-step pipeline (III). Sequential pred. of unit bound→type for two_types^+ sent.			
W/Basic argumentative-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-type sentences	**0.6812**	**0.7140**	**0.6941**
Multiple-type sentences	0.3797	0.2621	0.3055
All sentences	**0.6283**	**0.6171**	**0.6210**
W/Ensemble argumentative-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-type sentences	0.6216	0.6723	0.6419
Multiple-type sentences	**0.3885**	**0.3103**	**0.3431**
All sentences	0.5763	0.5947	0.5837

Table 5.24: Comparison of token-level prediction of unit boundaries and types in SciARG with and without sentence complexity-aware pipelines. All models trained with an intermediate fine-tuning stage. Weighted-averaged metrics at unit level with strict matching of boundaries and types.

We observe that the *argumentative-complexity-aware pipelines* improve the predictions of units and its types both for single and multi-type sentences with respect to the *argumentative-complexity-agnostic*. As in the case of MAZEA, when the *ensemble argumentative-complexity classifier* is used as first step of the pipeline more sentences are classified as two_types^+ and are subsequently processed by the specialized classifier for the detection of the units contained in it. Conversely, when the *basic argumentative-complexity classifier* is used, more sentences are classified as one_type and processed by the sentence-level classifier. This can be observed, in particular, in the gains in *recall* for the respective types of sentences in each of these scenarios.

As in the case of MAZEA, using one or the other classification strategy as the first step of the pipeline would depend on the specific needs of the application where the predictions are to be used.

5.3.8 Prediction of intra-sentence relations in SciARG-CL

We consider that a sentence as a whole has an argumentative function through which it is linked to another sentence and, on a different level of analysis, we consider how the various parts of the sentence interplay from an argumentative point of view. In this section we evaluate the results obtained when combining the predicted relations at both levels, which conforms the fine-grained argumentative structure of the abstracts.

Note that we consider that each sentence is linked to another one only through one segment, the sentence *argumentative root*. Therefore, the parents of *non-root* segments can only be other units within the same sentence. In this sense, we are adopting the composionality principle of discourse-based analysis.

In this section we evaluate the performance of the argumentative-complexity-aware classification pipelines for the prediction of intra-sentence relations.

Note that, based on the results obtained for the prediction of units types in Section 5.3.7 we can discard the *complexity-agnostic classifier* as the best alternative for the prediction of units and their types within sentences–even more so for determining the relations between them. In addition, there is a practical difficulty in that we would need to model two different token-level classification tasks–prediction of relations between sentences and prediction of relations between units–in a unified way for the different models to be comparable. A way to do this would be to model the tasks at the document (abstract) level and use, to model the *parent attachment* task, token positions–or the position of the discourse units as we did in our preliminary experiments, described in Appendix A. The practical complexities that this involves[34] does not seems to be justified in this case, based on the previous results.

As observed in Section 5.3.7, in the case of SciARG-CL the only feasible alternative is to implement a three-step pipeline in which, in the first step, sentences are classified according to their argumentative complexity and, for sentences predicted to contain more than one unit, their boundaries and labels are predicted in two subsequent steps.

[34]Including, for instance, the need to consider the possibility of having input sequences longer than BERT's maximum sequence length of 512 tokens.

In the case of sentences predicted to be argumentatively not complex (i.e., classified as one_type), its parent and type of relation are predicted–jointly with its type–by means of a multi-task classifier pre-fine-tuned with SciDTB sentence-level tasks and trained in a five-fold cross-validation setting, as described in 5.3.7.

When a sentence is classified as argumentatively complex in the first step of the pipeline (i.e., classified as two_types^{+}), the boundaries of its components are obtained by means of the classifier described in Section 5.3.2.2. For each of the identified units, the relative positions of their parents within the sentence (*parent attachment* task) and their argumentative function (*relation type* task)[35] are predicted jointly by means of a multi-task pair classifier similar to the ones used to predict relations at sentence-level. In this case, the segments considered to train the models are not full sentences but intra-sentence units paired together with the same criteria applied for training the models at the sentence level and described in Chapter 4. Similarly, these classifiers are also trained with a STILT-learning approach. In this case, instead of considering discourse-relations only at the sentence level, the annotations used as intermediate pre-training task are the fine-grained discourse relations between elementary discourse units in SciDTB described in Chapter 3.

To indicate the attachment of a sentence to its parent we consider the relative position of the parent sentence within the text. Analogously, to indicate the attachment of an intra-sentence unit to its parent, we consider their respective positions within the sentence. Units for which no parents are predicted are considered to be *root* units at the sentence level. In an additional step, we label the intra-sentence *root* units with the argumentative function and parent position of the sentence as a whole. For instance, consider the following example from (Kolhatkar and Hirst, 2014), where we represent sentence-level argumentative function (the type of the relation though which it is linked to its parent) by means of the attribute `afu` and the parent attachment by means of the attribute `par`, while unit types are expressed by the tag labels (e.g., `<prop>`). At the intra-sentence level, the type of the relation and the parent position are expressed by the attributes `sent-afu` and `sent-par`, respectively. The absolute position of the sentence in the text is represented by the attribute `sent` and the unit position within the sentence by the attribute `segm`.

[35]The relation types considered are the same as the ones used at sentence level and described in Chapter 3 (Table 3.4).

<motiv-back sent=1 **segm**=1 **afu**=info-req **par**=1**>**
Shell nouns, such as fact and problem, occur frequently in all kinds of texts.
</motiv-back>

<motiv-prob sent=2 **segm**=1 **afu**=support **par**=1**>**
These nouns themselves are unspecific, and can only be interpreted together with the shell content. **</motiv-prob>**

<prop sent=3 **segm**=1 **afu**=root**>**
We propose a general approach to automatically identify shell content of shell nouns. **</prop>**

<prop-imp sent=4 **segm**=1 **afu**=elab **par**=-1**>**
Our approach exploits lexico-syntactic knowledge derived from the linguistics literature. **</prop-imp>**

<means sent=5 **segm**=1 **sent-afu**=by-means **sent-par**=1**>**
We evaluate the approach on a variety of shell nouns with a variety of syntactic expectations, **</means>**
<obs sent=5 **segm**=2 **sent-afu**=root-sent **afu**=support **par**=-2**>** achieving accuracies in the range of 62% (baseline=33%) to 83% (baseline=74%) on crowd-annotated data. **</obs>**

Note that, in the example, only the fifth sentence of the abstract contains two units, of types *means* and *observation*, respectively, being *observation* the main type.

In the pipeline, this sentence should be classified as *argumentatively complex*, the boundaries of the units contained in it would have to be determined and, in the following step, the intra-sentence parent and argumentative function of each of the predicted units would be obtained by means of intra-sentence sequence-level classifiers. We should obtain, as predictions, that the first unit (of type *means*) is a child of the second unit (of type *observation*) and, therefore, the predicted relative parent position of the first unit within the sentence is 1–indicating that the parent is one position ahead, while the label of the relation is *by-means*.

The *observation* segment (in the second position in the sentence) would be predicted as being the *root* unit of the sentence. The sentence-level parent and type of relation predicted using the full sentence[36] are therefore assigned to this unit

[36]By means of the sentence-level classifier

with the attributes `afu` and `par`. Table 5.25 shows the intermediate and final predictions at token-level for this sentence for the *relation type* task.

Token	Intermediate prediction	Final prediction
We	B-BY-MEANS	B-BY-MEANS
evaluate	I-BY-MEANS	I-BY-MEANS
the	I-BY-MEANS	I-BY-MEANS
approach	I-BY-MEANS	I-BY-MEANS
...		
expectations	I-BY-MEANS	I-BY-MEANS
,	I-BY-MEANS	I-BY-MEANS
achieving	B-ROOT-SENTENCE	B-SUPPORT
accuracies	I-ROOT-SENTENCE	I-SUPPORT
in	I-ROOT-SENTENCE	I-SUPPORT
...	I-ROOT-SENTENCE	I-SUPPORT
annotated	I-ROOT-SENTENCE	I-SUPPORT
data	I-ROOT-SENTENCE	I-SUPPORT
.	I-ROOT-SENTENCE	I-SUPPORT

Table 5.25: Example of predicted relation types at token level.

Table 5.27 shows the results obtained for the combined prediction of intra and inter-sentence relations. As in the previous experiments described in this chapter, the evaluation is done at the token-level with strict matching of boundaries and labels.

The pipeline that uses in its first step the basic *argumentative-complexity classifier* performs better for single-type sentences, and therefore also is the one for which the best overall results are obtained, while the pipeline that uses the *ensemble argumentative-complexity classifier* performs better for multiple-type sentences.

The performances obtained for the different tasks cannot be compared to each other, but we can observe that, as expected, the relative differences obtained for single-type sentences are in line with the differences observed for these tasks in Chapter 4. In the case of multiple-type sentences, once the boundaries and types of units are defined within sentences, identifying how they are linked to each other is quite straightforward. It is therefore expected that the difficulty of the prediction of relations is to some extent capped by the difficulty in the prediction of the types of units.

Three-step pipeline. Sequential pred. of unit bound→rel. type for two_types^{+} sent.			
W/Basic argumentative-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-type sentences	**0.7323**	**0.7366**	**0.7336**
Multiple-type sentences	0.3570	0.2207	0.2671
All sentences	**0.6339**	**0.6260**	**0.6288**
W/Ensemble argumentative-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-type sentences	0.7032	0.7015	0.7016
Multiple-type sentences	**0.4070**	**0.2793**	**0.3214**
All sentences	0.6102	0.6109	0.6094

Table 5.26: Token-level prediction of unit boundaries and relation types in SciARG-CL with three-step complexity-aware pipeline. Models trained with an intermediate fine-tuning stage with SciDTB relations. Weighted-averaged metrics at unit level with strict matching of boundaries and labels.

Three-step pipeline. Sequential pred. of unit bound→parent for two_types^{+} sent.			
W/Basic argumentative-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-type sentences	**0.7239**	**0.7357**	**0.7260**
Multiple-type sentences	0.3310	0.1986	0.2415
All sentences	**0.6204**	**0.6215**	**0.6167**
W/Ensemble argumentative-complexity classifier	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Single-type sentences	0.6759	0.6817	0.6752
Multiple-type sentences	**0.3453**	**0.2648**	**0.2938**
All sentences	0.5766	0.5930	0.5798

Table 5.27: Token-level prediction of unit boundaries and parents in SciARG-CL with three-step complexity-aware pipeline. Models trained with an intermediate fine-tuning stage with SciDTB relations. Weighted-averaged metrics at unit level with strict matching of boundaries and labels.

5.4 Conclusions

In this chapter we focused on the identification of intra-sentence unit boundaries and types. We compared results obtained by token-level classifiers that predict intra-sentence units without considering the level of rhetorical/argumentative complexity of the sentences, to those obtained by means of pipelines in which this information is taken into consideration in order to decide the type of classifier better suited in each case. These pipelines rely on the possibility of classifying the complexity of sentences. We therefore explored different approaches to this task, including a basic sentence-level classifier to directly predict the sentence complexity, as well as methods that include predicting the–potentially multiple–types of units contained in a sentence, and a combination of both.

The results obtained confirm our initial hypothesis in relation to the potential benefits obtained by discriminating the way in which sentences are processed depending on their rhetorical/argumentative complexity. In particular, in cases where the overwhelming majority of the sentences contain only one rhetorical move, as in the case of SciARG-CL, MAZEA and several other corpora–including AZ (Teufel et al., 1999) and CoreSC (Liakata et al., 2012).
One alternative–which is the one that we adopted in our SciARG annotation scheme and the tasks described in Chapter 4–is to prioritize the most frequent type of sentences (in this case, sentences containing only one type of unit). This, in turn, simplifies the annotation process, making it more probable to obtain more reliable annotations.
If the identification of intra-sentence units is necessary for a particular downstream application, several aspects are to be weighted, including how determining it is to get a high *recall* for these units, knowing that this will impact negatively on the classification of rhetorically simpler sentences. Depending on these decisions, different strategies for determining the complexity of the sentences and for the identification of their components can be considered, as described in this chapter.

We also continued exploring the benefits obtained by leveraging existing annotations. In this case, we considered, for supplementary training, the rhetorical-level annotations included in the MAZEA corpus of scientific abstracts (Dayrell et al., 2012) by means of a STILT transfer learning approach (Phang et al., 2018). The results obtained show that in general this strategy leads to better performances in the target tasks.

In addition to use MAZEA annotations as intermediate fine-tuning tasks, we took advantage of the availability of this corpus to validate our results. This contributed to confirm that the results obtained in SciARG-CL for the tasks considered in this chapter are not qualitatively different from those obtained in a larger corpus that covers significantly more scientific disciplines.

Chapter 6

EXTENDING SCIARG: FROM COMPUTATIONAL LINGUISTICS TO BIO-MEDICINE

The SciARG annotation scheme introduced in Chapter 3 and used in the experiments described in Chapters 4 and 5 was specifically developed to account for argumentative types and relations in computational linguistic abstracts. In this chapter we explore the applicability of this scheme to other scientific disciplines. In particular, we analyze if it can be successfully used to annotate argumentative information in biomedical abstracts.

As an additional research question, we are interested in exploring the extent to which models obtained from computational linguistics annotations capture discipline-independent knowledge. We therefore investigate the predictive potential of these models when used in abstracts that, as we analyze in Section 6.1.3, have a significantly more complex discourse structure than the texts used to train them.

To respond to these questions, we use the SciARG annotation scheme to annotate a set of biomedical abstracts and use these annotations in experiments aimed at predicting their argumentative structure.

It is relevant to note that, as our goal is to test our original annotation scheme in a scientific discipline different to the one in which it was initially developed, we are not proposing an alternative scheme specifically tailored to the analysis of discourse and/or the rhetorical classification of sentences in biomedical abstracts, which is an area that has been studied in other works. In particular, two of the most studied datasets in the biomedical area are the NICTA-PIBOSO corpus, developed by Kim et al. (2011), in which 1,000 medical abstracts were hand-annotated at sentence level by domain experts with the PICO schemes,[1] which is used in the analysis of randomized controlled trials (RCT); and the PubMed 200k RCT dataset released in 2017 by Dernoncourt and Lee (2017), proposed as a resource to train automatic sentence classifiers for unstructured abstracts. This dataset was constructed by retrieving 195,654 RCT structured abstracts from the 2016 MEDLINE/PubMed Baseline Database[2] and automatically labelling each sentence with the name of the section it belongs to. As we see in Chapter 3 is the case of corpora developed in other scientific disciplines (such as computational linguistics), the NICTA-PIBOSO corpus and the PubMed 200k RCT dataset are aimed at identifying the type of information conveyed by sentences but do not contain explicit annotations that evidence discourse or logical relations between them.

The rest of this chapter is organized as follows:

- In Section 6.1 we describe the application of the SciARG scheme to the annotation of a set of biomedical abstracts, thus generating a new annotated corpus: SciARG-BIO. This allows us to assess the applicability of our proposed annotation scheme to a new scientific discipline. We describe the annotation process, analyze the observed inter-annotator agreement, and report corpus statistics. We analyze, in particular, differences between the argumentative structures of computational linguistics and biomedical abstracts, observing a higher level of complexity and ambiguity in the latter.

[1] Which classifies text according to whether it provides information about the population/participants in the trail (P), the intervention (I) carried on, comparison (C) with other works, and the outcomes (O) of the intervention.

[2] The MEDLINE database of life sciences and biomedical information (`nlm.nih.gov/bsd/medline.html`) is maintained by the U.S. National Library of Medicine and available through the PubMed (`pubmed.ncbi.nlm.nih.gov`) search engine.

- In Section 6.2 we implement and evaluate experiments aimed at predicting argumentative components and relations in SciARG-BIO. We explore, in particular, to what extent a BERT model fine-tuned with SciARG-CL annotations embeds knowledge about the tasks being considered that makes it possible to use it directly–without further fine-tuning of the encoder parameters–in the prediction of argumentative components and relations in the biomedical domain.
- In Section 6.3 we summarize the main conclusions of this chapter and reflect on the potential need to incorporate domain experts in the annotation process for the interpretation–and disambiguation–of specific types of information contained in biomedical texts.

6.1 SciARG-BIO Corpus

In this section we describe the application of the SciARG annotation scheme developed for the generation of a new corpus of biomedical texts (SciARG-BIO). We analyze the main differences between both sub-corpora and the resulting annotations and evaluate the level of agreement obtained.

6.1.1 Data and annotation process

The annotation of the biomedical subset of SciARG (SciARG-BIO) is part of a collaboration developed with Mariana Neves, from the German Federal Institute for Risk Assessment (BfR).[3] The dataset used in these experiments includes 285 abstracts of MEDLINE/PubMed articles which had previously been used in a work aimed at the evaluation of different annotation schemes and tools used for ranking biomedical abstracts based on their textual similarity (Neves et al., 2019). In this work, 562 articles were collected and clustered based on their similarity[4] to seven initial MEDLINE documents. A stratified subset of these articles was selected for our annotations so it includes approximately the same number of documents from each cluster.

[3] `bfr.bund.de`

[4] According to PubMed's *similar articles* functionality.

In the computational linguistics documents originally included in SciARG-CL, sentences had already been identified as part of the development of the SciDTB corpus. In the case of SciARG-BIO, the segmentation process, needed to prepare the abstracts for the sentence-level annotation, is done automatically by means of the syntok tool.[5]

Two annotators participated in the annotation of the SciARG-BIO corpus, which took 8 months to complete. In this case no training phase was considered, as both annotators had already taken part in the annotation of SciARG-CL and, in fact, the experiment implied that no modifications were to be done to the annotation scheme described in Chapter 3. Likewise, no changes were made to the annotation guidelines or to the tool used. From the 285 total annotated abstracts, 50 (476 sentences) were annotated by both annotators in order to compute inter-annotator agreement.

6.1.2 Agreement

Table 6.1 shows the agreement obtained for the 50 abstracts annotated by both annotators. As in the case of SciARG-CL, we compute Cohen's κ as well as the accuracy obtained when considering one of the annotations as the gold standard.

Task	**Cohen's κ**	**Accuracy**
Fine-grained unit type	0.66	0.72
Coarse-grained unit type	0.93	0.96
Parent position	0.49	0.54
Relation type	0.43	0.58
Main unit	0.94	0.99
All combined	0.39	0.40

Table 6.1: Agreement in SciARG-BIO (Cohen's κ and accuracy)

We observe substantial agreement between annotators for the fine-grained type of the units and moderate agreement for the relations, both in terms of the labels and the parent attachments. A lower level of agreement when moving from one discipline to the other is expected, taking into consideration that i) the annotation

[5] `github.com/fnl/syntok`

scheme was designed and adjusted specifically for the CL domain, ii) while annotators have a high level of familiarity with computational linguistics texts it is not the case for the BIO domain, and iii) biomedical abstracts have a considerably higher level of complexity when compared to computational linguistics ones in terms of their structure, the number of units that they contain and their lengths, as shown in Section 6.1.3.

When analyzing discrepancies in the SciARG-BIO annotations we observe that units of types *observation* and *result* give origin to the most frequent disagreements between annotators. In fact, one annotator labeled as *observation* 64% of the units that the other annotator labeled as *result*, which makes us believe that a clear distinction between these two types is difficult to establish without specific domain knowledge. When coarse-grained types are considered, in fact, these two types are not distinguished and the level of agreement reaches 0.93 Cohen's κ, similar to the observed in SciARG-CL (0.94). This can explain, in part, the lower level of agreement in the annotation of unit types. It is, nevertheless, relevant to consider in more detail the possible reasons for the differences in agreement observed between the different tasks and, in particular, in the identification of the parent units,[6] as, in contrast to what we observe in SciARG-BIO, in SciARG-CL there was a similar level of agreement for all the main tasks.

If we consider the annotations in which the two annotators disagreed with respect to the parent attachment, we observe that in 79.3% of the cases the child unit was annotated as either *observation* or *result*, with 67.3% of the cases in which at least one of the annotators assigned the type *observation* to the unit. What might be more surprising is that in 35.9% of all the disagreements with respect to the parent, both annotators annotated the child unit as *observation* (this is more than 53% of the cases of disagreements that involve an *observation* unit). This indicates that it is considerably difficult (at least for non domain experts) to identify how the different *observations* are linked to each other and to the outcomes of multiple experiments. In fact, the frequency and role of *observation* units in biomedical abstracts constitute the main difference with respect to computational linguistics ones, as described in Section 6.1.3. For other types of units, such as those used to introduce motivations and to describe proposals, there is less ambiguity in their identification by annotators with familiarity with the scientific language but who

[6]Differences in the attachment of a unit to its parent would also explain differences in terms of assessing the argumentative role that it plays and, therefore, in the labels assigned to the relation

are not experts in the specific domain.

The difficulty that arises from the inherent subjective interpretation of argument in scientific text has already been pointed out in Section 2.3. Al Khatib et al. (2021) refer specifically to the challenge that this poses for non-expert annotators in the biological domain:

> *A common dilemma in argument mining is that an argumentative text may have multiple valid interpretations of its structure. This is a concern for scientific documents, where the connection between a claim and its evidence can be implicit, i.e., the author leaves this connection to the readersâ interpretations. In particular, experimental papers can follow a line of reasoning that makes e.g. âbiological senseâ, i.e. where a specific experiment follows another experiment to address a potential alternate interpretation of the previous experiment. For a non-biologist, this reasoning is unclear, and the reason for these subsequent results are generally never explicitly stated in the text.* (Al Khatib et al., 2021, p.58)

Another consideration to be made is that, as part of the experiment we decided to keep, in the annotation of the biomedical texts, the same annotation guidelines and criteria used for the annotation of computational linguistics abstracts where, in general, there is less ambiguity with respect to the role of *observation* units, as seen in Fig. 3.5 in Chapter 3. Therefore, it is likely that with minor modifications in the guidelines which contemplate in more detail alternative uses of *observation* and *result* units in biomedical texts, part of the ambiguity could have been cleared up. We believe, nevertheless, that the inclusion of domain experts is required for the fine-grained annotation of specific parts of the abstracts (in particular, the outcomes). We leave as future work the exploration of a hierarchical annotation process, in which *difficult* and/or *domain-knowledge-dependent* annotations are left for domain experts. The implementation of a semi-automatic annotation pipeline could make such a process more feasible in terms of time and resources.

6.1.3 Corpus statistics and analysis

In Tables 6.2, 6.3, and 6.4 we show overall statistics for the SciARG-BIO corpus, the distribution of the different types of units and relations, and the distances between children and parent units.

Statistics	**CL**	**BIO**
Number of abstracts	225	285
Total number of units	1199	2787
Avg. #units/abstract	5.3 ($\sigma = 1.7$)	9.8 ($\sigma = 3.1$)
Max. #units/abstract	13	25
Min. #units/abstract	2	2
Avg. #tokens/unit	24.4 ($\sigma = 9.9$)	30.1 ($\sigma = 14.2$)
Max. #tokens/unit	101	155
Min. #tokens/unit	5	5
Forward relations	32%	34%
Backward relations	68%	66%

Table 6.2: Statistics of SciARG-BIO

We include also the statistics for SciARG-CL in order to facilitate the comparison between both sub-corpora. Substantial differences can be observed between CL and BIO. Abstracts in BIO are, in average, longer and argumentatively more complex than those in CL, as can be seen in Table 6.2. In part this can be explained because it is frequent in BIO that abstracts describe a series of experiments, each one with their respective outcomes.

In some cases, observations/results from one experiment are used as motivation and/or justification for additional experiments. This makes the description of research outcomes and their interpretation much more complex in BIO abstracts, which leads to a significant difference in the number of units of type *observation*, *result* and *conclusion* when compared to CL abstracts, as observed in Table 6.3. Fig. 6.1 also shows how the proportion of *proposal units* (*proposal*, *proposal-implementation*) and *outcomes units* (*observation*, *result*, *conclusion*) are inverted in CL and BIO.

Consider, for instance, the abstract from (Walsh et al., 2000)[7]:

> [The progressive aggregation and deposition of amyloid beta-protein (Abeta) in brain regions subserving memory and cognition is an early and invariant feature of Alzheimer's disease, the most common cause of cognitive failure in aged humans.]$_1$ [Inhibiting Abeta aggregation is therapeutically attractive because this process is believed to be an exclusively pathological event.]$_2$ [Whereas many studies have examined the aggregation of synthetic Abeta peptides under nonphysiological conditions and concentrations, we have detected and characterized the oligomerization of naturally secreted Abeta at nanomolar levels in cultures of APP-expressing CHO cells [CIT].]$_3$ [**To determine whether similar species occur in vivo, we probed samples of human cerebrospinal fluid (CSF) and detected SDS-stable dimers of Abeta in some subjects.**]$_4$ [Incubation of CSF or of CHO conditioned medium at 37 degrees C did not lead to new oligomer formation.]$_5$ [This inability to induce oligomers extracellularly as well as the detection of oligomers in cell medium very early during the course of pulse-chase experiments suggested that natural Abeta oligomers might first form intracellularly.]$_6$ [**We therefore searched for and detected intracellular Abeta oligomers, principally dimers, in primary human neurons and in neuronal and nonneural cell lines.**]$_7$ [These dimers arose intracellularly rather than being derived from the medium by reuptake.]$_8$ [The dimers were particularly detectable in neural cells: the ratio of intracellular to extracellular oligomers was much higher in brain-derived than nonbrain cells.]$_9$ [We conclude that the pathogenically critical process of Abeta oligomerization begins intraneuronally.]$_{10}$

The text reports a *sequence* of experiments, described by the *proposal-implementation* sentences (4) and (7) (in bold). The justification for the second experiment is the result reported by sentence (6) (underlined), which, in turn, is supported by the observation included in sentence (5).

[7] pubmed.ncbi.nlm.nih.gov/10978169/

Type	CL	BIO	CL (%)	BIO (%)
proposal	290	289	24	10
proposal-implementation	260	274	22	10
observation	40	505	3	18
result	157	703	13	25
conclusion	50	301	4	11
means	27	58	2	2
result-means	70	31	6	1
motivation-problem	102	97	9	4
motivation-background	159	487	13	17
motivation-hypothesis	21	16	2	1
information-additional	23	26	2	1
Total	1199	2787	100	100

Table 6.3: Distribution of unit types in SciARG-BIO

Relation	CL	BIO	CL (%)	BIO (%)
support	420	1581	35	57
elaboration	355	535	30	19
info-required	118	303	10	11
sequence	31	2	2	0
by-means	28	57	2	2
info-optional	22	24	2	1
root	225	285	19	10
Total	1199	2787	100	100

Table 6.4: Distribution of relations in SciARG-BIO

Overall, the distinction between the plain report of observed data, the interpretation of results and the extraction of conclusions from them is more ambiguous in BIO than in CL and, therefore, differentiating these types of units is more difficult, as we observe in Section 6.1.2 when we analyze inter-annotator agreement in this domain.

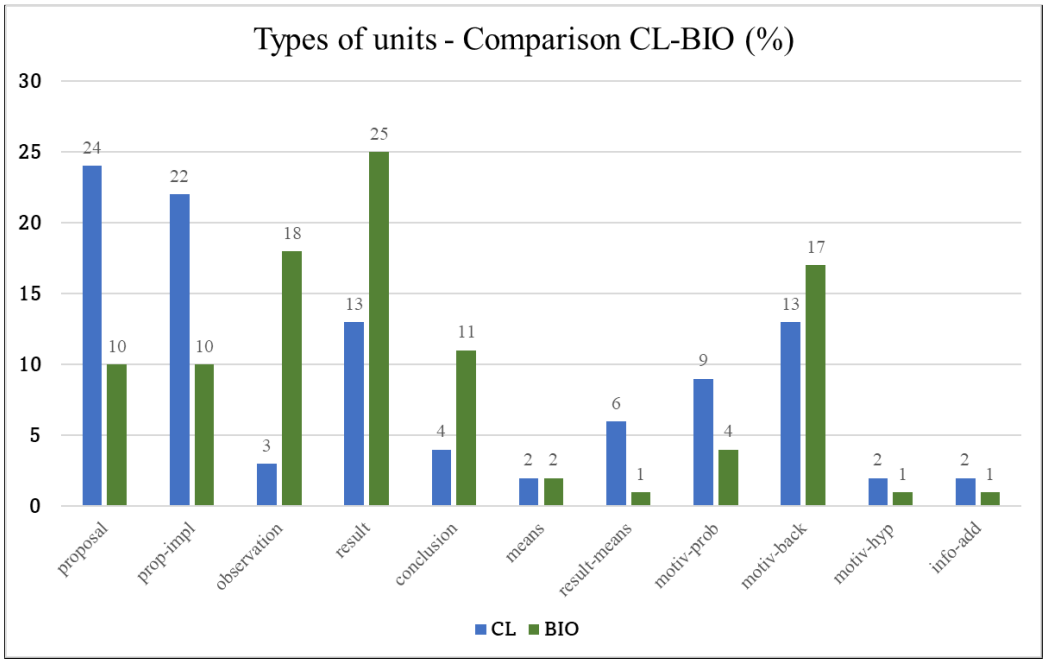


Figure 6.1: Comparison of percentages of each type of unit in CL and BIO.

The distances between units and their parents are also greater in BIO, as seen in Tables 6.5 and 6.6.

In fact, 20.8% of the times a unit is five or more units away from its parent. In CL this occurs only in 2.7% of the cases. In 85.5% of the cases CL units are only one or two units away from its parent. In BIO this occurs 64.2% of the times. The distribution between *forward* and *backward* relations is similar in both domains. In both cases *backward* relations are more frequent: the parent occurs before the child 68.1% and 66.2% of the times in CL and BIO, respectively.

Distance	Number	Percentage
1	648	25.9%
2	80	3.2%
3	46	1.8%
4	28	1.1%
5	15	0.6%
6	13	0.5%
7	6	0.2%
8	5	0.2%
9	2	0.1%
10	2	0.1%
Total	845	33.8%

Table 6.5: *Forward* relations. Distribution of the distances to parent units.

Distance	Number	Percentage
1	673	26.9%
2	203	8.1%
3	155	6.2%
4	147	5.9%
5	129	5.2%
6	105	4.2%
7	75	3.0%
8	58	2.3%
9	40	1.6%
10	25	1.0%
11	17	0.7%
12	12	0.5%
13	10	0.4%
14	4	0.2%
15	2	0.1%
Total	1655	66.2%

Table 6.6: *Backward* relations. Distribution of the distances to parent units.

6.2 Experiments with SciARG-BIO

In this section we conduct experiments with the newly annotated SciARG-BIO and assess the performance of models trained with it. As mentioned, we are particularly interested in exploring whether models trained with annotated abstracts in one scientific discipline can be easily adapted so they can be leveraged to predict the argumentative structure of texts in another discipline.

6.2.1 Experimental setups

We train and evaluate models for the four main tasks described in Section 4.1 in Chapter 4, considering units at the sentence level: i) given a unit, predict whether it is the *main unit* of the abstract, ii) given a unit predict its type, and iii) predict relations between units, which, we model by means of two two sub-tasks: given two units s_1, s_2, predict, on one hand, whether there is a link from s_1 to s_2 or from s_2 to s_1 and, on the other hand, predict the type of the relation.

We train the SciARG-BIO models both in single and multi-task settings, with the same architectures and methods described in Chapter 4 for the SciARG-CL corpus, using also SciBERT (Beltagy et al., 2019) as base model.

In the case of SciARG-BIO we do not reproduce the experiments done with CL in order to investigate the potential benefits of leveraging annotations included in other corpora. Based on the previous results we understand that improvements in performance could probably be obtained, for instance, by pre-fine-tuning models with the life-sciences and health section of the MAZEA corpus and/or with discourse-annotated corpora in the biomedical domain,[8] but this would divert us from the main objective of this chapter: to analyze the potential flow of information between SciARG tasks when trained in two different scientific disciplines.

In addition to comparing the results obtained by fine-tuning SciBERT with SciARG-BIO annotations in single and multi-task settings, we consider models obtained by

[8]Such as *BioDRB*, which includes 24 full-text biomedical articles annotated with 16 types of discourse relations adapted from the Penn Discourse Treebank (PDTB) (Prasad et al., 2011), or *BioCause*, which contains 19 full-text documents been manually annotated with biomedical entities and events for the study of causality relations. (Mihăilă et al., 2013)

fine-tuning SciBERT with the union of SciARG-CL and SciARG-BIO annotations at the same time.

In order to explore to what extent an encoder trained with SciARG annotations captures task-specific information that can be used to predict the argumentative structure of abstracts in different disciplines, we explore results obtained when using a BERT encoder fine-tuned with SciARG-CL annotations in SciARG-BIO models, without additional fine-tuning. To do this, we fine-tune SciBERT with SciARG-CL annotations in a multi-task setting and then freeze all BERT attention layers. We compare the results obtained with the SciARG-CL encoders to the results obtained when applying the same procedure to SciBERT. To do a fair comparison between the performance of both encoders, we train linear classifiers on top of them with SciARG-BIO tasks. For these experiments SciARG-BIO tasks are trained independently in single-task settings (as there are not shared parameters modified in the training process).

6.2.2 Results and analysis

In the SciARG-CL domain we use, as validation set, the consensus annotations obtained from 30 abstracts annotated in common by the three annotators. This set is used for evaluation, while the remaining 195 abstracts are used to train the models.[9] In the case of SciARG-BIO we have 50 abstracts annotated by two annotators (ann_1, ann_2). In order to build the validation set we consider, from these 50 abstracts, the subset of 35 abstracts annotated by ann_1 with the highest levels of agreement with ann_2 (when all tasks are considered), and use the other 250 abstracts annotated by ann_1 for training.[10]

In order to compare the results obtained when implementing the different training settings, we follow the same criteria applied in Chapter 4 for the evaluation of the SciARG-CL experiments: we consider the mean of the metrics obtained when taking a set of five models including the model at the training loss *elbow epoch*,[11] as well as the models obtained two epochs before and two epochs after it. We include in the tables the confidence intervals for the mean F_1 scores.

[9] With a proportion of 87% of documents used for training and 13% for validation.
[10] With a proportion of 88% of documents used for training and 12% for validation.
[11] Calculated automatically by applying the method described in 4.4 in Chapter 4.

SciBERT fine-tuned and evaluated with SciARG-BIO							
Single-task							
Task	**Avg.**	**Elbow**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weight.	6	4-8	0.6993	0.6568	0.6560	±0.0092
Relation type	Weight.	6	4-8	0.7448	0.7481	0.7334	±0.0132
Main unit	Macro	6	4-8	0.8302	0.8749	0.8498	±0.0158
Parent attachment	Macro	9	7-11	0.7206	0.6747	0.6908	±0.0058
Multi-task							
Task	**Avg.**	**Elbow**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weight.	6	4-8	0.7034	0.6865	**0.6791**	±0.0092
Relation type	Weight.	6	4-8	0.7568	0.7703	**0.7515**	±0.0113
Main unit	Macro	6	4-8	0.8954	0.9286	**0.9108**	±0.0140
Parent attachment	Macro	8	6-10	0.7169	0.6928	**0.6992**	±0.0086

Table 6.7: Results of fine-tuning SciARG-BIO in single and multi-task setting. Average of models in epochs $[elbow - 2, elbow + 2]$ with confidence intervals for F_1.

In Table 6.7 we observe that the SciARG-BIO models yield significantly poorer performances when compared to the results obtained in SciARG-CL (Chapter 4). In particular, for the *unit type* and *parent attachment* tasks, the results in BIO are between 0.12 and 0.13 F_1-points below those obtained in CL in single-task settings without pre-fine-tuning.

Differences in performance in both disciplines are, to some extent expected, considering the greater argumentative complexity of biomedical abstracts, which, as seen in Section 6.1.2, is also reflected in the lower levels of inter-annotator agreement.

As it is also the case in SciARG-CL, the models in which the tasks are trained jointly in a multi-task settings perform better than those in which the tasks are trained independently, in line with our observations with respect to the level of shared information between the different SciARG tasks.

Training with the full SciARG corpus (SciARG-CL+SciARG-BIO)

Tables 6.8, 6.9, 6.10 show the performance of a single-task model trained with the full SciARG training set (the union of SciARG-CL and SciARG-BIO) and evaluated both with discipline-specific validation sets as well as with their union.

We compare the results with the ones obtained by the models trained and evaluated within a single discipline. We consider only models trained in single-task settings so we can clearly assess the differences obtained when we use instances in both disciplines for training. In the case of multi-task models it is difficult to determine to what extent an observed variation responds of the interaction of the training signals or to the characteristics of the training data.

Evaluation in SciARG-BIO							
Train w/SciARG-BIO (single-task)							
Task	**Avg.**	**Elbow**	**Epochs**	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{CI_{F_1}}$
Unit type	Weight.	6	4-8	0.6993	0.6568	0.6560	±0.0092
Relation type	Weight.	6	4-8	0.7448	0.7481	**0.7334**	±0.0132
Main unit	Macro	6	4-8	0.8302	0.8749	0.8498	±0.0158
Parent attachment	Macro	9	7-11	0.7206	0.6747	0.6908	±0.0058
Train w/SciARG-CL+SciARG-BIO (single-task)							
Task	**Avg.**	**Elbow**	**Epochs**	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{CI_{F_1}}$
Unit type	Weight.	6	4-8	0.6971	0.6833	**0.6770**	±0.0114
Relation type	Weight.	6	4-8	0.7245	0.7405	0.7238	±0.0046
Main unit	Macro	7	5-9	0.8729	0.9117	**0.8900**	±0.0181
Parent attachment	Macro	8	6-10	0.7424	0.6881	**0.7085**	±0.0099

Table 6.8: Single-task fine-tuning SciARG-BIO and SciARG-CL+BIO, evaluated in SciARG-BIO. Avg. of models in epochs $[elbow - 2, elbow + 2]$ with conf. int. for F_1.

We observe that–with the exception of the main unit task for SciARG-BIO–the results obtained with the models trained with the whole SciARG corpus (SciARG-CL+SciARG-BIO) are not significantly different to those obtained when training only with the domain-specific training set. Only minor to moderate gains in the averaged scores are obtained, in general. As differences fall within the confidence intervals, no definite conclusions can be drawn, but a tendency for improved performances is observed in the prediction of *unit type* and *parent attachment* tasks in the case of SciARG-BIO, and for the *relation type* task in the case of SciARG-CL.

In the case of the *main unit* task a more significant gain is observed in SciARG-BIO, in line with the lesser ambiguity of this task in both domains (in the case of SciARG-CL there is less margin for improvement as the original score is already high).

Evaluation in SciARG-CL							
Train w/SciARG-CL (single-task)							
Unit type	Weight.	7	5-9	0.8060	0.7748	0.7821	±0.0167
Relation type	Weight.	6	4-8	0.7991	0.7854	0.7832	±0.0204
Main unit	Macro	7	5-9	0.9023	0.9035	0.9026	±0.0066
Parent attachment	Macro	9	7-11	0.8042	0.8293	0.8150	±0.0207
Train w/SciARG-CL+SciARG-BIO (single-task)							
Task	**Avg.**	**Elbow**	**Epochs**	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{CI_{F_1}}$
Unit type	Weight.	6	4-8	0.8189	0.7947	**0.7985**	±0.0100
Relation type	Weight.	6	4-8	0.8069	0.8013	**0.8017**	±0.0158
Main unit	Macro	7	5-9	0.9139	0.9127	**0.9124**	±0.0101
Parent attachment	Macro	8	6-10	0.8063	0.8260	**0.8154**	±0.0101

Table 6.9: Results of fine-tuning SciARG-CL alone and SciARG-CL+BIO, evaluated in SciARG-CL. Avg. of models in epochs $[elbow - 2, elbow + 2]$ with conf. int. for F_1.

Train and evaluation in full SciARG corpus (SciARG-CL+SciARG-BIO) (single-task)							
Task	**Avg.**	**Elbow**	**Epochs**	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$	$\boldsymbol{CI_{F_1}}$
Unit type	Weight.	6	4-8	0.7339	0.7156	0.7146	±0.0086
Relation type	Weight.	6	4-8	0.7525	0.7581	0.7500	±0.0059
Main unit	Macro	7	5-9	0.8920	0.9128	0.9014	±0.0100
Parent attachment	Macro	8	6-10	0.7618	0.7239	0.7391	±0.0065

Table 6.10: Results of fine-tuning and evaluating models trained with the full SciARG corpus. Avg. of models in epochs $[elbow - 2, elbow + 2]$ with conf. intervals for F_1.

For ease of comparison, we also include a summary of the results obtained for the considered tasks within each discipline and with the combined corpora in Table 6.11, where the differences in performance of the various models in their respective validation sets can be seen. The results obtained with the full SciARG corpus are, for all the tasks and settings, in between of the results obtained within the specific subsets, being the identification of the unit types and their attachment to the parent unit the tasks for which the greatest differences are observed.

Summary of F_1 scores in SciARG-CL, SciARG-BIO and SciARG-CL+BIO			
Single-task			
Task	**SciARG-CL+BIO**	**SciARG-CL**	**SciARG-BIO**
Unit type	0.7146	0.7821	0.6560
Relation type	0.7500	0.7832	0.7334
Main unit	0.9014	0.9026	0.8498
Parent attachment	0.7391	0.8150	0.6908
Multi-task			
Task	**SciARG-CL+BIO**	**SciARG-CL**	**SciARG-BIO**
Unit type	0.7130	0.8005	0.6791
Relation type	0.7873	0.7910	0.7515
Main unit	0.9138	0.9323	0.9108
Parent attachment	0.7343	0.8316	0.6992

Table 6.11: Summary of average F_1 scores in SciARG-CL, SciARG-BIO and SciARG-CL+BIO.

Performance of SciARG-CL encoder in SciARG-BIO models

We now evaluate the results obtained when we keep the parameters of the BERT encoder trained with SciARG-CL (in a multi-task setting) fixed, and contrast them to the results obtained when freezing SciBERT's encoder. We compare the performance of linear classifiers for SciARG-BIO tasks trained on top of both encoders.

We observe in Table 6.12, that significant gains are obtained with the frozen BERT encoder pre-trained with SciARG-CL annotations when compared to the results obtained with the original SciBERT encoder. Considering the notable differences in computational linguistics and biomedicine papers, both in terms of the types of problems addressed and the implemented methodologies, these results encourage to consider the possibility that the information encoded by means of fine-tuning SciBERT with SciARG annotations conveys knowledge about the argumentative structure of the abstracts that does not depend on their specific discipline. In addition, we observe that the results obtained with the frozen SciARG-CL encoder are very similar–and better in the case of *relation type* and *main unit* tasks–to the results obtained when fine-tuning SciBERT with SciARG-BIO's annotations in a single-task setting.

SciARG-BIO classifiers trained on top of frozen BERT encoders							
SciBERT encoder without further fine-tuning							
Task	**Avg.**	**Elbow**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weight.	6	4-8	0.5159	0.5535	0.5144	±0.0224
Relation type	Weight.	6	4-8	0.6229	0.6519	0.6026	±0.0414
Main unit	Macro	6	4-8	0.8390	0.7852	0.8084	±0.0303
Parent attachment	Macro	5	3-7	0.6239	0.4740	0.5090	±0.0195
SciARG-CL encoder without further fine-tuning							
Task	**Avg.**	**Elbow**	**Epochs**	P	R	F_1	CI_{F_1}
Unit type	Weight.	7	5-9	0.6711	0.6514	**0.6429**	±0.0125
Relation type	Weight.	8	6-10	0.7321	0.7437	**0.7347**	±0.0055
Main unit	Macro	6	4-8	0.8650	0.8780	**0.8713**	±0.0045
Parent attachment	Macro	6	4-8	0.7962	0.6414	**0.6986**	±0.0060

Table 6.12: Results of training SciARG-BIO classifiers on top of frozen SciBERT and pre-fine-tuned SciARG-CL BERT encoders. Average of models in epochs $[elbow - 2, elbow + 2]$ with confidence intervals for F_1.

The fact that the BERT encoders fine-tuned with SciARG-BIO do not perform substantially better in its own validation set can be a result of the higher complexity in SciARG-BIO's training and validation annotations. We could hypothesize that the training signal that the models obtain from SciARG-BIO annotations does not add significantly to the information that can be obtained from the less complex SciARG-CL annotations. In other words, with the current amount of data in SciARG-BIO, the benefit that the models could obtain from being trained with texts similar to the ones used for evaluation seems to be counter-balanced by their greater complexity.

Note on intra-sentence unis in SciARG-BIO

Only 99 sentences (3.5%) are annotated with a second type in SciARG-BIO, and only 31 sentences (1.1%) are assigned the type *result-means*, in contrast with the 5.8% observed in the case of SciARG-CL (Table 6.3).

It is likely that the low number of sentences annotated with more than one type is a consequence of the annotators not having domain-specific knowledge and, therefore, not being able to identify fine-grained information within them.

The high level of complexity of the biomedical discourse and the minimal number of examples in which more than one argumentative type is identified makes it unfeasible to implement the tasks described in Chapter 5 for the identification of intra-sentence units and relations in SciARG-BIO.

6.3 Conclusions

In this chapter we explored the application of the SciARG scheme to the annotation of a corpus of abstracts in biomedicine, a scientific discipline different to the one for which it was originally developed. The analysis of the annotations and the results of the experiments conducted with them encourage us to think that the SciARG annotation scheme can be successfully applied to scientific disciplines other than computational linguistics, even when we identified some significant differences in the argumentative structure of the biomedical abstracts with respect to the computational linguistic ones–in particular, in terms of their higher complexity and ambiguity. Some of these aspects had not been originally contemplated in the original annotation guidelines (for instance, in the way in which results of multiple related experiments are reported in biomedicine). We believe that, without changing the annotation scheme, minor adaptations to the guidelines could contribute to avoid some level of ambiguity in the annotations and to improve the inter-annotator agreement for the most difficult tasks–in particular, the identification of relations and their types. Nevertheless, some characteristics of the discourse structure and the language used in biomedical abstracts seem to require specific domain knowledge for the proposed annotation tasks. This could explain in part the low percentage of sentences annotated with more than one type of unit.

We conducted experiments in single and multi-task settings with the new annotated corpus, SciARG-BIO, and analyzed their results, which confirm our previous observations regarding the potential benefits of training argument mining tasks jointly in a multi-task setting. We also explored results obtained by training a model with the combined annotations in SciARG-CL and SciARG-BIO, observing only minor improvements with respect to the models trained with the smaller, discipline-specific, training sets.

More relevantly, we conducted experiments that showed that BERT encoders fine-tuned with computational linguistics annotations capture knowledge about the argumentative structure of the abstracts–as modeled by the considered tasks–that makes them perform significantly better than the base BERT model without fine-tuning for the prediction of biomedical argumentative units and relations. Even more, the SciARG-CL-trained encoder performs competitively–and for some tasks better–when compared to an encoder fine-tuned with annotations in SciARG-BIO, the same discipline as the texts included in the validation set.

Part II

Prediction of argumentative quality dimensions

Chapter 7

ARGUMENTATIVE QUALITY ASSESSMENT: BACKGROUND AND RELATED WORK

In this chapter we provide context for the topics addressed in the second part of the thesis and review preceding works closely related to them.

- In Section 7.1 we first consider works aimed at assessing argumentative quality in diverse types of texts, and then examine few initiatives in which scientific text mining and argumentative quality analysis intersect, including the assessment of scientific claims. Finally, we briefly describe a theory-grounded systematization of the multiple dimensions involved in the assessment of arguments, which we adopt as framework for our analyses in the two following chapters.
- In Section 7.2 we consider antecedents in the prediction of scores assigned by referees in peer-review processes, which are designed to capture quality aspects the papers–including some argumentative dimensions–in order to decide their acceptance or rejection in a particular venue.

7.1 Argumentative quality assessment

The quality of arguments can be analyzed from many different perspectives (Tindale, 2007; Johnson and Blair, 2006), and work aimed at clarifying what a *good* argument is can be traced back to Aristotle (Rapp, 2010). Depending on the adopted viewpoint, assessing argumentation can be so difficult as to consider somewhat intangible aspects–such as the intentions and ethical frameworks adopted by the participants in a debate.

Research aimed at assessing the quality of natural language arguments from a computational perspective has, expectably, focused on argumentative dimensions that can be ranked in terms of their *persuasive effectiveness* (e.g., the *rhetorical organization* of arguments, the *type and amount of evidence* provided to support a claim). This is the case, for instance, of scores assigned to student essays.

7.1.1 Textual genres and argumentative dimensions

Persing and Ng (2015) is one of the first works to propose a feature-rich model to predict an *argument strength score* of student essays.[1] In addition to features that are common to several NLP tasks (such as POS n-grams, cue words, etc.) they introduce information about the *argumentative structure* of the essays by considering *major claims*, *claims* and *premises*. They find that the feature-rich proposal outperforms a simpler rule-based approach that had been previously considered in a pilot experiment by Ong et al. (2014). Similarly, Ghosh et al. (2016) propose a rich set of argumentative features that includes the number of *claims*, *premises*, *relations* (number of supported claims) and typology of argumentative structure (chains or trees) to train a machine learning model that predicts human scores in a dataset of 107 TOEFL[2] essays. Both Persing and Ng (2015) and Ghosh et al. (2016) adopt the approach by Stab and Gurevych (2014a) to identify argumentative units and relations in texts.

Stab and Gurevych (2017b) propose as a task the automatic assessment of whether arguments in student essays are *sufficiently* supported (i.e., whether its premises

[1] From the International Corpus of Learner English (ICLE) (Granger et al., 2009).

[2] Test of English as a Foreign Language (TOEFL): `ets.org/toefl`.

provide enough evidence for accepting its claim). They find that *insufficiently* supported arguments exhibit specific lexical indicators and can be identified with high accuracy using convolutional neural networks (CNN). In turn, in (Stab and Gurevych, 2016), they address a related task:[3] the recognition of *myside biases*– the tendency in a document to ignore opposing viewpoints. They compare the performance obtained with different sets of features (including syntactic, semantic, discourse and sentiment-based features), finding that lexical indicators–in particular, unigrams and adversative transitional phrases–are the most predictive features for this task.

Wachsmuth et al. (2016) explore the possibility of leveraging the argumentative structure automatically extracted from persuasive essays to predict four argumentative quality dimensions: *organization*, *thesis clarity*, *prompt adherence*, and *argument strength*. They consider the argumentative components proposed by Stab and Gurevych (2014a) and a simplified version of the model by (Stab and Gurevych, 2014b) in which argumentative units are considered at the sentence level. For the argument mining experiments, they investigate six types of features that provide information about *content*, *style*, and *position* of sentences,[4] finding the latter to be the most predictive for the classification of the type of argumentative component. For the prediction of essay scores they consider the argumentative structure of the essays in terms of the *sequences of argumentative units* that they contain, which they compare to standard content-based features (tokens, POS n-grams), and to features derived from considering sequences of paragraph-level discourse functions, sentence-level discourse relations, and paragraph-level sentiments.[5] Their results show that automatically-extracted argumentative information does contribute to predict argumentative quality aspects of the essays. In particular, those related to the textual *organization*.

The automatic assessment of argumentative quality dimensions has been explored in textual genres other than persuasive essays, including online debates. The initiative by Cabrio and Villata (2012a) described in Chapter 2, leverages textual entailment relations in order to construct a graph of arguments which is used to predict the *acceptability* of arguments on Debatepedia.org.[6]

[3] As the inclusion of opposing arguments in essays is correlated with their argumentation quality (Wolfe et al., 2009).

[4] Within paragraphs and the full text of the essay.

[5] These features are described in (Wachsmuth et al., 2014, 2015).

[6] Now iDebate: `idebate.org/debatabase`

Other works have focused, for instance, on the assessment of the *persuasiveness* of user posts on web fora and debate portals. Wei et al. (2016) compare the performance of argumentative and non-argumentative features to predict a *persuasiveness* score–as voted by the community of users–of posts on the *ChangeMyView* Reddit community.[7] Among the non-argumentative features, they consider standard *surface* features (e.g., length of the post), as well as *interaction* features in which the post is taken in the context of the debate (e.g., position of the post in a discussion thread). Within the argumentative features they consider, for instance, the comment's *relevance* (by computing the similarity with the original post), as well as the number and percentage of sentences classified as *argumentative* by a binary classifier trained with features from (Stab and Gurevych, 2014b). Their results show that argumentation-based features work well for short threads, while *interaction* features perform better as the number of comments in the thread grows.

Habernal and Gurevych (2017) approach the prediction of arguments' *convincingness*, which they assess in a collection of posts from debate portals. They rely on crowd-sourced annotations and model the task as the classification of pairs of arguments, based on the idea that the relative assessment of arguments can be easier and/or more reliable than scoring arguments individually.[8] As a result of this work, the *UKPConvArg1* corpus was made available, which cover 16,000 pairs of arguments over 32 topics. In addition to the annotation work, the authors investigate ordering properties of the "*more convincing*" relation, which can contribute to characterize graphs to represent sets of related arguments. They empirically confirm that the relation can be considered to define a total strict order, a fact that they use to generate additional datasets, including *UKPConvArgRank*, where individual arguments are ranked based on their *convincingness*. As a follow-up to this work, Habernal and Gurevych (2016a) are interested in assessing qualitative properties of *convincingness*, for which they gather–by means of a crowd-source platform–26,000 explanations written in natural language describing *why* an argument is *more* or *less* convincing. They apply automatic discourse-parsing and pattern-matching strategies to extract, from the users' texts, a set of *reason units*. After additional crowd-sourced validation processes, they end up with a set of 19

[7] `reddit.com/r/changemyview`

[8] Additional research, including (Wachsmuth and Werner, 2020; Toledo et al., 2019), has shown that this is not necessarily the case.

classes[9] representing *reasons* for argumentative *convincingness*, which they use in a final question-guided annotation through which they obtain argument pairs annotated with multiple labels that indicate the reasons for considering one argument to be *more convincing* than the other one. The reason-annotated corpus, *UKPConvArg2* was also made available to the research community. Based on these initial investigations in the assessment of arguments' *convincingness*, additional works were developed, including (Potash et al., 2017; Simpson and Gurevych, 2018; Gleize et al., 2019; Toledo et al., 2019).

7.1.2 Assessing scientific claims

The automatic assessment of argumentative quality of scientific texts from an argument mining perspective is a largely unexplored area. A somewhat related task–even if more narrowly focused–is the identification of potential *contradictions* in research literature, which requires the identification of some types of argumentative units, such as *claims*. This task is addressed by Blake (2010); Park and Blake (2012), as mentioned in Chapter 2. To the best of our knowledge, most of the methods that have been proposed for the identification of *contradictory claims* are tied to specific domains and therefore not easily extended. Sarafraz (2011), for instance, identify conflicting claims in the report of chemical interactions in the BioNLP09 corpus by means of exploiting *domain-specific* features (type of event, participants, anatomical location), the degree of *assertiveness* of the statements, and their *polarity*.

The detection of contradictory claims in biomedical abstracts is the subject of Abdulaziz Alamri's PhD thesis (Alamri, 2016), where one of the challenges that he addresses is the lack of annotated corpora for the identification of conflicting claims. He therefore proposes both a manual and an automated method to facilitate these annotations. For the automated process, claims are identified by exploiting *subject-predicate-object* triples extracted from PubMed abstracts and contained in the *SemMedDB* repository (Kilicoglu et al., 2012). Both the manual and the automatically generated corpora are used to train and evaluate a classification pipeline for the identification of sentences in documents that can be considered as potentially *affirmative* or *negative* answers with respect to a given research question. The system achieves an F_1 score of 0.83 using the manually-annotated

[9]The number of classes used in other experiments has then changed.

corpus, and an F_1 score of 0.78 with the automatically generated one. Pinto and Balke (2020) also explore the possibility of identifying potential contradictions between claims[10] included in a document with those obtained from querying a repository. They also work with biomedical documents and, in the same line as Alamri (2016), use semantic knowledge contained in the SemMedDB repository for claim identification.

While in the past years there has been a growing body of research in the area of automated fact checking (Vlachos and Riedel, 2015; Thorne and Vlachos, 2017; Thorne et al., 2018; Hanselowski et al., 2019), it is only recently that resources and methods for the automatic verification of scientific claims have been proposed. Wadden et al. (2020) take steps to start filling this void by developing SciFACT,[11] a dataset of scientific claims paired with evidence-annotated abstracts that *support* or *refute* them. The authors use the annotated dataset to train a model that they then evaluate by assessing the verifiability of claims concerning COVID-19, where the retrieved *pro* and *against* evidence for the considered claims is evaluated by a domain expert. The authors affirm that the results show the practical value of the corpus and encourage further research with it. To this end, a scientific-claim verification shared-task (SCIVER) was proposed, the first edition of which took place within the context of the Second Workshop on Scholarly Document Processing.[12]

7.1.3 Theory-grounded assessment of arguments

As seen in Section 7.1.1, with increased interest in assessing different facets of arguments, initiatives targeting multiple tasks began to proliferate in various domains and textual genres. In some cases, works embraced argumentative quality criteria based on theories of argumentation and, in others, practical approaches were adopted, with *ad hoc* definitions of argumentative quality. It became necessary, therefore, to establish a common framework through which links could be established between proposals for the computational assessment of arguments with each other and with existing theories of argumentation quality.

[10] They define this task as assessing the *plausability* of a document within the current state-of-the-art.

[11] `github.com/allenai/scifact`

[12] `sdproc.org/2021/sharedtasks.html#sciver`

This emerging need was identified and addressed by Wachsmuth et al. (2017a). In this work the authors conduct a thorough review of theories for the assessment of argumentation as well as NLP initiatives for the automatic assessment of argumentative quality. Based on this analysis, they distill a taxonomy that provides a common ground for the quality assessment of natural language arguments. In Chapters 8 and 9 we adopt this taxonomy. The taxonomy includes three top-level argumentative quality dimensions: *cogency*, *effectiveness*, and *reasonableness*, which represent, respectively, the *logical*, *rhetorical*, and *dialectical* aspects of argumentative quality–even if the authors make it clear that there are no clear-cuts between the different aspects of argumentation. The three top-level dimensions are, in turn, divided into 15 fine-grained dimensions, as shown in Table 7.1.

Wachsmuth et al. (2017a) evaluate the applicability of the theory-motivated taxonomy by using the quality dimensions to annotate 320 arguments from *UKP-ConvArgRank* (Habernal and Gurevych, 2016b) with a 3-point scale, producing the *Dagstuhl-15512 ArgQuality Corpus*, which is made available for further research. The annotators assess the 15 quality dimensions in the taxonomy, and also score the texts' *overall quality*. Wachsmuth et al. (2017a) analyze the corpus statistics and inter-annotator agreement in terms of the Krippendorff's α (Krippendorff, 2007) for the most agreeing pair of annotators. In addition, they study the level of correlation between quality dimensions, as well as with the *overall* score.[13] The agreement analysis shows that the largest α values are obtained for the assessment of the *overall* quality, which is a positive indicator of the usability of the taxonomy to guide the assessment of arguments. The variation in agreement for some theory-motivated dimensions, in turn, yields some light on their level of subjectivity and, therefore, difficulty for being evaluated–even by human experts. Finally, the analysis of correlations between coarse and fine-grained dimensions, as well with the *overall* score, shows expected results, which suggests the adequacy of the proposed taxonomy. In (Wachsmuth and Werner, 2020), a follow-up work to (Habernal and Gurevych, 2016a) and (Wachsmuth et al., 2017a), the authors compare the theory-based absolute quality ratings assigned by experts in the *Dagstuhl-15512 ArgQuality Corpus* with the relative quality scores assigned by crowd annotators in *UKPConvArg2*, finding that there is a clear correlation between the two annotations. They also observe that the explanations offered by lay annotators–in relation to why one argument is more convincing than another one–are well-captured by the theory-based quality dimensions.

[13] Majority scores are considered for the corpus statistics and correlation analyses.

At the same time, considering the dimensions that prove to be more feasible to assess in practice can contribute to simplify the theory-based taxonomy, thus improving its applicability.

Dimension	Definition
Cogency	An argument is cogent if it has acceptable premises that are relevant to its conclusion and that are sufficient to draw the conclusion.
Local acceptability	A premise of an argument is acceptable if it is rationally worthy of being believed to be true.
Local relevance	A premise of an argument is relevant if it contributes to the acceptance or rejection of the argument's conclusion.
Local sufficiency	An argument's premises are sufficient if, together, they give enough support to make it rational to draw its conclusion.
Effectiveness	Argumentation is effective if it persuades the target audience of (or corroborates agreement with) the author's stance on the issue.
Credibility	Argumentation creates credibility if it conveys arguments and similar in a way that makes the author worthy of credence.
Emotional Appeal	Argumentation makes a successful emotional appeal if it creates emotions in a way that makes the target audience more open to the author's arguments.
Clarity	Argumentation has a clear style if it uses correct and widely unambiguous language as well as if it avoids unnecessary complexity and deviation from the issue.
Appropriateness	Argumentation has an appropriate style if the used language supports the creation of credibility and emotions as well as if it is proportional to the issue.
Arrangement	Argumentation is arranged properly if it presents the issue, the arguments, and its conclusion in the right order.
Reasonableness	Argumentation is reasonable if it contributes to the issue's resolution in a sufficient way that is acceptable to the target audience.
Global acceptability	Argumentation is acceptable if the target audience accepts both the consideration of the stated arguments for the issue and the way they are stated.
Global relevance	Argumentation is relevant if it contributes to the issue's resolution, i.e., if it states arguments or other information that help to arrive at an ultimate conclusion.
Global sufficiency	Argumentation is sufficient if it adequately rebuts those counterarguments to it that can be anticipated.

Table 7.1: Taxonomy of argumentation quality. Source: (Wachsmuth et al., 2017a).

7.2 Prediction of scores for peer-reviewed manuscripts

In Chapter 9 we explore whether features conveying information about the argumentative structure of abstracts can contribute to predict scores assigned by referees in a peer-review process. Closely related to this task is the prediction of the acceptance/rejection of research manuscripts in scientific venues, which has been addressed in several works, either by exploiting content-features extracted from the texts, meta-data of the manuscripts (e.g., keywords, authors and their affiliations), information about the venues to which the manuscripts are submitted, or a combination of them all.

Some of the recent research around the prediction of reviewer's decision was motivated by the availability of the *PeerRead* dataset[14] (Kang et al., 2018), which contains a collection of submitted manuscripts to computational linguistics and machine learning conferences–Neural Information Processing Systems (NIPS) 2013-2017, Annual Meeting of the Association for Computational Linguistics (ACL) 2017, Conference on Computational Natural Language Learning (CoNLL) 2017 and International Conference on Learning Representations (ICLR) 2017–as well as papers published on the arXiv.org[15] platform between 2007 and 2017. Kang et al. (2018) first describe the dataset and provide statistical information about how reviewers' scores–both *overall* and *aspect-specific* scores, such as *novely*, *substance*, etc.–are distributed in the different venues, and then propose two tasks based: i) given a manuscript, predict whether it is *accepted* in a target conference, and ii) given the truncated text of a review and/or a manuscript,[16] predict the *aspect scores* assigned by the reviewers. For the first task they use a set of features obtained from the paper's abstract (e.g., occurrence of certain words, standard information-retrieval features), the structure of the paper (e.g., sections, number of equations, figures, tables), its metadata (e.g., number of authors), and references (e.g., number and years of references). They test multiple standard classifiers, obtaining significant gains with respect to a majority-class classifier that they consider as baseline.

[14] `github.com/allenai/PeerRead`

[15] `arxiv.org`

[16] They consider three scenarios: only the text of the review, only the text of the paper, and both texts together, truncating the texts of the papers to the first 1,000 tokens and the texts of the reviews to the first 200 tokens. It is not said explicitly, but we assume that when only using the text of the manuscript the predicted values are the average scores.

The second task is modeled as a regression problem, where the mean of the scores assigned by the reviewers as is taken as baseline. They use GloVe embeddings (Pennington et al., 2014) and different neural-network architectures, including a convolutional neural network (CNN) (Zhang et al., 2015), a long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997), and a deep averaging network (DAN) (Iyyer et al., 2015). They report the results in the ACL and ICLR subsets, observing that for some *aspects* the mean score is difficult to beat. Overall, the best performing models are those trained with the combined texts of the manuscript of the review, while–somewhat surprisingly–considering only the texts of the reviews yields, in general, poor performances.[17]

Since the presentation of PeerRead and the publication of the results obtained with the preliminary experiments conducted with it, the tasks and dataset have been adopted as benchmarks by several subsequent works. Qiao et al. (2018), for instance, use the PeerRead data to train and evaluate a hierarchical model that combines CNN and LSTM modules, with the goal of obtaining better representations of the manuscripts' structure and content, obtaining moderate gains for several (not all) of the predicted *aspect* scores over simpler CNN or LSTM networks. Skorikov and Momen (2020) address the classification of the acceptance/rejection of PeerRead manuscripts, with a similar approach to that implemented in the original paper. The most significant difference is that they use a smaller set of features, which prove to perform comparatively better with a Random Forest algorithm. In a more innovative proposal, Ghosal et al. (2019) consider the predicted sentiment of the reviews to predict manuscripts' *acceptance* and *recommendation scores* in PeerRead, presenting their work as a contribution to the prediction of the *reliability* of the reviews.

[17] A potential explanation to this could lie in the fact of the reviews being truncated.

Chapter 8

PREDICTING CLARITY AND SUFFICIENCY IN SCIARG-CL

From the three top-level theory-based argumentative quality dimensions included in the taxonomy proposed by Wachsmuth et al. (2017a)–*cogency*, *effectiveness* and *reasonableness*–we are interested is predicting the logical quality of abstracts included in the SciARG-CL corpus in terms of their *cogency* or argumentative strength, and their rhetorical quality in terms of their persuasive *effectiveness*. In this chapter we explore whether features obtained from the argumentative structure of abstracts can contribute to predict these specific argumentative quality dimensions.

When we analyze the quality scores assigned by different annotators we observe that there is a significant variation in terms of the consistency of the annotations, in line with the high level of subjectivity involved in the task. We propose a simple method to incorporate information about the annotators–and, therefore, their reliability–into the models, either explicitly, or by weighing training instances according to who annotated them.

This chapter is organized as follows:

- In Section 8.1 we describe the specific argumentative quality dimensions that we deem feasible to assess in the SciARG-CL corpus.

- In Section 8.2 we describe the annotation process, analyze annotation statistics and agreement, and finally propose a method to incorporate information about the annotations' reliability when training models.
- In Section 8.3 we describe the experimental setup, including a description of the algorithms and features considered, and the weighting strategy proposed.
- In Section 8.4 we report the results obtained by means of the proposed machine learning algorithms and sets of features, and analyze to what extent correspondences can be established between the features automatically extracted from the argumentative structures of the abstracts and the quality dimensions being assessed.
- In Section 8.5 we summarize the main findings of the chapter and provide some concluding remarks.

8.1 Argumentative quality dimensions

As mentioned, in this work we focus on two broad argumentative quality dimensions of the texts: their *cogency* and their *effectiveness*. In turn, for each of them we consider which specific dimensions are the most applicable to the assessment of the argumentative structure of scientific abstracts. For instance, the analysis of the *emotional appeal* of the texts was considered not to be directly applicable in our case, while other dimensions such as *local acceptability*, *credibility*, or *appropriateness* were expected to have little variation within a set of published abstracts.

While we initially aimed at evaluating also the *arrangement* and *relevance* of the abstracts, after a preliminary round of annotations and subsequent discussions with the annotators, we observed that these dimensions were either too difficult to assess within a manageable level of subjectivity and/or could not be clearly distinguished from others. For instance, the perspectives on what was perceived as the preferred *arrangement* varied greatly from one annotator to the other–depending, also, with their familiarity with the discipline, while it was difficult for annotators to identify information that could be deemed as not *relevant* in the abstracts. This is expected as we are working with abstracts accepted in top-level conferences in the area.

These considerations led us to score only two specific dimensions: *clarity* and *(local) sufficiency*. We hypothesize that, in the case of scientific abstracts, these specific dimensions can be considered as proxies for the evaluation of the corresponding top-level ones–*effectiveness* and *cogency*, respectively. This idea is supported by the analysis of correlations between general and specific dimensions in other textual genres–in particular, in arguments from online debate portals–as described in (Wachsmuth et al., 2017a).

8.2 Annotation of quality dimensions

The CL section of the SciARG corpus, SciARG-CL, contains 225 computational linguistics abstracts from the ACL Anthology annotated with discourse and argumentative annotation layers. The three annotators involved in the annotation of argumentative units and relations in the CL section of the SciARG corpus were asked to rank each abstract with a score in a three-point scale for the considered argumentative quality dimensions. We asked the annotators to first read the abstracts and assign the scores before proceeding to annotate the argumentative units and relations.

In order to assign the scores, the annotators were asked to respond to the following questions:

> ***Clarity**: For the reasonably well-prepared reader, is it clear what was done and why?*
>
> ***Sufficiency**: Are the premises (motivation, evidence) provided enough to justify the proposed solution / approach?*

Table 8.1 shows how the correlation between the scores assigned to *clarity* and *sufficiency* vary depending on the annotator. We analyze both overall and individual Pearson's correlation coefficients r.[1] While there is a clear correlation between both dimensions for annotator ann_3, this is not the case for annotators ann_1 and ann_2. Taking into consideration these results, we assume that, at least for annotators ann_1 and ann_2 the two tasks can be clearly distinguished from each other.

[1] We also analyzed mutual information coefficients obtaining the same results, so we omit them here.

Annotator	Correlation
ann_1	0.17
ann_2	0.10
ann_3	0.44
All	*0.29*

Table 8.1: Pearson's correlation coefficients for *clarity* and *sufficiency* scores, by annotator and overall. Statistically significant values in italics.

It is also relevant to analyze the trends followed by the different annotators in the use of the different scores. Figs. 8.1 and 8.2 show the percentage of use of each score by each annotator for the *clarity* and *sufficiency* dimensions, respectively.

We can observe that annotator ann_1 tends to annotate abstracts with the lowest score (*1*) much more frequently than annotators ann_2 and $ann3$, while the opposite occurs in the annotation of the *sufficiency* dimension. In turn, the contrary situation is observed with respect to the annotation of the highest score (*3*). These seemingly different criteria confirms the high level of subjectivity involved in this task.

It is relevant to note that these statistics are calculated over the full set of annotations. In the next section we analyze how score assignments by each annotator translates into inter-annotator agreements when computed in the subset of overlapping annotations.

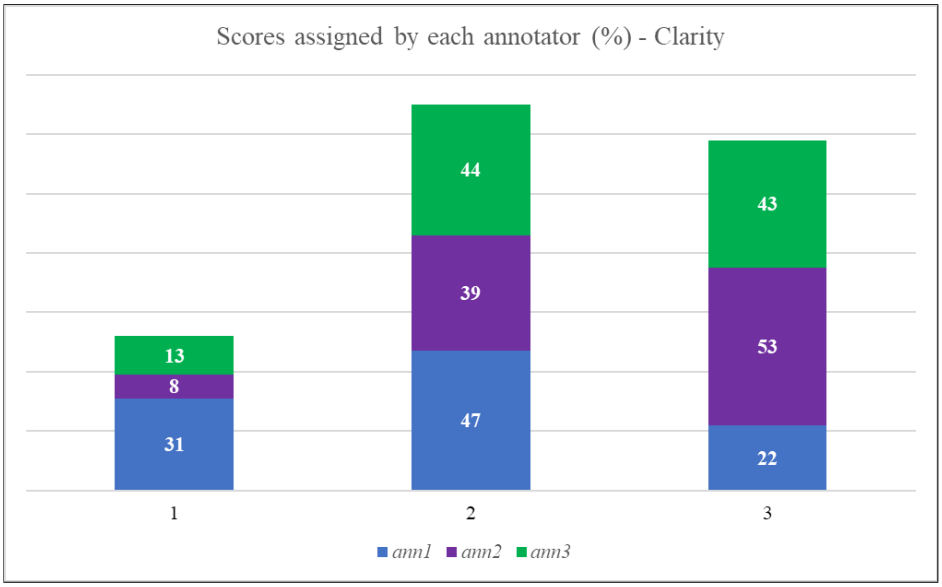


Figure 8.1: Percent distribution of *clarity* scores by annotator.

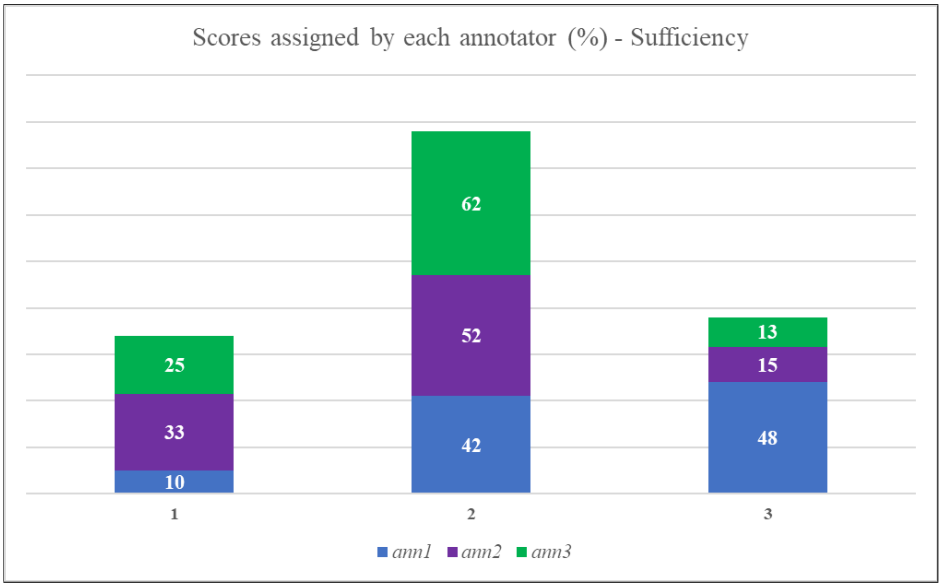


Figure 8.2: Percent distribution of *sufficiency* scores by annotator.

8.2.1 Agreement and confidence scores

In order to evaluate the level of inter-annotator agreement we compute Krippendorff's α coefficients (Krippendorff, 2007) for overlapping SciARG-CL annotations–both globally and pairwise.

Dimension	**Overall**	$ann_1 - ann_2$	$ann_2 - ann_3$	$ann_1 - ann_3$
Clarity	0.33	0.34	0.29	0.33
Sufficiency	0.27	0.37	0.08	0.26

Table 8.2: Agreement in SciARG argumentative quality annotations: Krippendorff's overall and pairwise αs.

Even if moderate, the overall level of agreement for the *clarity* and *sufficiency* dimensions are acceptable for these tasks and comparable to the results obtained for larger-scale argumentative quality annotation efforts in other domains (Ng et al., 2020).

Full discrepancy between the annotators–i.e., each annotator assigning a different score–is observed only in 2 cases (0.07%) for *clarity* and in no cases for the *sufficiency* dimension.

We use MACE (Multi-Annotator Competence Estimation)[2] (Hovy et al., 2013) both to obtain a unique score for each quality dimension in the overlapping annotations and to attach a *reliability score* to each annotator. The results are shown in Table 8.3.

Dimension	ann_1	ann_2	ann_3
Clarity	0.70	0.37	0.31
Sufficiency	0.96	0.29	0.12

Table 8.3: MACE confidence scores for annotators.

In line with the differences in confidence scores, the resulting MACE scores coincide, in fact, with ann_1's annotations for both dimensions.

[2] `isi.edu/publications/licensed-sw/mace/`

The high degree of subjectivity involved in the assignment of quality scores poses a challenge when using these annotations to train machine learning models. A question that emerges is how can we feed the models with reliable enough data so they can learn how to produce useful predictions. If we had multiple predictions for all instances, an alternative could be to use majority annotations for training (although depending on the task and distribution of the labels we might risk approximating all the scores to a middle point).

Given the rather small size of our dataset for these tasks and the fact that only a small fraction of it was annotated by more than one annotator, we propose to incorporate information about the annotators and/or their reliability into the models. We compare two approaches:

- To provide the models with information about *who* (which annotator) assigned the score to be predicted. The idea is that the models, in the training phase, might be able to learn the differences in criteria followed by the different annotators.
  This information would be included by means of an additional *feature* indicating the annotator of each instance.
  For prediction/evaluation, the value of this feature would be constant and represent the most reliable annotator.

- To explicitly *weight* training instances according to the annotator that produced them. The machine learning algorithms implemented in this chapter can handle weighted instances. If this were not the case, this is equivalent to over-sampling or under-sampling instances in the training set proportionally to the weight that we want them to have.
  The computation of the weights is based on the annotators' confidence scores, as described below.

8.3 Experimental setup

For the prediction of the argumentative quality scores we consider a set traditional machine learning algorithms: *Naïve Bayes* with and without kernels (NB_K and *NB*, respectively), *Decision lists* (*DL*),[3] *Nearest neighbours* (*NN*), *Random forests* (*RF*) and *Support Vector Machines* (*SVM*). These algorithms are suited for the size of the corpus–as for these tasks each abstract gives origin to one single training instance–and have the advantage of being more interpretable than algorithms with a larger number of parameters. In these experiments we use implementations available in the *Weka* software package (Witten et al., 2016).[4]

How to best model tasks with ordinal scales is an open issue and greatly depends on the semantics of the scale. We choose to model the score prediction task as three-class classification rather than a regression problem, as we do not attach any meaning to in-between values nor impose any assumption on the equivalence of the distances between adjacent scores. We therefore consider the scale as a discrete qualitative measure rather than continuous quantitative one. This allows us to make use of simple algorithms and evaluate them with metrics that are easily interpretable such as F_1 score. Other alternatives that would allow to exploit the ordering between classes would have been to transform the task into two binary classification problems or to implement cost-sensitive algorithms, so misclassification of instances between classes *1* and *3* were highly penalized. We did some preliminary attempts in this direction without obtaining significant improvements, but further exploration of these alternatives is interesting and left as future work.

As the goal of these experiments is not to obtain the best possible performing models for the prediction of quality scores but, instead, to assess the potential benefits of using features derived from the argumentative structure of the abstracts, we fix the hyper-parameters for all the algorithms across this set of experiments.

For these experiments we consider 36 features derived from the argumentative structure of the abstracts.[5]

[3] As decision list classifier we use the PART algorithm introduced in (Frank and Witten, 1998)

[4] `cs.waikato.ac.nz/ml/weka/`

[5] In a previous pilot experiment–for the prediction of recommendations by peer-reviewers–we used directly the sequences of labels predicted for each of the argument mining tasks–type of units,

The features include:

- Length (number of characters) (1 feature)
- Number of children of the main unit (1 feature)
- Number of units for each type of unit (coarse and fine-grained types) (16 features)
- Number of children for each type of unit (11 features)
- Number of relations for each type of relation (7 features)

Different combinations of the features can be considered in order to capture different aspects of the argumentative quality. For instance, the number of *support* relations can be considered to provide relevant information about the *sufficiency* dimension while the number of *elaboration* relations could be more closely related to the *clarity* of the abstract.

We train models with the full set of features as well as with subsets that convey information about the quality dimensions being considered. In total, we consider 37 potential combinations of argumentative structure features.

It is relevant to note that the length of texts has been observed to correlate strongly with some argumentative quality dimensions (Potash et al., 2017; Wachsmuth and Werner, 2020). We are therefore interested in evaluating the extent to which argumentative structure features can contribute to improve predictions that are based on the length.

For each algorithm we consider 76 potential combination of features:

- Each of the 37 subsets of features considered to provide argumentative quality information with and without including the length (74 subsets);
- The length alone;
- The full set of features.

We consider as baselines a majority-class classifier and a rules-based classifier with the length of the abstract as sole feature.

position of the parent and type of the relation (Accuosto and Saggion, 2019b).

As mentioned, we would like our models to perform as similarly as possible to the annotator with the highest confidence score.

We have, in our dataset, the following distribution of annotations:

Annotator	**Individual**	**Overlapping**	**Total**
ann_1	80	30	110
ann_2	77	30	107
ann_3	38	30	68
Total	195	30	225

Table 8.4: Number of individual and overlapping annotations by each annotator in the SciARG-CL corpus.

We train and evaluate the models in a five-fold cross validation setting, which ensures more stable results than evaluating only on one validation set, and allows us to use the whole dataset for training.

We use the 110 annotations made by ann_1 (the most reliable annotator) both for training and evaluation, while the annotations by annotators ann_2 and ann_3 are used only for training, weighted according to their respective confidence score.

We generate the training-validation sets for each of the five folds stratified by annotator and class, obtaining the following distribution of instances:

Fold	**Training instances**			**Total training**	**Validation instances**
	ann_1	ann_2	ann_3		ann_1
1	89	61	30	180	21
2	87	62	31	180	23
3	87	61	32	180	23
4	88	62	30	180	22
5	89	62	29	180	21

Table 8.5: Number of training/test instances by annotator in each fold.

In order to compute weights for each annotator's instances, we consider the confidence scores obtained by MACE and the number of annotations present in the training set. As mentioned, another way of looking at this is to consider that we are down-sampling annotations by ann_2 and ann_3 so for each annotation by ann_1 there is only a fraction of annotations by ann_2 and ann_3, in proportion to their respective confidence scores.

For instance, suppose that we had two annotators, ann_a and ann_b and that ann_b's annotations are half as reliable as the annotations made by ann_a. If we had the same number of annotations by both annotators in the training set, we would need to weight ann_b's annotations with 0.5, to indicate that every *two* annotations by ann_a, it should consider *one* annotation by ann_b.
If in the original set of annotations we do not have the same number of annotations by both ann_a and ann_b, but instead have twice as many annotations by ann_b as the number of annotations by ann_a, we should take this factor into consideration and assign a weight of 0.25 to ann_b's annotations to make sure that their overall weight reflect the fact that they are half as reliable than ann_a's.

We therefore calculate the weight W_{ann_x} for annotations by a given annotator ann_x, with the following formula:

$$W_{ann_x} = \frac{Nw_{ann_x}}{Nt_{ann_x}} \;\; where \;\; Nw_{ann_x} = \frac{Nt_{mra}}{Cf_{ann_x}} \;\; ; \;\; Cf_{ann_x} = \frac{Cs_{mra}}{Cs_{ann_x}}$$

Which can be interpreted in the following way:

$Cf_{ann_x}-$ Confidence correction factor for ann_x;

$Cs_{mra}-$ Competence score of the most reliable annotator (mra);

$Cs_{ann_x}-$ Competence score of ann_x;

$Nt_{mra}-$ Number of annotations by the most reliable annotator in training set;

$Nt_{ann_x}-$ Number of annotations by ann_x in training set.

Cf_{ann_x} is a correction factor that tells us how many annotations by the most reliable annotator (mra) should be in the training set for each annotation of ann_x.

For instance, if ann_i is half as reliable as mra, $Cf_{ann_i} = 2$, meaning that there should be two instances by mra for each instance of ann_i.

Nw_{ann_x} tells us how many annotations by ann_x should be in the training set considering ann_x's correction factor and the number of instances of mra.
For instance, if ann_i is half as reliable as mra, and there are 10 annotations by mra, there should be $\frac{10}{2} = 5$ annotations by ann_i.

Finally, W_{ann_x} tells us how much each annotation by ann_x should weight considering how many annotations there are effectively in the training set and how many there should be.[6]
For instance, if ann_i is half as reliable as mra, there are 10 annotations by mra, and there are 20 annotations by ann_i, each annotation by ann_i should weight $\frac{5}{20} = 0.25$.

8.4 Results and analysis

As the classes are not perfectly balanced we report, in each case, macro-averaged *F1-scores*, which provides information about how well the classifier performs for the minority class.

In order to simplify the analysis and have a clear picture of the impact of the different combinations of features we consider, in addition to the baseline classifiers, the results obtained with the overall three best performing algorithms, with their respective combination of features.

8.4.1 Clarity

As expected, introducing information about the annotators or their reliability contributes to improve the models' performance when predicting the most reliable score.

[6] In practice, we use 10 as weight for the annotations made by ann_1 (the most reliable annotator) and $10.W_{ann_x}$ for instances annotated by ann_2 or ann_3.

Clarity scores without weights nor *annotator* feature		
Algorithm	F_1	**Features**
Majority	0.2140	–
$Rule_L$	0.3598	*length*
NB	**0.5878**	*child-main, prop-impl, elab, seq*
NB_K	0.5488	*child-main, prop-impl, elab, seq, supp, child-prop*
NN	0.4936	*child-main, prop-impl, elab, seq*

Table 8.6: Results for *clarity* five-fold CV classification with no weights nor annotators. Top-performing classifiers: *Naïve Bayes* with and without kernels, *Nearest neighbours*.

Clarity scores with weights			
Algorithm	F_1	Δ	**Features**
Majority	0.2140	–	–
$Rule_L$	0.3623	+ 0.0025	*length*
NB	**0.5990**	+ 0.0112	*length, prop-impl, elab, seq*
NB_K	0.5853	+ 0.0365	*length, prop-impl, elab, seq, supp, means*
NN	0.5345	+ 0.0409	*prop-impl, elab, seq*

Table 8.7: Results for *clarity* five-fold CV classification with weighted instances. Δ indicates the difference with respect to the models with no weights nor annotators.

Clarity scores with *annotator* feature			
Algorithm	F_1	Δ	**Features**
Majority	0.2140	–	–
$Rule_L$	0.3598	–	*length*
NB	**0.6129**	+ 0.0251	*length, prop-impl, elab, seq, supp, ch-resu*
NB_K	0.5894	+ 0.0406	*prop-impl, elab, seq*
NN	0.5335	+ 0.0399	*prop-impl, elab, seq*

Table 8.8: Results for *clarity* five-fold CV classification with *annotator* feature. Δ indicates the difference with respect to the models with no weights nor annotators.

In particular, for the *clarity* task, explicitly including the annotators as a feature improves in 4% the macro-averaged F_1 score for the best performing algorithm. The performance gain when weighting the training instances is lower (2%) for the best performing algorithm. As expected, the gain in performance when considering weights and/or annotators as a feature is greater for the algorithms with initial lower performances.

A small number of the 36 considered features intervene in the best performing configurations: in particular, the number of units of type *proposal-implementation* and the number of relations of types *elaboration* and *sequence* are present in all cases. These relations are used to provide more detailed information (in terms of implementation) about the solutions proposed by the authors.

The number of children of units of type *proposal*, which is present in one of the configurations, conveys similar information. In turn, the number of children of units of type *result*–and, in particular, those of type *means*–are used to provide precise information about methods and/or resources used to obtain the results. It is expected, then, that these features are predictive of the perceived clarity in which the authors explain what was done and why. The fact that this is the case confirms that information provided by the analysis of the argumentative structure of the abstracts contribute to predict this quality dimension.

It is interesting to observe that the number of children of the main unit is present in the best three configurations only in the models with no information about the annotators. This can be explained considering the differences in the correlation of this feature with the class depending on the annotator. This feature is correlated with the class with $r = 0.50$ $(p < 0.001)$ for ann_3, but only with $r = 0.19$ (in the limit of significance) for annotator ann_1, and with $r = 0.22$ $(p < 0.005)$ for ann_2. It is natural, then, that when the number of instances of ann_3 are down-weighted this feature loses relevance. It is also possible that when no information about the annotators is present, this feature can contribute to discriminate instances by different annotators, which would no longer be relevant when this information is provided explicitly.

The length of the abstracts is positively correlated with its perceived clarity for all the annotators, even if this feature by itself does not discriminate well between the predicted classes. The correlation coefficient for ann_1 is $r = 0.49$, for ann_2, $r = 0.42$ and, for ann_3, $r = 0.33$ (in all cases, with $p < 0.001$).

Conversely to what happens with the number of children of the main unit, it is expected that the length gains in relevance as a discriminating factor as the models are instructed to give more weight to the annotations by ann_1.

8.4.2 Sufficiency

Sufficiency scores without weights nor *annotator* feature		
Algorithm	F_1	**Features**
Majority	0.2275	–
$Rule_L$	0.4111	*length*
NB	**0.5995**	*length, child-prop, motiv-prob*
NB_K	0.5696	*length, child-prop, child-main, motiv-prob, supp*
DL	0.5592	*child-prop, motiv-prob*

Table 8.9: Results for *sufficiency* five-fold CV classification with no weights nor annotators. Top-performing classifiers: *Naïve Bayes* with and without kernels, *Decision lists*.

Sufficiency scores with weights			
Algorithm	F_1	Δ	**Features**
Majority	0.2275	–	–
$Rule_L$	0.4111	–	*length*
NB	**0.6185**	+ 0.0190	*length, child-prop, motiv-prob, supp*
NB_K	0.6092	+ 0.0396	*length, child-prop, motiv-prob*
DL	0.5738	+ 0.0146	*coarse-motiv, motiv-prob*

Table 8.10: Results for *sufficiency* five-fold CV classification with weighted instances. Δ indicates the difference in performance with respect to the models with no weights nor annotators.

Sufficiency scores with *annotator* feature			
Algorithm	F_1	Δ	**Features**
Majority	0.2275	–	–
$Rule_L$	0.4111	–	*length*
NB	**0.6052**	+ 0.0057	*length, child-prop, motiv-prob, supp*
NB_K	0.5776	+ 0.0080	*child-prop, child-main, coarse-motiv, motiv-prob, supp*
DL	0.5571	- 0.0021	*child-prop, child-main, coarse-motiv, coarse-out, supp*

Table 8.11: Results for *sufficiency* five-fold CV classification with *annotator* feature. Δ indicates the difference in performance with respect to the models with no weights nor annotators.

For the prediction of the *sufficiency* dimension we observe, again, that a small number of features give origin to the best performing models. As expected, the number of *motivation* units, in general (as expressed by the *coarse-motivation* type)–and, in particular, those indicating existing problems addressed by the proposed solution–are consistently relevant to predict the perceived sufficiency of the abstracts.

The number of relations of type *support* appear in some of the best performing subset of features for the prediction of the *clarity* dimension, but the weight of this feature in the prediction of the perceived *sufficiency* is more evident–in particular, when information about the annotators or their relevance is included in the models. It is expected that this feature provides relevant information about the abstracts' perceived sufficiency, as the *support* relation is used precisely to provide evidence for claims. In particular, this feature is highly correlated with the class for annotator ann_2 ($r = 0.56$, with $p < 0.001$) and it has also a positive correlation for ann_1 ($r = 0.38$, with $p < 0.001$). For ann_3 the correlation is less significant ($r = 0.28$, with $p < 0.005$). It is therefore natural that this feature gains in relevance when information about the annotators' confidence is reflected in the training instances.

The number of children of the main unit and of all units of type *proposal* also play an important role in the prediction of the abstracts' sufficiency, as would be expected, since these units provide explanations and/or justifications for the main explicit or implicit claims of the abstract. In this case, and in contrast to what happens in the *clarity* dimension, the *sufficiency* score and the number of children of the main unit are more similarly correlated across annotators, with Pearson's correlation coefficients of $r = 0.35$, $r = 0.45$ and $r = 0.45$ (with $p < 0.01$) for ann_1, ann_2 and ann_3, respectively.

Another difference that we can see between the prediction of the clarity and sufficiency scores is that, in the latter, the greatest improvement in performance is not obtained with the explicit inclusion of the annotators as a feature but, instead, when instances are weighted according to the annotators' reliability. This could be explained by the significant differences in inter-annotator agreements for the *sufficiency* dimension, as observed with the pairwise Krippendorff's α coefficients and the MACE confidence scores. This would indicate that the information available, including the identification of the annotators, is not enough for the models to learn the differences in criteria and, instead, it is more important in this case to explicitly weight (or, equivalently, to up or down-sample) the instances according to their reliability.

The best performing algorithm for both quality dimensions is *Naïve Bayes*, while other simple algorithms like *Decision lists* and *Nearest neighbours* classifiers also perform well for sufficiency and clarity, respectively. This might be explained by the fact that *Naïve Bayes* can produce good results with small-sized datasets without much hyper-parameter fine-tuning. In these experiments, as mentioned, we focused more on the set of features than on the algorithms themselves, so additional investigation would be required to determine whether better results could be obtained for other algorithms with more data and/or other hyper-parameter values.

8.5 Conclusions

In this chapter we explored whether argumentation-level annotations from the SciARG-CL corpus could be used to obtain features that reflect argumentative quality dimensions of the abstracts. In particular, we focused on the assessment of the abstract's *clarity* and *sufficiency* dimensions. To do this, we considered scores

assigned by annotators for these two dimensions and trained machine learning models with different subsets of features that convey information about the components found in the abstracts and the relations between them. The results obtained show that, in fact, argumentation-informed features can significantly contribute to predict the abstracts' perceived clarity and sufficiency. Moreover, the better-performing sets of features in each case can be considered to convey information that is in line with the argumentative quality dimension intended to be assessed.

We also considered the difficulties posed by the high level of subjectivity involved in the annotation of argumentative quality dimensions, which is reflected in high differences in the levels of pairwise inter-annotator agreements, as well as in the confidence scores assigned by the MACE algorithm (Hovy et al., 2013) to annotators. We proposed to deal with differences in the reliability of the annotations either by weighting the training instances with a factor that takes into account the annotators' confidence score, or by explicitly including information about annotators in the training instances–as an additional feature. The results obtained show that these strategies can in fact contribute to improve the performance of the models when predicting the scores assigned by the most reliable annotator. This leads to potentially relevant follow-up research, in order to determine whether this approach could be extended to other cases in which annotations are produced by annotators with different levels of reliability.

As a final observation, it is relevant to consider that, when analyzing the correlations between the features and the quality scores assigned to the abstracts it does not look as if "automatic" decisions were made by the annotators. Nevertheless, we cannot exclude the possibility that the annotators' perception of the abstracts' *clarity* and *sufficiency* could somehow be biased by their own analysis of the argumentative structure of the texts–even when they were asked to assign these scores as a first step in the annotation process.

In the experiments included in this chapter we used gold annotations of argumentative units and relations to produce the features used to train and evaluate the models. In the next section we explore whether features obtained by the predicted argumentative structure of the abstracts can contribute to anticipate quality scores. In this case, the scores considered are those assigned by reviewers in a peer-review process.

Chapter 9

PREDICTING PEER REVIEW SCORES: CLARITY, SOUNDNESS AND OVERALL RECOMMENDATIONS

In this chapter we investigate whether argumentative information automatically extracted from the abstracts of the manuscripts conveys information that can contribute to predict fine-grained and overall recommendation scores assigned by reviewers in a peer-review process.[1]

We address these questions by:

- Automatically extracting, from abstracts included in the ACL, CoNLL and ICLR subsets of PeerRead, the argumentative-structure features described in Section 8.3.
- Identifying, from seven manuscript assessment areas included in the ACL 2017 review form,[2] those that capture argumentative quality information and are feasible of being predicted by means of the argument-based features considered for SciARG-CL in Chapter 8.

[1]The goal of this chapter is, therefore, not to produce models to be used for the automatic prediction of argumentative quality dimensions of research manuscripts.

[2]The form used in reviews for ACL 2017 is available as an appendix in (Kang et al., 2018)

- Conducting experiments aimed at predicting the identified argumentative quality scores for the ACL and CoNLL subsets of PeerRead, as well as overall recommendation scores assigned by reviewers for the ACL, CoNLL and ICLR subsets.

This chapter is organized as follows:

- In Section 9.1 we briefly describe the PeerRead dataset and, in particular, the sub-sections used in the experiments reported in this chapter;
- In Section 9.2 we identify, from the scores assigned to manuscripts in ACL-ConLL, the ones that can be considered to convey argumentative information: *clarity* and *soundness*. We conduct experiments aimed at predicting a weighted average of the scores assigned by peer-reviewers and analyze the results obtained. We finally consider, from a qualitative point of view, similarities and differences found when predicting argumentative scores in SciARG-CL and ACL-CoNLL;
- In Section 9.3 we address the prediction of overall recommendation scores assigned by reviewers in the ACL-CoNLL and ICLR datasets. We propose the prediction of these scores as binary classification tasks, describe the implemented experiments and its results, and analyze differences observed between the outcomes of the experiments in both subsets.
- In Section 9.4 we summarize the main results obtained. Finally, we briefly discuss the limitations of the experiments conducted in this chapter and how they should be contextualized according to the objectives pursued in this work.

9.1 The PeerRead dataset

The PeerRead dataset (Kang et al., 2018) contains manuscripts submitted to computational linguistics and machine learning conferences (NIPS 2013-2017, ACL 2017, CoNLL 2017 and ICLR 2017) as well as papers published on the arXiv platform between 2007 and 2017. The papers submitted to peer-reviewed venues include also their corresponding reviews: opted-in, in the case of ACL and CoNLL,

and available on the OpenReview platform[3] in the case of ICLR. For NIPS, the reviews are available together with the papers in the conference proceedings (therefore, the available reviews correspond only to accepted papers).

The ACL, CoNLL and ICLR subsets include drafts and reviews both for accepted and rejected papers and include recommendation scores assigned by the reviewers: a numeric score between 1-5 for ACL and CoNLL and between 1 and 10 for ICLR. Reviews for these datasets also contain reviewers confidence scores in the range 1-5 for ACL and CoNLL and in the range 1-10 for ICLR. The ACL and CoNLL subsets contain, in addition to the overall recommendation, scores for seven specific areas of assessment.[4] In the case of ICLR, the original reviews do not contain fine-grained scores assigned by reviewers but the scores were added in the corpus development process by two of the PeerRead paper's authors based on the contents of the reviews. We will only use in our experiments the scores directly assigned by reviewers.

Venue	Manuscripts	Reviews	Accepted/Rejected
ACL 2017	137	275	88/49
CoNLL 2016	22	39	11/11
ICLR 2017	427	1304	172/255

Table 9.1: Distribution of ACL, CoNLL and ICLR manuscripts and reviews in the PeerRead dataset. Source: (Kang et al., 2018).

9.2 Prediction of argumentative quality aspects

The abstracts of scientific manuscripts can provide insights of what a reviewer can expect to find in the full paper in terms of its *relevance* for a particular venue, its *originality*, or even its potential *impact*. These aspects are part of the items that referees are asked to assess in a peer-review process. Most of these elements, nevertheless, are aimed at assessing the contents of the paper. For our experiments we are interested in considering the aspects included in reviews–and, in particular, the scores assigned to them–that can be linked to argumentative quality dimensions.

[3] `openreview.net`

[4] in (Kang et al., 2018) and in ACL's instructions for reviewers these assessment areas are referred to as *aspects*

The questions included in the ACL 2017 review form address seven assessment areas: *impact*, *substance*, *appropriateness*,[5] *comparison*, *soundness/correctness*, *originality* and *clarity*. For each of these areas we consider whether i) they reflect, to some extent, theory-motivated argumentative dimensions considered in (Wachsmuth et al., 2017a), and ii) whether it is reasonable to assume that the information contained in the abstract is aligned–in terms of its quality and quantity– with the information contained in the body of the paper for the considered area. The two ACL review areas that come closest to what we are looking for are *clarity* and *soundness/correctness*. The specific questions included in the instructions for ACL reviewers for the assessment of these two aspects of the manuscripts are:

> ***Clarity***:
>
> > *For the reasonably well-prepared reader, is it clear what was done and why?*; *Is the paper well-written and well-structured?*
>
> ***Soundness / correctness***:[6]
>
> (a) *Theoretical: Is the technical approach sound and well-chosen?; Can one trust the empirical claims of the paper – are they supported by proper experiments and are the results of the experiments correctly interpreted?*
>
> (b) *Empirical: Is the mathematical approach sound and well-chosen?; Are the arguments in the paper cogent and well-supported?*

We can observe that, even when there are clear overlaps between these ACL assessment areas and dimensions considered in the taxonomy by Wachsmuth et al. (2017a), no perfect mappings can be established. While the *soundness / correctness* questions assess, to some degree, all of the *cogency* dimensions of the arguments contained in the manuscript, the *clarity* questions are intended to assess the *clarity* and the *arrangement* of the texts but not other specific dimensions such as their *credibility*, *emotional appeal* or *appropriateness*.

[5] *Appropriateness* here refers to whether the paper would fit in the venue and has no relation to the type of language used by the authors.

[6] While this is separated into *theoretical* and *empirical* soundness/correctness in the ACL 2017 form for reviewers, the PeerRead data contains only one score for this area, which is the one that we use in our experiments.

9.2.1 Experimental setup

For this set of experiments we adopt the same experimental setup used for the experiments with the SciARG-CL corpus, including the considered sets of features, as well as the algorithms used and their hyper-parameters.[7] The difference is that, in this case, the features are obtained from the predicted argumentative structures of the abstracts obtained by means of sentence-level models trained with the SciARG-CL corpus in a multi-task setting and pre-fine-tuned with SciDTB sentence-level tasks, as described in Chapter 4.

As mentioned, only the ACL 2017 and CoNLL 2016 subsets of PeerRead contain scores for each assessment area assigned directly by the reviewers so we are using these two subsets of PeerRead for the experiments described in this section. As the information available for both subsets is the same, and there are only 22 CoNLL manuscripts included in PeerRead, we consider these sets jointly for our experiments. In Table 9.1 we include the number of manuscripts and reviews in each subset as well the number of accepted/rejected papers.[8]

Each manuscript in the ACL and CoNLL sections of PeerRead contains between one and three reviews. In addition to providing an integer score between 1 and 5 for their overall recommendation and each considered area, reviewers are asked to indicate their own confidence about their evaluation, also with scores from 1 to 5. We use these confidence scores to obtain weighted averages, which we then use to place each manuscript in a classification bin as described below.

Fig. 9.1 shows the distribution of scores for *clarity*, *soundness* and *overall recommendation* in all of the ACL-CoNLL reviews.

Suppose, for instance, that three reviewers r_1, r_2, r_3 assigned the scores $s_1 = 4$, $s_2 = 3$, $s_3 = 2$ to a manuscript, and reported confidence scores $c_1 = 2$, $c_2 = 5$, $c_3 = 4$.

Naturally, given the discrepancy of criteria, the score assigned by the reviewer that reported the highest confidence (r_2) should have a greater weight when computing the final score.

[7] We use the same set of fixed hyper-parameters.

[8] As reported in (Kang et al., 2018), as acceptance/rejection information for this subset is not available in the published data.

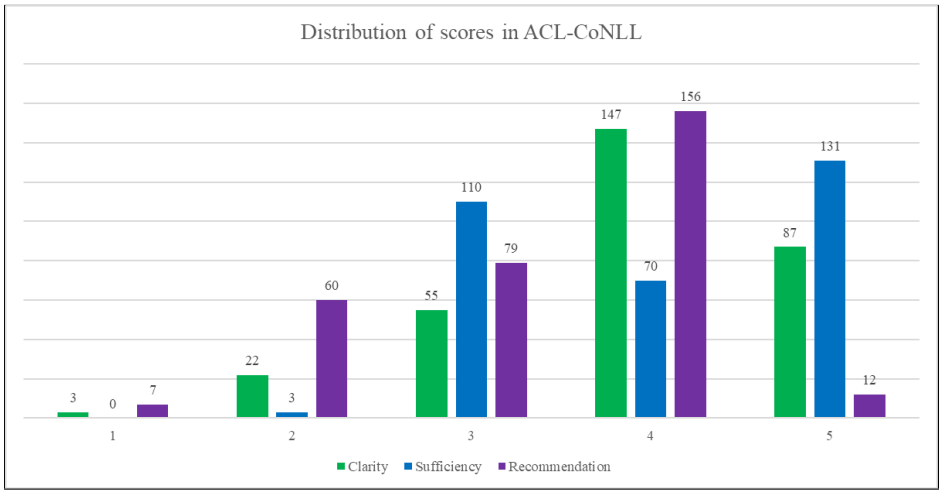


Figure 9.1: Distribution of scores for *clarity*, *soundness* and *overall recommendation* scores in ACL-CoNLL.

To compute the weighted average W_s for score s in a given manuscript m, we use the following formula:

$$W_s = \sum_{r=1}^{nr_m} W_r . s_r$$

Where nr_m is the number of reviews available for the manuscript m, s_r is the score assigned by reviewer r, and W_r is the reviewer's confidence weight, computed as:

$$W_r = \frac{c_r}{\sum_{i=1}^{nr_m} c_i} \quad \text{where } c_i \text{ is the confidence value reported by reviewer } i$$

In the example of the three reviewers mentioned above, the reviewers' weights are computed:

$$W_{r_1} = \frac{2}{11} = 0.18; \quad W_{r_2} = \frac{5}{11} = 0.45; \quad W_{r_3} = \frac{4}{11} = 0.36$$

And the final score:

$$W_s = 0.18 \times 4 + 0.45 \times 3 + 0.36 \times 2 = 0.73 + 1.36 + 0.73 = 2.82$$

Figures 9.2 and 9.4 show the distribution of scores averaged with the described weighting method for the *clarity* and *soundness* dimensions, respectively.

To have a detailed picture of how the scores are distributed, we consider the number of manuscripts whose weighted average falls within 0.25-width ranges from 1 to 5.

We observe in Fig. 9.2 that, for *clarity*, the distribution is left-skewed and a large percentage (33%) of the manuscripts fall in a small range between 4 and 4.25 points.

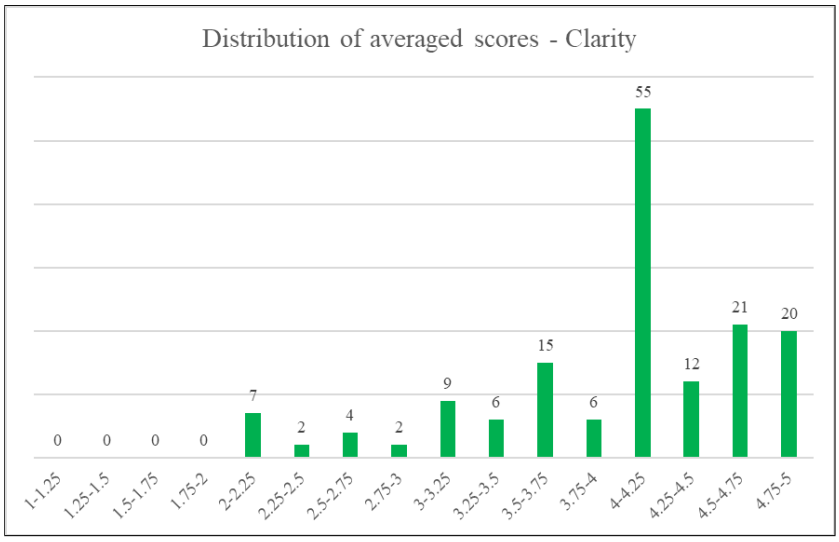


Figure 9.2: Distribution of *clarity* average scores weighted by reviewers' confidence in ACL-CoNLL.

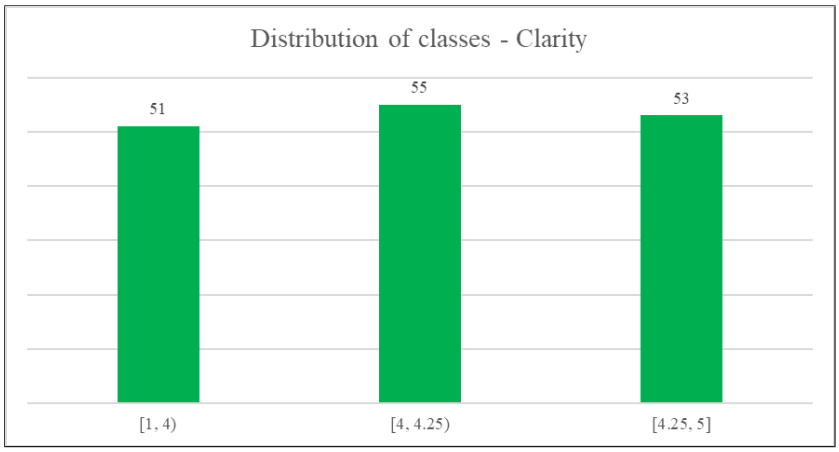


Figure 9.3: Final distribution of classes for *clarity* scores.

The percentages of manuscripts falling to the left or to the right of this range are mostly balanced, with 32% and 33% of the manuscripts, respectively. As mentioned in Section 8.3, we do not believe that such a fine-grained distinction between different values is relevant–and very unlikely to be predictable with the available data. We therefore distribute the manuscripts according to their average scores into three classification bins, seeking to establish cutting points that make the resulting classes as balanced as possible. The resulting distribution of manuscripts in their classes is shown in Fig. 9.3.

Formulating the task as a three-class classification problem allows us to adopt the same experimental setup that we used in the prediction of argumentative quality dimensions in the SciARG corpus and qualitative compare the results between both experiments–even if, as mentioned, the meaning of the dimensions assessed in each case do not perfectly coincide.

We proceed analogously with the *soundness* scores. The fine-grained distribution of the averaged scores is shown in Fig. 9.4. We can observe, again, a left-skewed distribution in this case but, unlike the previous situation, the bottom, middle and top values are all concentrated in three small ranges. It is, therefore, even more natural to formulate the prediction of *soundness* scores as a three-class classification problem.

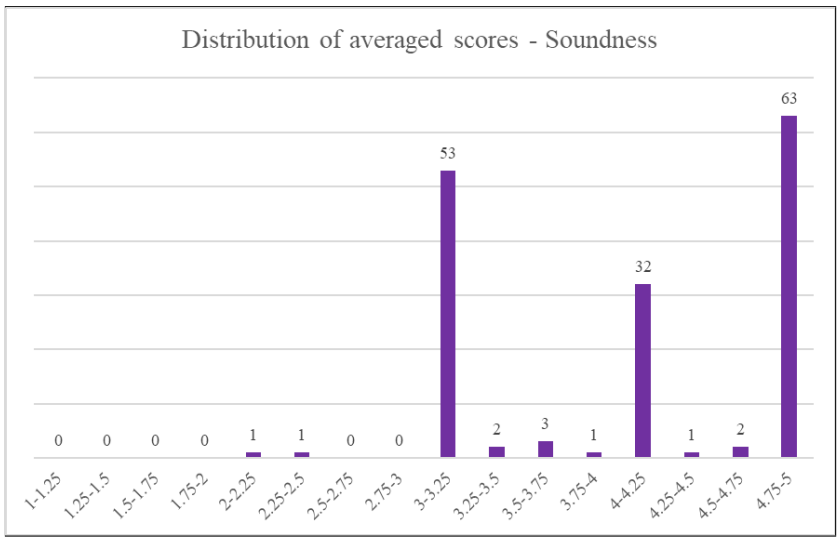


Figure 9.4: Distribution of *soundness* average scores weighted by reviewers' confidence in ACL-CoNLL.

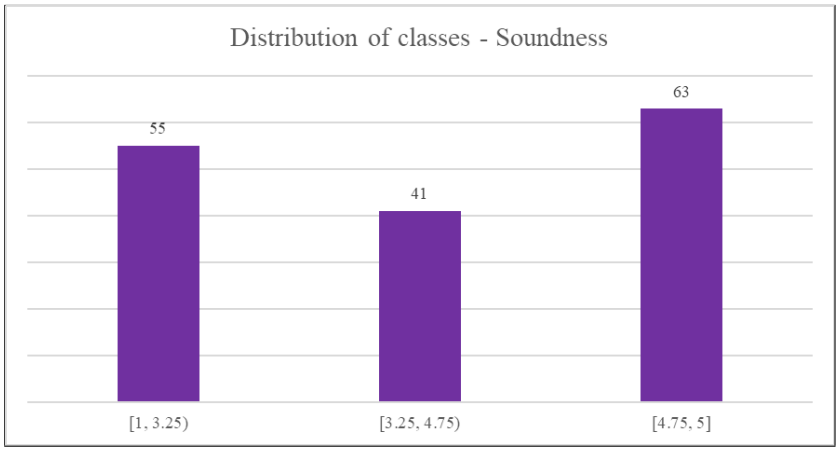


Figure 9.5: Final distribution of classes for *soundness* scores in ACL-CoNLL.

Unlike what happens with the *clarity* scores, the resulting classes are more poorly balanced in the *soundness* tasks, with the majority of the manuscripts being assigned to the top-valued class.

9.2.2 Results and analysis

As in the previous experiments, we analyze the performance of different combinations of algorithms and features by considering macro-averaged F_1 scores resulting from five-fold cross-validation experiments.

9.2.2.1 Clarity

Table 9.2 shows the results obtained for the prediction of the *clarity* score.

Similarly as what we observe in the experiments carried out with the SciARG-CL corpus, there are also small sets of features that contribute to the best performing models. The number of *proposal-implementation* units seem again to be relevant for the prediction of the *clarity* dimension–in particular, in combination with the length. The number of *elaboration*, *sequence* and *support* relations also

Features derived from the number of units that provide details about reported *results*–in particular, the number of units of type *observation*–seem to have a relevant weight in the prediction of the *clarity* score.

Clarity scores for ACL-CoNLL		
Algorithm	F_1	**Features**
Majority	0.1721	–
$Rule_L$	0.3132	*length*
SVM	**0.4355**	*length, prop-impl, obs, result, supp*
NN	0.4289	*coarse-motiv, motiv-problem*
DL	0.4118	*length, prop-impl, obs, means, supp, elab, seq*

Table 9.2: Results for *clarity* five-fold CV classification. Top-performing classifiers: *Support Vector Machine*, *Nearest neighbours*, *Decision list*.

As mentioned, the meaning of *clarity* in ACL covers other dimensions of the argumentative *effectiveness* of the texts. This could explain the apparently greater weight of the number of *support* relations for this task and, in particular, the fact that a *nearest neighbours* algorithm performs close to the top, based solely on the number of *motivation* units.[9] It is not evident, nevertheless, the difference in the selection of the best features for this particular algorithm, while there is a greater overlap in the set of preferred features for SVM and *decision lists*.

9.2.2.2 Soundness

Soundness scores for ACL-CoNLL		
Algorithm	**Macro F1**	**Features**
Majority	0.1900	–
$Rule_L$	0.2948	*length*
NN	**0.4498**	*child-main, child-prop, supp, coarse-motiv, coarse-out*
RF	0.4414	*child-main, child-prop, supp, coarse-motiv, coarse-out*
DL	0.4267	*child-main, child-prop, supp*

Table 9.3: Results for *soundness* five-fold CV classification. Top-performing classifiers: *Nearest neighbours*, *Random Forest*, *Decision list*.

[9] In SciARG these features play a relevant role for the prediction of the *sufficiency* score.

We observe a great overlap between the set of features that give rise to the best performing algorithms when predicting SciARG's *sufficiency* scores and the features for which the best configurations are obtained for ACL-CoNLL's *soundness*.

All of the involved features indicate how well proposals–and, in particular, the main proposal of the abstract–are supported by premises.

Units of type *motivation*–in general–seem to play a relevant role in the perceived quality of this dimension. This is to some point expected as, in the case of scientific texts, authors persuade the potential readers about the validity and relevance of the proposed solutions in part referring to background information and describing existing unsolved problems.

9.2.2.3 Comparison with SciARG argumentative quality predictions

Even if the quality dimensions in SciARG cannot be mapped to the aspects considered in ACL reviews, it is interesting to compare–from a qualitative perspective–the respective gains obtained for SciARG and ACL-CoNLL scores predicted by means of models trained with argumentation-based features.

Table 9.4 shows the gains obtained with the top performing *argumentative-aware* classifiers in each case, with respect to majority baselines.

	SciARG-CL		**ACL-CoNLL**	
	Clarity	**Sufficiency**	**Clarity**	**Soundness**
Majority classifier (*macro* F_1)	0.2140	0.2275	0.1721	0.1900
Model w. arg. feat. (*macro* F_1)[10]	0.5878	0.5995	0.4355	0.4498
Absolute gain	0.3738	0.3729	0.2634	0.2598
Percent gain	175%	164%	153%	137%

Table 9.4: Comparison of SciARG and PeerRead best *argumentative-aware* models with respect to majority classifier.

It is important to have in mind the differences between the way in which these numbers should be interpreted which, as mentioned, makes it impossible to compare them directly:

- SciARG dimensions are more narrowly defined than ACL assessment areas, and were annotated with the specific objective to assess argumentative quality dimensions of the abstracts;
- SciARG models predict the score assigned by the most reliable annotator, while ACL-CoNLL models predict the average weighted by the reviewers' reported confidence (for the whole review);
- We use gold annotations for argumentative units and relations in SciARG, while in the ACL-CoNLL subset of PeerRead we use predicted argumentative units and relations;
- While SciARG scores directly assess argumentative dimensions of the abstracts, the scores in PeerRead reflect the opinion of the reviewer about the whole manuscript.

Taking these considerations into account, the gains in performance for ACL-CoNLL's models provide an interesting ground for further research. These gains would indicate not only that features obtained from the automatically-extracted argumentative structure of the abstracts can be useful to improve the prediction of the specific quality scores being considered, but also that these quality dimensions in abstracts and in full manuscripts are, up to a certain point, aligned. Additional experiments would be needed to confirm whether the prediction of these dimensions in abstracts could be used as an estimation of what can be expected to occur when assessing the full texts.

9.3 Prediction of recommendation scores

As mentioned, the quality of argumentation is expected to play only a partial role in the overall assessment of manuscripts by reviewers. Yet, it is relevant, in the context of this work, to explore the persuasive potential of the manuscripts' abstracts and, in particular, to what extent their argumentative structure can influence reviewers' overall recommendations.

The reviewers' recommendation scores are some of the main elements considered to determine the acceptance or rejection of a manuscript in a conference. In any conference, yet, acceptance decisions involve not only the quality of the submitted manuscript but several other practical considerations, including the balance in the

number of papers among the different sub-disciplines and topics covered by the conference, as well as the number and quality of all the submitted manuscripts. As the ACL 2017 chairs indicate, *"different areas had different average acceptance scores"*.[11] In fact, the ACL 2017 chairs made it clear that they *"did not use score cutoffs to determine acceptances but instead used scores as a guide towards arguing for paper acceptances"*. Therefore, we prefer to consider as a task predicting the recommendation score assigned by the manuscript reviewers and not whether it was actually accepted or not.

9.3.1 Experimental setup

The PeerRead dataset contains overall recommendation scores assigned by reviewers for the ACL, CoNLL and ICLR subsets. We use these three datasets for our experiments. As with the previous experiments, there is no relevant distinction to be made between ACL and CoNLL manuscripts and reviews (in particular considering the small number of CoNLL manuscripts/reviews available) so we consider them together as one dataset.

Overall recommendations are assigned integer values between 1 and 5 in ACL-CoNLL, like for specific area scores. In contrast, in the case of ICLR, recommendation scores range between 1 and 10. Reviewer confidence scores, yet, follow the same criteria as in ACL-CoNLL and are assigned values between 1 and 5. While in ACL-CoNLL all the manuscripts have three reviews, in ICLR a small percentage (5%) includes four reviews and a yet smaller percentage (1%) includes five reviews. The are some important distinctions to consider between both datasets:

- ICLR contains 2.7 times more manuscripts (and 4 times more reviews) than ACL-CoNLL.
- While for ACL-CoNLL the acceptance rate of the manuscripts included in PeerRead is of 62%, in ICLR it is only of 40%.
- ICLR abstracts have 24% more sentences than those in the ACL-CoNLL dataset. This could indicate that they are argumentatively more complex.

[11] `acl2017.wordpress.com/2017/07/31/a-final-look-at-the-decision-process/`

Based on these distinctions and the different scoring criteria used in ACL-CoNLL and ICLR, we understand that it is better to conduct experiments with each dataset separately and then analyze the results obtained in each case.

For the experiments described in this section we use the same algorithms and hyper-parameters as in the prediction of the argumentative dimensions in SciARG and the fine-grained assessment scores in ACL-CoNLL.

9.3.1.1 Experiments with the ACL-CoNLL dataset

Fig. 9.6 shows the distribution of recommendation scores for the ACL-CoNLL subset, weighted-averaged according to the reviewers' confidence scores, in the same way as do for *clarity* and *soundness* and described in Section 9.2.1.

Similarly to what happens with *clarity* scores, most of the averages (37%) fall within the range [4, 4.25). The difference is that, in the case of the *overall recommendation*, there are few values (only 6%) above this range and most of the other scores (55%) are scattered in the range [1, 3.75).

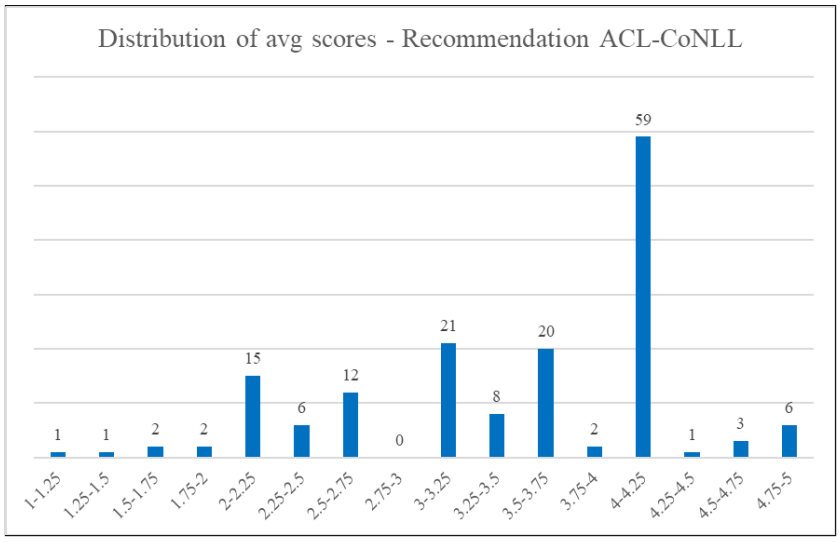


Figure 9.6: Distribution of *overall recommendation* average scores weighted by reviewers' confidence in ACL-CoNLL.

The most natural option in this case, therefore, is to consider two classification bins: the first one for manuscripts with average scores in the range [1, 3.75) and the second one for scores in the range [3.75, 5]. In this way we obtain a split with 55-45 percent of instances in each class, which is shown in Fig. 9.7.

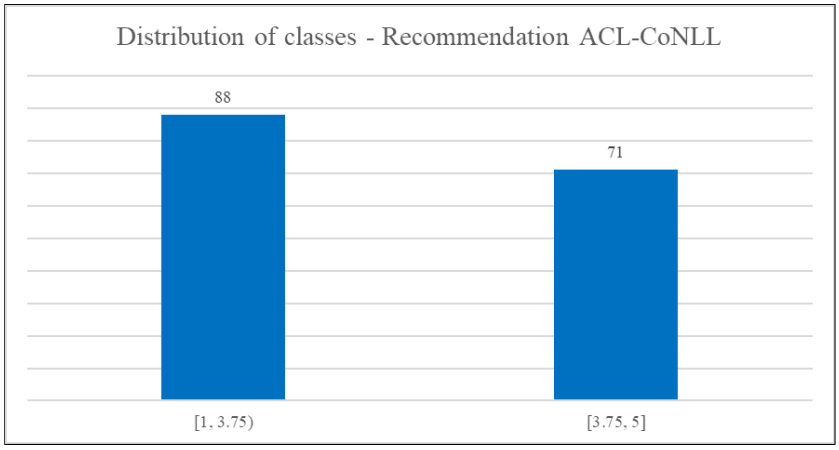


Figure 9.7: Final distribution of classes for *overall recommendation* scores in ACL-CoNLL.

9.3.1.2 Experiments with the ICLR dataset

Fig. 9.8 shows the distribution of recommendation scores for ICLR, weighted-averaged according to the reviewers' confidence scores.

It can be seen that, in this case, we obtain a more normal distribution of scores when compared to ACL-CoNLL. We can split the scores at the considered value closest to the median (which is 5.67) obtaining two fairly balanced classes with 52 and 48% of instances in each case, as show in Fig. 9.9

Having two classes for the evaluation of the *overall recommendation* scores for both PeerRead subsets allows us to compare results obtained in dataset. In addition, the classes can be considered as an approximation of what the reviewers' consensus recommendation would be in terms of acceptance/rejection of the manuscript,[12] although, as mentioned, this does not necessarily indicate whether that the manuscript would actually be accepted or not.

[12]This is not available in the dataset. Only final acceptance/rejection is indicated.

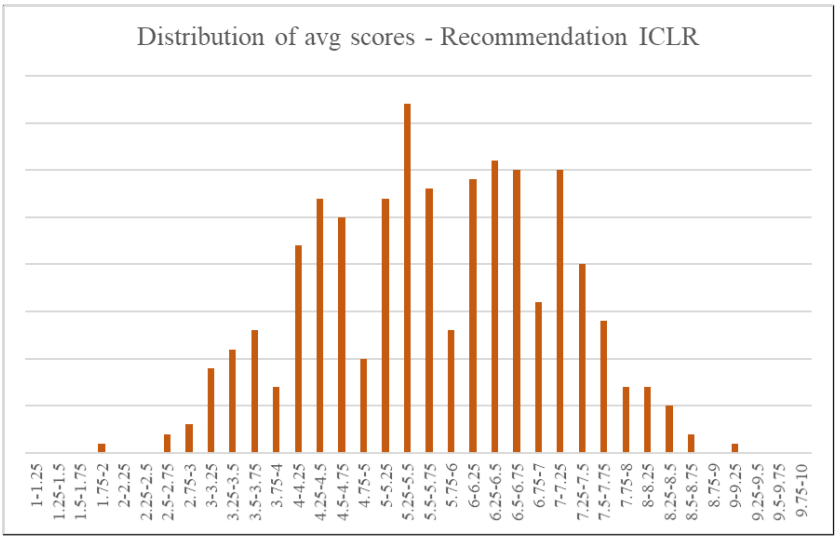


Figure 9.8: Distribution of *overall recommendation* average scores weighted by reviewers' confidence in ICLR.

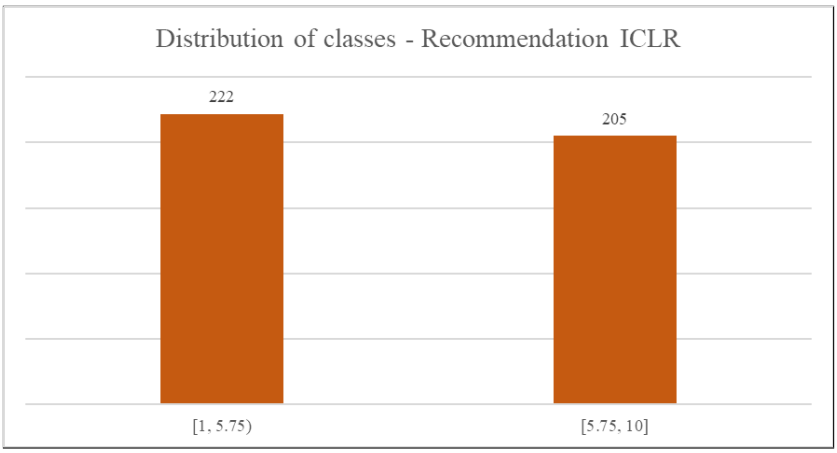


Figure 9.9: Final distribution of classes for *overall recommendation* scores in ICLR.

9.3.2 Results and analysis

Tables 9.5 and 9.6 show the results obtained for the prediction of the *overall recommendation* score for ACL-CoNLL and ICLR, respectively. In both cases, *random forest* is the best performing algorithm and, for both, the best configurations are obtained with features than can be considered to provide information about the *cogency* of the texts, except for the case of the second-top performing classifier for ICLR, where features such as the number of *proposal-implementation* units

and the number of *elaboration* and *sequence* relations, more clearly aligned with the *effectiveness* dimension, are the most predictive ones one when used by a *SVM* classifier.

Overall recommendation scores for ACL-CoNLL		
Algorithm	F_1	**Features**
Majority	0.3551	–
$Rule_L$	0.4609	*length*
RF	**0.6944**	*child-main, child-prop, motiv-prob, supp, coarse-motiv*
NN	0.6696	*child-main, child-prop, motiv-prob, supp, coarse-motiv*
DL	0.6398	*child-main, child-prop, motiv-prob, supp*

Table 9.5: Results for *overall recommendation scores* five-fold CV classification in ACL-CoNLL. Top-performing classifiers: *Random Forest*, *Nearest neighbours*, *Decision list*.

Overall recommendation scores for ICLR		
Algorithm	F_1	**Features**
Majority	0.3421	–
$Rule_L$	0.4692	*length*
RF	**0.5631**	*child-prop, motiv-prob*
SVM	0.5524	*child-main, prop-impl, elab, seq*
NB_K	0.5466	*obs, result, supp*

Table 9.6: Results for *overall recommendation scores* five-fold CV classification in ICLR. Top-performing classifiers: *Random Forest*, *Nearest neighbours*, *Decision list*.

We can observe that there is a considerable difference in performance of classifiers in both datasets. We cannot discard a hypothesis formulated by the authors of the PeerRead dataset with respect to a potential bias with respect to a higher degree of confidence of the ACL scores: *"We note, however, that ACL 2017 reviews were explicitly opted-in while the ICLR 2017 reviews include all official reviews, which is likely to result in a positive bias in review quality of the ACL reviews included in this study"*. (Kang et al., 2018)

In addition, we can observe, when comparing the distribution of scores in both datasets (in Figs. 9.7 and 9.9, respectively) that in ICLR the scores are much more concentrated around the median and, therefore, the two-class division is less straightforward than in the case of ACL-CoNLL. There are many more *ambiguous* instances which are, naturally, more difficult to classify–in particular, considering the size of the dataset and the algorithms used in these experiments. In order to explore to what extent this effect could impact on the performance of the classifiers, we conduct another experiment in which we split the ICLR dataset into two classes but excluding manuscripts with weighted-average scores too close to the median (less than 0.5 points in the 10-points scale). We thus obtain the distribution of manuscripts in two classes shown in Fig. 9.10.

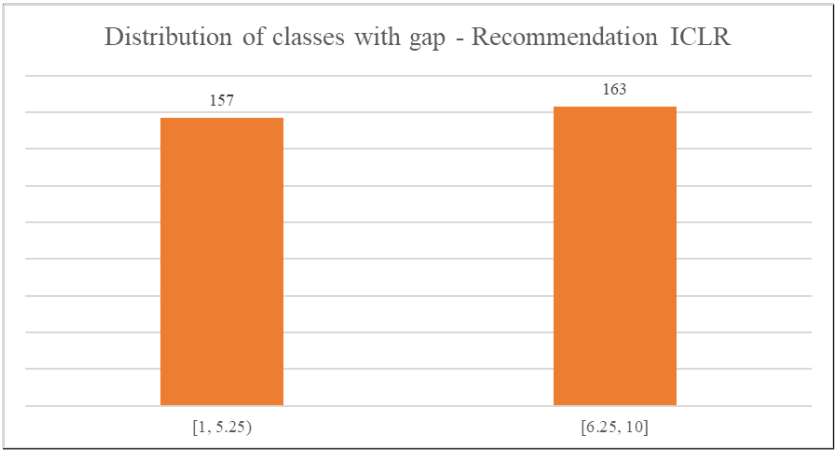


Figure 9.10: Final distribution of classes for *overall recommendation* scores in ICLR with gap between classes.

Table 9.7 shows the results obtained for the predicted classes in this case. Even considering that with the *difficult* training instances we are probably throwing away *useful* ones, we can see that with the same classifier and features *(random forest* with number of children of *proposal* units and number of *motivation-problem* units) we obtain a 5% gain-0,03 macro-averaged F_1 points.

Overall recommendation scores for ICLR w/gap between classes		
Algorithm	**Macro F1**	**Features**
Majority	0.3375	–
$Rule_L$	0.4656	*length*
RF	**0.5927**	*child-prop, motiv-prob*
NN	0.5948	*child-prop, motiv-prob*
SVM	0.5489	*obs, result, supp, prop-impl*

Table 9.7: Results for *overall recommendation scores* five-fold CV classification in ICLR with gap between classes. Top-performing classifiers: *Random Forest*, *Nearest neighbours*, *Support Vector Machine*.

9.4 Conclusions

In this section we explored both the prediction of *overall recommendation* scores as well as of scores that inform about two argumentative quality dimensions of the texts: their *cogency* and their *effectiveness*–and, more specifically, sub-dimensions including their *clarity*, *arrangement*, and *local sufficiency*.

The experiments described and their results indicate that a small set of features derived from argumentative units and relations automatically extracted from abstracts do convey information that can contribute to predict scores assigned to the whole manuscripts in a peer review process. Furthermore, the features involved in the best performing models can in fact be interpreted as providing relevant information for the two considered aspects (*clarity* and *soundness*).

In the prediction of the *overall recommendation* scores we observe differences in the performance of the classifiers depending on the venue and the method by which the reviews were obtained–opted-in vs. available online–which might indicate differences in the quality of the reviews themselves, as suggested by (Kang et al., 2018). Further experiments are be necessary to confirm whether the certainty of the predictions obtained can, in some way, provide information about the reliability of the reviews.

These experiments are intended as a downstream application of our argument mining models and, therefore, address very broad questions in relation to the applicability of the predictions obtained by our models–in particular, for the computational assessment of argumentative quality of scientific abstracts. The results obtained–in particular, when analyzing the scores for the *clarity* and *soundness* aspects–show that, even significant gains can be obtained when using these features when compared to very simple baselines.

It is evident, nevertheless, that these models alone cannot be used to obtain reliable predictions about the argumentative quality of the manuscripts and that our results provide only a starting point for further research in this direction.

Chapter 10

CONCLUSIONS AND FUTURE WORK

In this thesis we addressed the identification of argumentative components and relations in scientific abstracts. We proposed a new annotation scheme intended to bridge the gap between fine-grained discourse-level analysis, high-level classification of rhetorical components, and the identification of argumentative structures in scientific texts. In addition, we aimed to contribute to the advancement of argument mining research in a domain that, according to several authors, has not received sufficient attention (Al Khatib et al., 2021).

Even when we took as reference works that rely on theoretical approaches to argumentation and discourse analysis–both in general and in scientific texts–we adopted a pragmatic perspective in our work. In Chapter 3 we described our proposed annotation scheme, SciARG, and explained several of the decisions that guided its development, including the repertoire of types of units and relations considered, the choice to adopt sentences as main annotation units, and the possibility of annotating sentences with more than one type. We applied the proposed scheme to generate the SciARG-CL corpus, which enriches a subset of the abstracts included in the SciDTB corpus (Yang and Li, 2018) with an argumentation annotation layer. The decision of considering the SciDTB corpus as the basis for our annotations was made expressly with the purpose of analyzing potential benefits of leveraging discourse-level annotations in the identification of argumentative units and relations.

By making the SciARG corpus available, we expect it to serve as a basis for future research on how the various discourse and argumentative levels of analyses interplay.

We proposed to model the identification of argumentative structures in SciARG annotations by means of four related tasks (*unit type*, *parent attachment*, *relation type*, *main unit*). These tasks, as well as the experimental setups considered to train and evaluated them, were described in Chapter 4. We investigated the potential benefits obtained by leveraging discourse-level SciDTB annotations by means of a sequential transfer learning approach consisting in using SciDTB annotations to pre-fine-tune SciBERT (Beltagy et al., 2019), the base BERT model used in our experiments, before fine-tuning it with our target tasks. We observed that, in fact, this procedure contributed to improve the performance of our argument mining models. Moreover, the observed gains were inversely proportional to the size of the target training data, confirming the adequacy of exploiting existing discourse-level annotations for argument mining tasks. Intermediate fine-tuning is as a simple transfer learning method (Phang et al., 2018) which has mainly been applied to low-level tasks. Our results contribute to validate its effectiveness also in high-level, difficult semantic tasks. We also took into consideration the fact that the four SciARG tasks are closely linked to each other and therefore proposed to train them in a multi-task setting, observing that, in general, combining the training signals of the four tasks contributed to improve the models performances with respect to the models obtained when training the tasks independently. We adopted, as multi-task loss function, the sum of all the tasks' losses weighted according to the task difficulty, which was learned by the network as a trainable parameter. In order to deal with the fact that automatically identifying the argumentative structure of the abstracts by means of four different tasks could lead to predicted graphs that are not necessarily trees, we proposed and evaluated a set of heuristics for changing edges that violate the structure well-formedness–making sure, in addition, that in each abstract one and only one sentence is identified as *main unit*.

One of the stated objectives of this thesis (Section 1.1) was to test our hypothesis that, in the frequent cases in which the proportion of sentences containing two or more argumentative units is significantly lower than that of sentences containing only one unit, it is not the best option to use indiscriminately a method designed to deal with the most difficult cases. We proposed, instead, to first distinguish between these two types of sentences and then apply the method that best suits each

case: sequence-level classification for sentences with a single unit and token-level classification for sentences with more than one unit, for which we must also predict their boundaries. In Chapter 5 we addressed this issue. As the proposed method requires the implementation of pipelines that rely on the possibility of classifying the rhetorical/argumentative complexity of sentences, we explored different approaches for this task, including: i) a basic sentence-level classifier that predicts the complexity directly, ii) predicting the–potentially multiple–types of units contained in a sentence, and iii) a combination of both. The results obtained confirm that discriminating sentences according to their complexity and applying specific methods in each case can contribute to improve the overall predictions–as well as the predictions of each subset. In this chapter we took advantage of the availability of the MAZEA corpus of scientific abstracts (Dayrell et al., 2012), which we used to conduct several of our experiments–both for the identification of multiple types of rhetorical moves within scientific abstracts and for the classification of sentences according to their rhetorical complexity. In addition, we continued exploring the potential benefits obtained by leveraging existing annotations by means of an intermediate fine-tuning approach, considering, in this case, the prediction of rhetorical-level annotations in MAZEA as supplementary task, thus confirming the utility of this approach in yet another context.

In Chapter 6 we evaluated the application of the SciARG annotation scheme to biomedical abstracts. We observed that, in spite of the fact that the it was originally developed and refined for computational linguistics, the scheme accounts for the types of units and relations likely to be found in biomedical abstracts. Somewhat lower levels of agreement between annotations in this scientific discipline were attributed to the greater complexity of the biomedical discourse, as evidenced when analyzing the number and length of the components identified and the way in which they are linked to each other. In particular, disagreement was observed in the way in which the reported outcomes were analyzed in different annotations. After studying the sources of discrepancies, we observed that some these ambiguities could be partly solved by updating the annotation guidelines. For more precise results, although, we considered that it would be required to include domain expert annotators in the process. How to best incorporate this knowledge (for instance, by refining non-expert annotations, by fully annotating new texts, or by identifying only some types of units and relations–the ones for which domain knowledge is most needed), and to what extent this process can be partially automated, opens up new and interesting research questions.

The experiments conducted with the biomedical annotated abstracts, SciARG-BIO, confirmed some of the results that we had previously observed with SciARG-CL, in Chapter 4, including the benefits derived from training the tasks jointly in a multi-task setting. More relevantly, the experiments showed that BERT encoders fine-tuned with annotations in one scientific discipline capture knowledge about the argumentative structure of the abstracts that is useful to predict argumentative units and relations in abstracts from a different discipline–even in research fields where different argumentative complexity levels could be observed.

In the second part of this thesis we explored the practical usefulness of the proposed annotations and the models trained with them. In particular, we addressed the prediction of argumentative quality scores, both assigned directly to the abstracts, or attached to full papers in a peer-review process. We observed, in both cases, that a small set of features, which could be assigned semantics aligned with argumentative quality dimensions, could in fact contribute to predict scores intended to reflect those dimensions. These results can motivate many potential follow-ups. The models described in this chapter–both in terms of the features used and the algorithms implemented–were eminently exploratory. Should the assessment of argumentative quality be the main research focus, additional experiments would be necessary to identify more precisely the types of features that best convey different argumentative dimensions, as well as the best algorithms to use with them. Optimization of these algorithms' parameters would also be needed. Other works, including (Kang et al., 2018), use features extracted from the manuscripts' abstracts, among others, to predict scores assigned to papers in a peer-review process. In our experiments, we used review scores assigned to manuscripts as a proxy for argumentative quality scores of abstracts. While it is natural to expect some degree of alignment between them, they are clearly two different things. Additional research would therefore be needed to confirm their correlation, and to explore in more depth in which contexts and for which tasks information extracted from the abstracts can be considered as representative of information expected to be obtained from the full papers.

In the two chapters included in the second part of the thesis we considered the reliability of the quality scores assigned to the texts. In the case of the peer reviews, in Chapter 9, a confidence score is provided by the reviewers explicitly. In this case we had multiple scores for the same instance and used the reviewers' confidence scores to compute a weighted average. In the case of the scores assigned

to SciARG-CL abstracts, in Chapter 8, we were in a different situation: for most of the instances we had only one score but, in the set of instances annotated in common, we could observe significant differences in the criteria followed to assign them. Moreover, we observed differences in the levels of reliability between annotators. Proposals have been made to select annotators and/or instances when a large number of instances is available (for instance, in the context of crowdsourcing annotation initiatives). This was not our case. We therefore proposed to reflect the observed differences in reliability in the training process, so the final models could weight differently instances produced by different annotators. We observed that, either including information about the annotators explicitly in the training process, or weighting the instances according the the annotator's reliability, led to improvements in the predictions of the most reliable scores. This also opens up some relevant research questions. In particular, on how to best reflect the different levels of reliability of different annotators in the process of training models with a limited number of instances for highly-subjective tasks.

Even when our annotation scheme was developed specifically for abstracts, we believe it to have a level of generality that could make it applicable to other parts of scientific publications, such as papers' introduction and/or conclusion sections. Further research to confirm this hypothesis would be a natural continuation to the work described in this thesis. To extend the annotation of argumentative units and relations to other sections of the papers, and to explore potential mappings between the argumentative structure of the abstract and the argumentative structure of other sections, could also lead to interesting follow-up research, such as the generation of argumentative summaries based on predictions obtained by models trained with our proposed annotations.

Bibliography

Abdollahpour, Z. and Gholami, J. (2018). Rhetorical structure of the abstracts of medical sciences research articles. *La Prensa Medica Argentina*, 105(2):1–5.

Accuosto, P., Neves, M., and Saggion, H. (2021). Argumentation mining in scientific literature: From computational linguistics to biomedicine. In Frommholz, I., Mayr, P., Cabanac, G., and Verberne, S., editors, *Proceedings of the 11th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 43rd European Conference on Information Retrieval (ECIR 2021), Lucca, Italy (online only), April 1st, 2021*, volume 2847 of *CEUR Workshop Proceedings*, pages 20–36. CEUR-WS.org.

Accuosto, P. and Saggion, H. (2019a). Discourse-driven argument mining in scientific abstracts. In Métais, E., Meziane, F., Vadera, S., Sugumaran, V., and Saraee, M., editors, *Natural Language Processing and Information Systems - 24th International Conference on Applications of Natural Language to Information Systems, NLDB 2019, Salford, UK, June 26-28, 2019, Proceedings*, volume 11608 of *Lecture Notes in Computer Science*, pages 182–194. Springer.

Accuosto, P. and Saggion, H. (2019b). Transferring knowledge from discourse to arguments: A case study with scientific abstracts. In *Proceedings of the 6th Workshop on Argument Mining*, pages 41–51, Florence, Italy. Association for Computational Linguistics.

Accuosto, P. and Saggion, H. (2020). Mining arguments in scientific abstracts with discourse-level embeddings. *Data & Knowledge Engineering*, 129:101840.

Afantenos, S., Peldszus, A., and Stede, M. (2018). Comparing decoding mechanisms for parsing argumentative structures. *Argument & Computation*, 9(3):177–192.

Aharoni, E., Polnarov, A., Lavee, T., Hershcovich, D., Levy, R., Rinott, R., Gutfreund, D., and Slonim, N. (2014). A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68, Baltimore, Maryland. Association for Computational Linguistics.

Ahmed, S., Blake, C., Williams, K., Lenstra, N., and Liu, Q. (2013). Identifying claims in social science literature.

Ajjour, Y., Wachsmuth, H., Kiesel, J., Potthast, M., Hagen, M., and Stein, B. (2019). Data acquisition for argument search: The args.me corpus. In Benzmüller, C. and Stuckenschmidt, H., editors, *KI 2019: Advances in Artificial Intelligence*, pages 48–59, Cham. Springer International Publishing.

Al Khatib, K., Ghosal, T., Hou, Y., de Waard, A., and Freitag, D. (2021). Argument mining for scholarly document processing: Taking stock and looking ahead. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 56–65, Online. Association for Computational Linguistics.

Al-Khatib, K., Wachsmuth, H., Hagen, M., Köhler, J., and Stein, B. (2016). Cross-domain mining of argumentative text through distant supervision. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1395–1404, San Diego, California. Association for Computational Linguistics.

Alamri, A. (2016). *The detection of contradictory claims in biomedical abstracts*. PhD thesis, University of Sheffield.

Alshomary, M., Düsterhus, N., and Wachsmuth, H. (2020). Extractive snippet generation for arguments. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1969–1972.

Anand, P., Walker, M., Abbott, R., Fox Tree, J. E., Bowmani, R., and Minor, M. (2011). Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9, Portland, Oregon. Association for Computational Linguistics.

Asher, N., Asher, N. M., and Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

Azar, M. (1999). Argumentative text as rhetorical structure: An application of rhetorical structure theory. *Argumentation*, 13(1):97–114.

Azunre, P. (2021). *Transfer Learning for Natural Language Processing*. Manning Publications.

Bar-Haim, R., Eden, L., Friedman, R., Kantor, Y., Lahav, D., and Slonim, N. (2020). From arguments to key points: Towards automatic argument summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039, Online. Association for Computational Linguistics.

Baxter, J. (1997). A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39.

Beltagy, I., Lo, K., and Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Bentahar, J., Moulin, B., and Bélanger, M. (2010). A taxonomy of argumentation models used for knowledge representation. *Artificial Intelligence Review*, 33(3):211–259.

Biran, O. and Rambow, O. (2011). Identifying justifications in written dialogs by classifying text as argumentative. *International Journal of Semantic Computing*, 5(04):363–381.

Blake, C. (2010). Beyond genes, proteins, and abstracts: Identifying scientific claims from full-text biomedical articles. *Journal of biomedical informatics*, 43(2):173–189.

Boltužić, F. and Šnajder, J. (2014). Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland. Association for Computational Linguistics.

Cabrio, E., Tonelli, S., and Villata, S. (2013). From discourse analysis to argumentation schemes and back: Relations and differences. In Leite, J., Son, T. C., Torroni, P., van der Torre, L., and Woltran, S., editors, *Computational Logic in*

Multi-Agent Systems, pages 1–17, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cabrio, E. and Villata, S. (2012a). Combining textual entailment and argumentation theory for supporting online debates interactions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 208–212, Jeju Island, Korea. Association for Computational Linguistics.

Cabrio, E. and Villata, S. (2012b). Natural language arguments: A combined approach. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 205–210.

Cabrio, E. and Villata, S. (2013). A natural language bipolar argumentation approach to support users in online debate interactions. *Argument & Computation*, 4(3):209–230.

Cabrio, E. and Villata, S. (2014). NoDE: A benchmark of natural language arguments. In *Computational Models of Argument*, pages 449–450. IOS Press.

Cabrio, E. and Villata, S. (2018). Five years of argument mining: A data-driven analysis. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5427–5433.

Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

Cross, C. and Oppenheim, C. (2006). A genre analysis of scientific abstracts. *Journal of documentation*.

Dayrell, C., Candido Jr., A., Lima, G., Machado Jr., D., Copestake, A., Feltrim, V., Tagnin, S., and Aluisio, S. (2012). Rhetorical move detection in English abstracts: Multi-label sentence classifiers and their annotated corpora. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1604–1609, Istanbul, Turkey. European Language Resources Association (ELRA).

Dernoncourt, F. and Lee, J. Y. (2017). PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ding, J., Tarokh, V., and Yang, Y. (2018). Model selection techniques: An overview. *IEEE Signal Processing Magazine*, 35(6):16–34.

Doró, K. (2013). The rhetoric structure of research article abstracts in English studies journals. *Prague Journal of English Studies*, 2(1):119–139.

Dos Santos, M. B. (1996). The textual organization of research paper abstracts in applied linguistics. *Text-Interdisciplinary Journal for the Study of Discourse*, 16(4):481–500.

Dumani, L., Biertz, M., Witry, A., Ludwig, A.-K., Lenz, M., Ollinger, S., Bergmann, R., and Schenkel, R. (2021). The recap corpus: A corpus of complex argument graphs on german education politics. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 248–255.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357.

Dusmanu, M., Cabrio, E., and Villata, S. (2017). Argument mining on Twitter: Arguments, facts and sources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2317–2322, Copenhagen, Denmark. Association for Computational Linguistics.

Duthie, R., Budzynska, K., and Reed, C. (2016). Mining ethos in political debate. In *Computational Models of Argument: Proceedings from the Sixth International Conference on Computational Models of Argument (COMMA)*, pages 299–310. IOS Press.

Eckle-Kohler, J., Kluge, R., and Gurevych, I. (2015). On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Lisbon, Portugal. Association for Computational Linguistics.

Egan, C., Siddharthan, A., and Wyner, A. (2016). Summarising the points made in online political debates. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 134–143, Berlin, Germany. Association for Computational Linguistics.

Eger, S., Daxenberger, J., and Gurevych, I. (2017). Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada. Association for Computational Linguistics.

El-Assady, M., Hautli-Janisz, A., Gold, V., Butt, M., Holzinger, K., and Keim, D. (2017). Interactive visual analysis of transcribed multi-party discourse. In *Proceedings of ACL 2017, System Demonstrations*, pages 49–54, Vancouver, Canada. Association for Computational Linguistics.

El Baff, R., Wachsmuth, H., Al-Khatib, K., and Stein, B. (2018). Challenge or empower: Revisiting argumentation quality in a news editorial corpus. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 454–464, Brussels, Belgium. Association for Computational Linguistics.

Feltrim, V. D., Aluísio, S. M., and Nunes, M. d. G. V. (2003). Analysis of the rhetorical structure of computer science abstracts in Portuguese. In *Proceedings of Corpus Linguistics*, volume 16, pages 212–218.

Feltrim, V. D., Teufel, S., das Nunes, M. G. V., and Aluísio, S. M. (2006). *Argumentative Zoning Applied to Critiquing Novices' Scientific Abstracts*, pages 233–246. Springer Netherlands, Dordrecht.

Fisas, B., Ronzano, F., and Saggion, H. (2016). A multi-layered annotated corpus of scientific papers. In *Proceedings of the 2016 The International Conference on Language Resources and Evaluation*.

Florou, E., Konstantopoulos, S., Koukourikos, A., and Karampiperis, P. (2013). Argument extraction for supporting public policy formulation. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 49–54, Sofia, Bulgaria. Association for Computational Linguistics.

Frank, E. and Witten, I. H. (1998). Generating accurate rule sets without global optimization. In Shavlik, J., editor, *Fifteenth International Conference on Machine Learning*, pages 144–151. Morgan Kaufmann.

Freeman, J. B. (2011). *Dialectics and the Macrostructure of Arguments*. De Gruyter Mouton.

Galassi, A., Lippi, M., and Torroni, P. (2020). Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*.

García-Villalba, M. P. and Saint-Dizier, P. (2012). A framework to extract arguments in opinion texts. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 6(3):62–87.

Ghosal, T., Verma, R., Ekbal, A., and Bhattacharyya, P. (2019). DeepSentiPeer: Harnessing sentiment in review texts to recommend peer review decisions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1120–1130, Florence, Italy. Association for Computational Linguistics.

Ghosh, D., Khanam, A., Han, Y., and Muresan, S. (2016). Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 549–554, Berlin, Germany. Association for Computational Linguistics.

Gleize, M., Shnarch, E., Choshen, L., Dankin, L., Moshkowich, G., Aharonov, R., and Slonim, N. (2019). Are you convinced? choosing the more convincing evidence with a Siamese network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 967–976, Florence, Italy. Association for Computational Linguistics.

Gong, T., Lee, T., Stephenson, C., Renduchintala, V., Padhy, S., Ndirango, A., Keskin, G., and Elibol, O. H. (2019). A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632.

Goudas, T., Louizos, C., Petasis, G., and Karkaletsis, V. (2014). Argument extraction from news, blogs, and social media. In *Hellenic Conference on Artificial Intelligence*, pages 287–299. Springer.

Granger, S., Dagneaux, E., Meunier, F., Paquot, M., et al. (2009). *International Corpus of Learner English*. Presses universitaires de Louvain Louvain-la-Neuve.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Green, N. (2015). Identifying argumentation schemes in genetics research articles. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 12–21, Denver, CO. Association for Computational Linguistics.

Green, N. (2016). Implementing argumentation schemes as logic programs. In Bex, F., Grasso, F., and Green, N., editors, *Proceedings of the 16th Workshop on Computational Models of Natural Argument co-located with IJCAI 2016, New York, USA, July 9th, 2016*, volume 1876 of *CEUR Workshop Proceedings*, pages 1–7. CEUR-WS.org.

Green, N. L. (2018). Towards mining scientific discourse using argumentation schemes. *Argument & Computation*, 9:121–135.

Guo, Z., Schlichtkrull, M., and Vlachos, A. (2021). A survey on automated fact-checking. *arXiv preprint arXiv:2108.11896*.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Habernal, I. and Gurevych, I. (2016a). What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223, Austin, Texas. Association for Computational Linguistics.

Habernal, I. and Gurevych, I. (2016b). Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599, Berlin, Germany. Association for Computational Linguistics.

Habernal, I. and Gurevych, I. (2017). Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Hanselowski, A., Stab, C., Schulz, C., Li, Z., and Gurevych, I. (2019). A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 493–503.

Hao, Y., Dong, L., Wei, F., and Xu, K. (2019). Visualizing and understanding the effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4143–4152, Hong Kong, China. Association for Computational Linguistics.

Hartley, J. and Betts, L. (2009). Common weaknesses in traditional abstracts in the social sciences. *Journal of the American Society for Information Science and Technology*, 60(10):2010–2018.

Hewett, F., Rane, R. P., Harlacher, N., and Stede, M. (2019). The utility of discourse parsing features for predicting argumentation structure. In *Proceedings of the 6th Workshop on Argument Mining*, pages 98–103.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., and Hovy, E. (2013). Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

Hubert, L. (1977). Kappa revisited. *Psychological Bulletin*, 84(2):289.

Hyland, K. (1998). *Hedging in scientific research articles*, volume 54. John Benjamins Publishing.

Ibeke, E., Lin, C., Wyner, A., and Barawi, M. H. (2017). Extracting and understanding contrastive opinion through topic relevant sentences. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*

(Volume 2: Short Papers), pages 395–400, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Johnson, R., Watkinson, A., and Mabe, M. (2018). The STM report: An overview of scientific and scholarly publishing. *International Association of Scientific, Technical and Medical Publishers*.

Johnson, R. H. and Blair, J. A. (2006). *Logical self-defense*. The International Debate Education Association (IDEA).

Kailas, P. (2021). *Argument Mining for Understanding Media Bias and Misinformation*. PhD thesis, Northeastern University.

Kang, D., Ammar, W., Dalvi, B., van Zuylen, M., Kohlmeier, S., Hovy, E., and Schwartz, R. (2018). A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Ke, Z. and Ng, V. (2019). Automated essay scoring: A survey of the state of the art. In *IJCAI*, volume 19, pages 6300–6308.

Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Kiesel, J., Spina, D., Wachsmuth, H., and Stein, B. (2021). The Meant, the Said, and the Understood: Conversational argument search and cognitive biases. In *CUI 2021-3rd Conference on Conversational User Interfaces*, pages 1–5.

Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., and Rindflesch, T. C. (2012). SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics*, 28(23):3158–3160.

Kim, S. N., Martinez, D., Cavedon, L., and Yencken, L. (2011). Automatic classification of sentences to support evidence based medicine. In *BMC bioinformatics*, volume 12, pages 1–10. BioMed Central.

Kirschner, C., Eckle-Kohler, J., and Gurevych, I. (2015a). Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 1–11.

Kirschner, C., Eckle-Kohler, J., and Gurevych, I. (2015b). Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 1–11, Denver, CO. Association for Computational Linguistics.

Kolhatkar, V. and Hirst, G. (2014). Resolving shell nouns. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 499–510, Doha, Qatar. Association for Computational Linguistics.

Komninos, A. and Manandhar, S. (2016). Dependency-based embeddings for sentence classification tasks. In *Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies (NAACL 2016)*, pages 1490–1500.

Kreer, J. (1957). A question of terminology. *IRE Transactions on Information Theory*, 3(3):208–208.

Krippendorff, K. (2007). Computing Krippendorffś Alpha-Reliability (Annenberg School for Communication Working Paper 43).

Kwon, N., Zhou, L., Hovy, E., and Shulman, S. W. (2007). Identifying and classifying subjective claims. In *Proceedings of the 8th Annual International Conference on Digital Government Research: Bridging disciplines & domains*, pages 76–81.

Landwehr, N., Arzt, S., Scheffer, T., and Kliegl, R. (2014). A model of individual differences in gaze control during reading. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1810–1815, Doha, Qatar. Association for Computational Linguistics.

Lauscher, A., Glavaš, G., and Eckert, K. (2018a). ArguminSci: A tool for analyzing argumentation and rhetorical aspects in scientific writing. In *Proceedings of the 5th Workshop on Argument Mining (ArgMining 2018)*, pages 22–28.

Lauscher, A., Glavaš, G., and Ponzetto, S. P. (2018b). An argument-annotated corpus of scientific publications. In *Proceedings of the 5th Workshop on Argument Mining (ArgMining 2018)*, pages 40–46.

Lawrence, J. and Reed, C. (2014). Aifdb corpora. In *Fifth International Conference on Computational Models of Argument*, pages 465–466. IOS Press.

Lawrence, J. and Reed, C. (2020). Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818.

Lawrence, J., Visser, J., and Reed, C. (2018). BBC moral maze: Test your argument. In Modgil, S., Budzynska, K., Lawrence, J., and Budzynska, K., editors, *Computational Models of Argument - Proceedings of COMMA 2018*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 465–466, Netherlands. IOS Press. 7th International Conference on Computational Models of Argument, COMMA 2018 ; Conference date: 12-09-2018 Through 14-09-2018.

Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E., and Slonim, N. (2014). Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Li, S., Wang, L., Cao, Z., and Li, W. (2014). Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35, Baltimore, Maryland. Association for Computational Linguistics.

Li, Y. and Fung, P. (2014). Language modeling with functional head constraint for code switching speech recognition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 907–916, Doha, Qatar. Association for Computational Linguistics.

Liakata, M., Saha, S., Dobnik, S., Batchelor, C., and Rebholz-Schuhmann, D. (2012). Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000.

Liakata, M., Teufel, S., Siddharthan, A., and Batchelor, C. (2010). Corpora for the conceptualisation and zoning of scientific papers. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Liebeck, M., Esau, K., and Conrad, S. (2016). What to do with an airport? mining arguments in the German online participation project tempelhofer feld. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 144–153, Berlin, Germany. Association for Computational Linguistics.

Lippi, M. and Torroni, P. (2016a). Argument mining from speech: Detecting claims in political debates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Lippi, M. and Torroni, P. (2016b). Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, 16(2):10:1–10:25.

Lippi, M. and Torroni, P. (2016c). MARGOT: A web server for argumentation mining. *Expert Systems with Applications*, 65:292–303.

Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lu, X., Casal, J. E., and Liu, Y. (2020). The rhetorical functions of syntactically complex sentences in social science research article introductions. *Journal of English for Academic Purposes*, 44:100832.

Lugini, L. and Litman, D. (2020). Contextual argument component classification for class discussions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1475–1480, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lytos, A., Lagkas, T., Sarigiannidis, P., and Bontcheva, K. (2019). The evolution of argumentation mining: From models to social media and emerging tools. *Information Processing & Management*, 56(6):102055.

Macagno, F., Walton, D., and Reed, C. (2017). Argumentation schemes. history, classifications, and computational applications. *History, Classifications, and Computational Applications (December 23, 2017). Macagno, F., Walton, D. & Reed, C*, pages 2493–2556.

Maeda, T. (1981). An approach toward functional text structure analysis of scientific and technical documents. *Information Processing & Management*, 17(6):329–339.

Mann, W. C., Matthiessen, C., and Thompson, S. A. (1992). Rhetorical Structure Theory and text analysis. *Discourse Description: Diverse linguistic analyses of a fund-raising text*, 16:39–78.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization*. MIT press.

Marcu, D., Amorrortu, E., and Romera, M. (1999). Experiments in constructing a corpus of discourse trees. In *Towards Standards and Tools for Discourse Tagging*.

Menini, S., Cabrio, E., Tonelli, S., and Villata, S. (2018). Never retreat, never retract: Argumentation analysis for political speeches. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Mihăilă, C., Ohta, T., Pyysalo, S., and Ananiadou, S. (2013). Biocause: Annotating and analysing causality in the biomedical domain. *BMC bioinformatics*, 14(1):1–18.

Mochales-Palau, R. and Moens, M.-F. (2011). Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Morey, M., Muller, P., and Asher, N. (2017). How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 1319–1324, Copenhagen, Denmark. Association for Computational Linguistics.

Mosbach, M., Andriushchenko, M., and Klakow, D. (2021). On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations*, number CONF.

Musi, E., Stede, M., Kriese, L., Muresan, S., and Rocci, A. (2018). A multi-layer annotated corpus of argumentative text: From argument schemes to discourse relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Naderi, N. and Hirst, G. (2015). Argumentation mining in parliamentary discourse. In *Principles and practice of multi-agent systems*, pages 16–25. Springer.

Neves, M., Butzke, D., and Grune, B. (2019). Evaluation of scientific elements for text similarity in biomedical publications. In *Proceedings of the 6th Workshop on Argument Mining*, pages 124–135, Florence, Italy. Association for Computational Linguistics.

Ng, L., Lauscher, A., Tetreault, J., and Napoles, C. (2020). Creating a domain-diverse corpus for theory-based argument quality assessment. In *Proceedings of the 7th Workshop on Argument Mining*, pages 117–126, Online. Association for Computational Linguistics.

Nguyen, H. V. and Litman, D. J. (2018). Argument mining for improving the automated scoring of persuasive essays. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Niculae, V., Park, J., and Cardie, C. (2017). Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995, Vancouver, Canada. Association for Computational Linguistics.

Olshefski, C., Lugini, L., Singh, R., Litman, D., and Godley, A. (2020). The discussion tracker corpus of collaborative argumentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1033–1043, Marseille, France. European Language Resources Association.

Ong, N., Litman, D., and Brusilovsky, A. (2014). Ontology-based argument mining and automatic essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 24–28, Baltimore, Maryland. Association for Computational Linguistics.

Orasan, C. (2001). Patterns in scientific abstracts. In *Proceedings of Corpus Linguistics 2001 Conference*, pages 433–443. Citeseer.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Park, D. H. and Blake, C. (2012). Identifying comparative claim sentences in full-text scientific articles. In *Proceedings of the workshop on detecting structure in scholarly discourse*, pages 1–9.

Park, J. and Cardie, C. (2018). A corpus of eRulemaking user comments for measuring evaluability of arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Park, S. and Caragea, C. (2020). Scientific keyphrase identification and classification by pre-trained language models intermediate task transfer learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5409–5419, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Peldszus, A. and Stede, M. (2013). From argument diagrams to automatic argument mining: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.

Peldszus, A. and Stede, M. (2015a). An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon*, volume 2, pages 801–815.

Peldszus, A. and Stede, M. (2015b). Joint prediction in MST-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 938–948.

Peldszus, A. and Stede, M. (2016). Rhetorical structure and argumentation structure in monologue text. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 103–112, Berlin, Germany. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Persing, I. and Ng, V. (2014). Vote prediction on comments in social polls. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1127–1138, Doha, Qatar. Association for Computational Linguistics.

Persing, I. and Ng, V. (2015). Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552, Beijing, China. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Phang, J., Févry, T., and Bowman, S. R. (2018). Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *CoRR*, abs/1811.01088.

Pinto, J. M. G. and Balke, W. (2020). Assessing plausibility of scientific claims to support high-quality content in digital collections. *Int. J. Digit. Libr.*, 21(1):47–60.

Potash, P., Bhattacharya, R., and Rumshisky, A. (2017). Length, interchangeability, and external knowledge: Observations from predicting argument convincingness. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 342–351, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Poth, C., Pfeiffer, J., Rücklé, A., and Gurevych, I. (2021). What to pre-train on? efficient intermediate task selection. *arXiv preprint arXiv:2104.08247*.

Poudyal, P., Savelka, J., Ieven, A., Moens, M. F., Goncalves, T., and Quaresma, P. (2020). ECHR: Legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75, Online. Association for Computational Linguistics.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B. (2008). The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Prasad, R., McRoy, S., Frid, N., Joshi, A., and Yu, H. (2011). The biomedical discourse relation bank. *BMC bioinformatics*, 12(1):1–18.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. R. (2020). Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Pustejovsky, J. (1998). *The generative lexicon*. MIT press.

Qiao, F., Xu, L., and Han, X. (2018). Modularized and attention-based recurrent convolutional neural network for automatic academic paper aspect scoring. In *International Conference on Web Information Systems and Applications*, pages 68–76. Springer.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Radev, D. R., Muthukrishnan, P., and Qazvinian, V. (2009). The ACL Anthology network. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries (NLPIR4DL)*, pages 54–61, Suntec City, Singapore. Association for Computational Linguistics.

Rapp, C. (2010). Aristotle's Rhetoric. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2010 edition.

Ravenscroft, J., Oellrich, A., Saha, S., and Liakata, M. (2016). Multi-label annotation in scientific articles - the multi-label cancer risk assessment corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4115–4123, Portorož, Slovenia. European Language Resources Association (ELRA).

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009). Classifier chains for multi-label classification. In Buntine, W., Grobelnik, M., Mladenić, D.,

and Shawe-Taylor, J., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, Berlin, Heidelberg. Springer Berlin Heidelberg.

Reed, C., Budzynska, K., Lawrence, J., Pereira-Farina, M., De Franco, D., Duthie, R., Koszowy, M., Pease, A., Pluss, B., Snaith, M., et al. (2018). Large-scale deployment of argument analytics. In *In Argumentation and Society Workshop at the 7th International Conference on Computational Models of Argument (COMMA 2018)*.

Reed, C., Palau, R. M., Rowe, G., and Moens, M.-F. (2008). Language resources for studying argument. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Reed, C. and Rowe, G. (2004). Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(04):961–979.

Reimers, N. and Gurevych, I. (2017). Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Rigotti, E. and Morasso, S. G. (2010). Comparing the argumentum model of topics to other contemporary approaches to argument schemes: The procedural and material components. *Argumentation*, 24(4):489–512.

Rinott, R., Dankin, L., Alzate Perez, C., Khapra, M. M., Aharoni, E., and Slonim, N. (2015). Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal. Association for Computational Linguistics.

Rogers, E. M. (2010). *Diffusion of innovations*. Simon and Schuster.

Rosenthal, S. and McKeown, K. (2012). Detecting opinionated claims in online discussions. In *2012 IEEE sixth international conference on semantic computing*, pages 30–37. IEEE.

Ruder, S. (2019). *Neural transfer learning for natural language processing*. PhD thesis, NUI Galway.

Saint-Dizier, P. (2018). A two-level approach to generate synthetic argumentation reports. *Argument & Computation*, 9(2):137–154.

Sardianos, C., Katakis, I. M., Petasis, G., and Karkaletsis, V. (2015). Argument extraction from news. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 56–66, Denver, CO. Association for Computational Linguistics.

Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B. (2011). Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE.

Schaefer, R. and Stede, M. (2020). Annotation and detection of arguments in tweets. In *Proceedings of the 7th Workshop on Argument Mining*, pages 53–58, Online. Association for Computational Linguistics.

Schaefer, R. and Stede, M. (2021). Argument mining on twitter: A survey. *it - Information Technology*, 63(1):45–58.

Schuster, M. and Nakajima, K. (2012). Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Shannon, C. E. (1949). *The Mathematical Theory of Communication*. University of Illinois Press.

Siegel, S. and Castellan Jr, N. J. (1988). Nonparametric statistics for the behavioral sciences.

Simpson, E. and Gurevych, I. (2018). Finding convincing arguments using scalable Bayesian preference learning. *Transactions of the Association for Computational Linguistics*, 6:357–371.

Skorikov, M. and Momen, S. (2020). Machine learning approach to predicting the acceptance of academic papers. In *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 113–117.

Slonim, N., Bilu, Y., Alzate, C., Bar-Haim, R., Bogin, B., Bonin, F., Choshen, L., Cohen-Karlik, E., Dankin, L., Edelstein, L., et al. (2021). An autonomous debating system. *Nature*, 591(7850):379–384.

Soergel, D., Saunders, A., and McCallum, A. (2013). Open scholarship and peer review: a time for experimentation. In *ICML Workshop on Peer Reviewing and Publishing Models (PEER)*.

Somasundaran, S. and Wiebe, J. (2010). Recognizing stances in ideological online debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA. Association for Computational Linguistics.

Sonntag, J. and Stede, M. (2014). GraPAT: a tool for graph annotations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4147–4151, Reykjavik, Iceland. European Language Resources Association (ELRA).

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 228–235.

South, L., Schwab, M., Beauchamp, N., Wang, L., Wihbey, J., and Borkin, M. A. (2020). Debatevis: Visualizing political debates for non-expert users. In *2020 IEEE Visualization Conference (VIS)*, pages 241–245. IEEE.

Stab, C. (2017). *Argumentative Writing Support by means of Natural Language Processing*. PhD thesis.

Stab, C. and Gurevych, I. (2014a). Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Stab, C. and Gurevych, I. (2014b). Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Stab, C. and Gurevych, I. (2016). Recognizing the absence of opposing arguments in persuasive essays. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 113–118, Berlin, Germany. Association for Computational Linguistics.

Stab, C. and Gurevych, I. (2017a). Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Stab, C. and Gurevych, I. (2017b). Recognizing insufficiently supported arguments in argumentative essays. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 980–990, Valencia, Spain. Association for Computational Linguistics.

Stab, C., Kirschner, C., Eckle-Kohler, J., and Gurevych, I. (2014). Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. In *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21-25, 2014*, pages 21–25.

Stede, M., Afantenos, S., Peldszus, A., Asher, N., and Perret, J. (2016). Parallel discourse annotations on a corpus of short texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1051–1058, Portorož, Slovenia. European Language Resources Association (ELRA).

Stede, M. and Schneider, J. (2018). *Argumentation Mining*. Synthesis Lectures On Human Language Technologies. Morgan and Claypool.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Subramanian, S., Trischler, A., Bengio, Y., and Pal, C. J. (2018). Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*.

Swales, J. (1990). *Genre analysis: English in academic and research settings*. Cambridge University Press.

Syed, S., El Baff, R., Kiesel, J., Al Khatib, K., Stein, B., and Potthast, M. (2020). News editorials: Towards summarizing long argumentative texts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5384–5396, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer.

Teruel, M., Cardellino, C., Cardellino, F., Alonso Alemany, L., and Villata, S. (2018). Increasing argument annotation reproducibility by using inter-annotator agreement to improve guidelines. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Teufel, S. (2010). *The structure of scientific articles: Applications to citation indexing and summarization*. Center for the Study of Language and Information.

Teufel, S. et al. (1999). *Argumentative zoning: Information extraction from scientific text*. PhD thesis, University of Edinburgh.

Teufel, S., Siddharthan, A., and Batchelor, C. (2009). Towards domain-independent argumentative zoning: Evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1493–1502, Singapore. Association for Computational Linguistics.

Thorne, J. and Vlachos, A. (2017). An extensible framework for verification of numerical claims. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 37–40, Valencia, Spain. Association for Computational Linguistics.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Timmer, A., Hilsden, R. J., and Sutherland, L. R. (2001). Determinants of abstract acceptance for the digestive diseases week–a cross sectional study. *BMC medical research methodology*, 1(1):13.

Timmer, A., Sutherland, L. R., and Hilsden, R. J. (2003). Development and evaluation of a quality score for abstracts. *BMC medical research methodology*, 3(1):2.

Tindale, C. W. (2007). *Fallacies and argument appraisal*. Cambridge University Press.

Toledo, A., Gretz, S., Cohen-Karlik, E., Friedman, R., Venezian, E., Lahav, D., Jacovi, M., Aharonov, R., and Slonim, N. (2019). Automatic argument quality assessment - new datasets and methods. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5625–5635, Hong Kong, China. Association for Computational Linguistics.

Toulmin, S. E. (1958). *The Uses of Argument*. University Press.

Tsoumakas, G. and Vlahavas, I. (2007). Random k-Labelsets: An ensemble method for multilabel classification. In Kok, J. N., Koronacki, J., Mantaras, R. L. d., Matwin, S., Mladenič, D., and Skowron, A., editors, *Machine Learning: ECML 2007*, pages 406–417, Berlin, Heidelberg. Springer Berlin Heidelberg.

Vargas-Campos, I. and Alva-Manchego, F. (2016). SciEsp: Structural analysis of abstracts written in Spanish. *Computación y Sistemas*, 20(3):551–558.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Vecchi, E. M., Falk, N., Jundi, I., and Lapesa, G. (2021). Towards argument mining for social good: A survey. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1338–1352, Online. Association for Computational Linguistics.

Venant, A., Asher, N., Muller, P., Denis, P., and Afantenos, S. (2013). Expressivity and comparison of models of discourse structure. In *Proceedings of the SIGDIAL 2013 Conference*, pages 2–11, Metz, France. Association for Computational Linguistics.

Vlachos, A. and Riedel, S. (2015). Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2596–2601, Lisbon, Portugal. Association for Computational Linguistics.

Wachsmuth, H., Al-Khatib, K., and Stein, B. (2016). Using argument mining to assess the argumentation quality of essays. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691, Osaka, Japan. The COLING 2016 Organizing Committee.

Wachsmuth, H., Kiesel, J., and Stein, B. (2015). Sentiment flow - a general model of web review argumentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 601–611, Lisbon, Portugal. Association for Computational Linguistics.

Wachsmuth, H., Naderi, N., Hou, Y., Bilu, Y., Prabhakaran, V., Thijm, T. A., Hirst, G., and Stein, B. (2017a). Computational argumentation quality assessment in natural language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, Valencia, Spain. Association for Computational Linguistics.

Wachsmuth, H., Potthast, M., Al-Khatib, K., Ajjour, Y., Puschmann, J., Qu, J., Dorsch, J., Morari, V., Bevendorff, J., and Stein, B. (2017b). Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, Copenhagen, Denmark. Association for Computational Linguistics.

Wachsmuth, H., Stein, B., and Ajjour, Y. (2017c). "PageRank" for argument relevance. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1117–1127, Valencia, Spain. Association for Computational Linguistics.

Wachsmuth, H., Trenkmann, M., Stein, B., and Engels, G. (2014). Modeling review argumentation for robust sentiment analysis. In *Proceedings of COL-*

ING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 553–564, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Wachsmuth, H. and Werner, T. (2020). Intrinsic quality assessment of arguments. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6739–6745, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Wadden, D., Lin, S., Lo, K., Wang, L. L., van Zuylen, M., Cohan, A., and Hajishirzi, H. (2020). Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Walker, M., Tree, J. F., Anand, P., Abbott, R., and King, J. (2012). A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817, Istanbul, Turkey. European Language Resources Association (ELRA).

Walsh, D. M., Tseng, B. P., Rydel, R. E., Podlisny, M. B., and Selkoe, D. J. (2000). The oligomerization of amyloid β-protein begins intracellularly in cells derived from human brain. *Biochemistry*, 39(35):10831–10839.

Walton, D., Reed, C., and Macagno, F. (2008). *Argumentation schemes*. Cambridge University Press.

Wang, L. and Ling, W. (2016). Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.

Wei, Z., Liu, Y., and Li, Y. (2016). Is this post persuasive? ranking argumentative comments in online forum. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 195–200, Berlin, Germany. Association for Computational Linguistics.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 4th edition.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wolfe, C. R., Britt, M. A., and Butler, J. A. (2009). Argumentation schema and the myside bias in written argumentation. *Written Communication*, 26(2):183–209.

Wührl, A. and Klinger, R. (2021). Claim detection in biomedical Twitter posts. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 131–142, Online. Association for Computational Linguistics.

Yang, A. and Li, S. (2018). SciDTB: Discourse dependency TreeBank for scientific abstracts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 444–449, Melbourne, Australia. Association for Computational Linguistics.

Yang, Q., Zhang, Y., Dai, W., and Pan, S. J. (2020). *Transfer learning*. Cambridge University Press.

Zhang, L. and Wang, H. (2014). Go climb a dependency tree and correct the grammatical errors. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 266–277, Doha, Qatar. Association for Computational Linguistics.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Zhang, Y., Song, K., Song, L., Zhu, J., and Liu, Q. (2014). Syntactic SMT using a discriminative text generation model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 177–182, Doha, Qatar. Association for Computational Linguistics.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

Appendix A

PRELIMINARY EXPERIMENTS

In this appendix we report preliminary experiments aimed at mining arguments in scientific abstracts as well as a pilot application in which we used the automatically identified units and relations to predict the acceptance/rejection of manuscripts in a computer-science venues.

The experiments described in this section and its results were presented at ArgMining 2019 (6th Workshop on Argument Mining)[1].

This appendix is a summary of (Accuosto and Saggion, 2019b), included in the workshop proceedings.

[1] webis.de/events/argmining-19/

A.1 Argument mining annotations

A.1.1 Data

In order to explore the possibility of leveraging discourse information for the identification of argumentative components and relations we add a new annotation layer to the Discourse Dependency Tree-Bank for Scientific Abstracts (SciDTB) (Yang and Li, 2018). SciDTB contains 798 abstracts from the ACL Anthology (Radev et al., 2009) annotated with elementary discourse units (EDUs) and relations from the RST Framework (Mann and Thompson, 1988). Poly-nary RST discourse relations are binarized in SciDTB by means of a *right-heavy* transformation in order to represent discourse structures as dependency trees (Li et al., 2014).

We consider a subset of the SciDTB corpus consisting of 60 abstracts from the Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) and transformed them into a format suitable for the GraPAT graph annotation tool (Sonntag and Stede, 2014),[2] which had been previously tailored to the specificities of our proposed annotation scheme, described in Section A.1.2.

The corpus enriched with the argumentation[3] level contains a total of 327 sentences, 8012 tokens, 862 discourse units and 352 argumentative units linked by 292 argumentative relations.

A.1.2 Annotation scheme

Several argumentation mining works (Lippi and Torroni, 2016b) use *claims* and *premises* as basic argumentative units. In the case of scientific discourse, however, it is frequent to find that claims are not explicitly stated in an argumentative writing style but are instead left implicit (Hyland, 1998).

[2] angcl.ling.uni-potsdam.de/resources/grapat.html

[3] The annotations are made available to download at scientmin.taln.upf.edu/argmin/scidtb_argmin_annotations.tgz

The description of the problem addressed in the paper, for instance, usually conveys implicit claims in relation to the relevance of the problem at stake and/or the adequacy of the proposed approach.

We introduce a fine-grained annotation scheme aimed at capturing information that accounts for the specificities of the scientific discourse, including the type of evidence that is offered to support a statement (e.g., background information, experimental data or interpretation of results). This can provide relevant information, for instance, to assess the *argumentative strength* of a text.

The types of proposed units considered in our scheme can be mapped–even if with a different level of granularity–to concepts in CoreSC (Liakata et al., 2010) and AZ categories, which would enable additional research on the potential of using existing annotated corpora for argument mining tasks.

Like (Peldszus and Stede, 2016), we consider EDUs as the minimal spans that can be annotated. Argumentative units can, in turn, cover multiple sentences.

The proposed units include:

- ***proposal*** (problem or approach)
- ***assertion*** (conclusion or known fact)
- ***result*** (interpretation of data)
- ***observation*** (data)
- ***means*** (implementation)
- ***description*** (definitions/other information)

In line with (Kirschner et al., 2015b), we adopt in our annotation scheme the classic ***support*** and ***attack*** argumentative relations and the two discourse relations ***detail*** and ***sequence***.

Fig. A.2 shows a subset of the argumentative components and relations annotated in an abstract from (Zhang and Wang, 2014),[4] including a *proposal* and two supporting units: an *assertion* and a *result*. Fig. A.1 shows the original discourse units and relations as annotated in SciDTB.

[4] aclweb.org/anthology/D14-1033

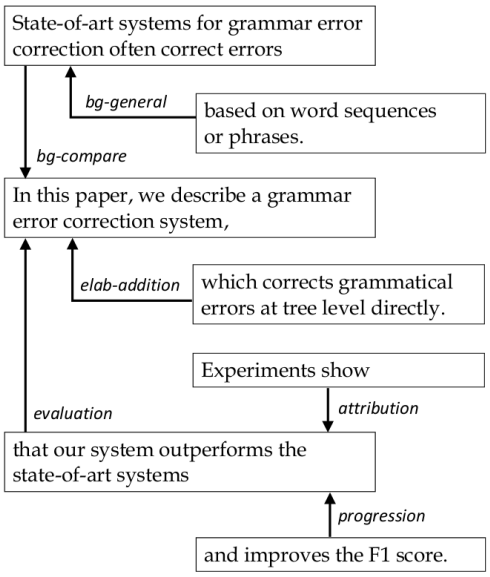


Figure A.1: Partial discourse structure

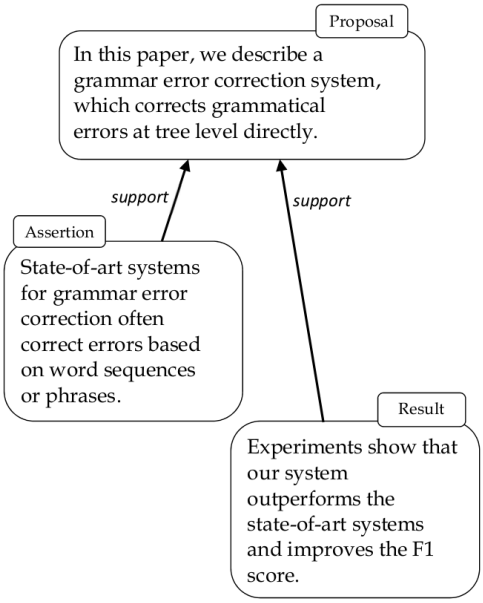


Figure A.2: Partial argumentative structure

In the subset of SciDTB annotated for our experiments, the types of argumentative units are distributed as follows: 31% of the units are of type *proposal*, 25% *assertion*, 21% *result*, 18% *means*, 3% *observation*, and 2% *description*.

In turn, the relations are distributed: 45% of type *detail*, 42% *support*, 9% *additional*, and 4% *sequence*. No *attack* relations were identified in the set of currently annotated texts.

When considering the distance[5] of the units to their parent unit in the argumentation tree, we observe that the majority (57%) are linked to a unit that occurs right before or after it in the text, while 19% are linked to a unit with a distance of 1 unit in-between, 12% to a unit with a distance of 2 units, 6% to a unit with a distance of 3, and 6% to a unit with a distance of 4 or more.[6]

[5]By *distance* we refer to the number of argumentative units that occur between two units in the text.

[6]According to the position of the parent unit, there are 200 relations pointing forward and 92 in which the parent occurs before in the text.

A.2 Argument mining experiments

The experiments in this section, are aimed at exploring the potential of applying a transfer learning method to improve the performance of argument mining tasks trained with a small corpus of 60 abstracts by leveraging the discourse annotations available in the full SciDTB corpus.

A.2.1 Tasks

We define the following set of argument mining tasks:

- **AFu (argumentative function)**: Identify the boundaries and argumentative functions of the components. In the example in Fig. A.2, it would imply to identify the boundaries of the three nodes and the two *support* relations that link them.
- **ATy (argumentative unit)**: Identify the boundaries and types of the components. In the example, the *proposal*, *assertion* and a *results* units.
- **APa (argumentative attachment)**: Identify the boundaries of the components and the relative position of the parent argumentative unit. For instance, the *assertion* unit in Fig. A.2 is attached to the *proposal* unit with a relative distance of one unit in the forward direction (as the assertion occurs right before the proposal in the text). The *result* unit, in turn, is attached to the *proposal* with a distance of four units in the background direction (the units that occur between these two nodes are omitted in the figure).

A.2.2 Experimental setup

We train each of the tasks described in A.2.1 separately and compare the results obtained with those obtained by an inductive transfer learning method in which we use encoders trained with the RST annotations available in the SciDTB corpus. These encoders are then used to produce contextualized representations of the input tokens that are fed to the argument mining learning processes.

The discourse parsing tasks considered to train the specialized encoders are:

- **DFu (discourse function)**: Identify the boundaries and discourse roles of the EDUs (*attribution*, *evaluation*, *progression*, etc.).
- **DPa (discourse attachment)**: Identify the boundaries of the EDUs and the relative position of the parent units in the RST tree.

The discourse tasks (DFu and DPa) are trained with the 738 abstracts left in the SciDTB corpus when excluding the 60 abstracts annotated with arguments. This is done in order to avoid introducing a bias that would not reflect the results obtained when no discourse annotations are available.

All of the argument mining models (AFu, ATy, APa) are trained and evaluated in a 10-fold cross-validation setting.

In all cases the models are generated by means of bi-directional long short-term memory (BiLSTM) networks, as this type of architecture has proven to perform reasonably well in argument mining tasks across different classification scenarios (Eger et al., 2017). In order to simplify the experiments and the interpretation of their results we use the same architecture for all tasks: two layers of 100 recurrent units, Adam optimizer, naive dropout probability of 0.25 and a conditional random fields (CRF) classifier as the last layer of the network. We use, for the BiLSTMs, the implementation made available by the Ubiquitous Knowledge Processing Lab of the Technische Universität Darmstadt (Reimers and Gurevych, 2017).[7] As our intention is to compare the different approaches and not necessarily obtain the best possible models for these tasks, no hyper-parameter optimization is done in these experiments and, in all of the cases, the networks are trained for 100 epochs.

All of the tasks are modeled as sequence labeling problems in which the tokens are tagged using the beginning-inside-outside (BIO) tagging scheme. The tokens are encoded as the concatenation of 300-dimensional dependency-based word embeddings (DEmb)[8] ($\vec{k}$) (Komninos and Manandhar, 2016) and 1024-dimensional contextualized word embeddings (ELMo) ($\vec{e}$) (Peters et al., 2018). In these experiments we use the 5.5 billion-token version of ELMo trained with Wikipedia and monolingual news from the WMT 2008-2012 corpora.[9]

[7] https://github.com/UKPLab/elmo-bilstm-cnn-crf

[8] https://www.cs.york.ac.uk/nlp/extvec/

[9] https://allennlp.org/elmo

For the experiments with the RST encoders we include the 200-dimensional embeddings obtained from the concatenation of the backward and forward hidden states of the top layers of the DFu or DPa models (RSTEnc) ($\vec{f}$ and $\vec{p}$, respectively). Table A.1 summarizes the sets of embeddings used in these experiments and their dimensions.

Each argument mining task is paired with one discourse parsing task for the transfer learning experiments. While AFu and ATy are paired with DFu, APa is paired with DPa. This means that the input for the AFu and ATy tasks is obtained as the concatenation of the vectors $[\vec{k}, \vec{e}, \vec{f}]$, while in the case of APa the input is $[\vec{k}, \vec{e}, \vec{p}]$.

Abbreviation	Notation	Dimensions
DEmb	$\vec{k}$	300
ELMo	$\vec{e}$	1024
GloVe	$\vec{g}$	200
RSTEnc (DFu/DPa)	$\vec{f}$ / $\vec{p}$	200

Table A.1: Word embeddings used in the experiments

A.2.3 Results

We adopt the ConNLL criteria for named-entity recognition[10] to evaluate the performances obtained in the identification of argumentative components and relations. Table A.2 shows the average F_1-measures obtained for each of the settings considering the epochs 10 to 100.[11]

Setting	AFu	ATy	APa
DEmb+ELMo	0.66	0.63	0.38
DEmb+ELMo+GloVe	0.65	0.65	0.38
DEmb+ELMo+RSTEnc	**0.69**	**0.67**	**0.40**

Table A.2: Average F1-measures in epochs 10-100

[10] A true positive is considered when both the boundary and the type of the entity match.

[11] The epochs before the 10th are not significant as the models have not had enough time to learn anything.

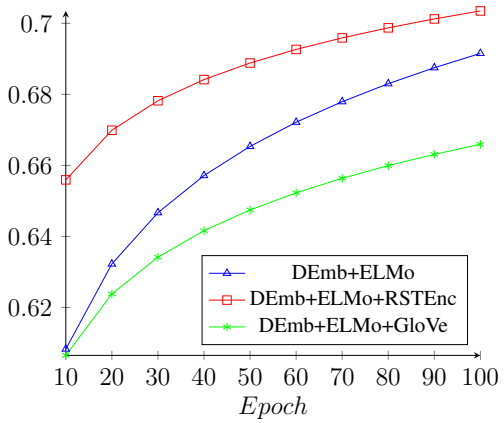


Figure A.3: Trend lines for F1-measures in epochs 10-100 for AFu

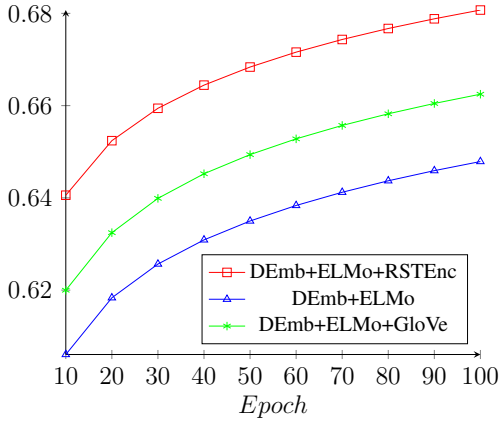


Figure A.4: Trend lines for F1-measures in epochs 10-100 for ATy

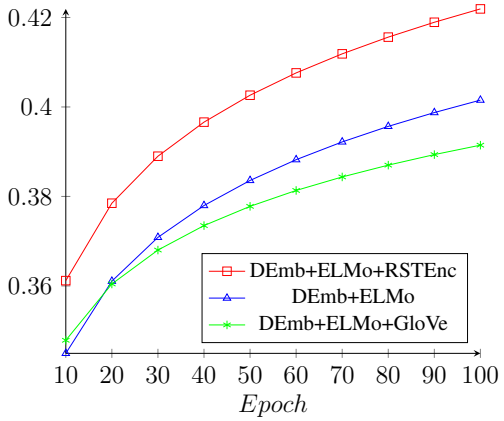


Figure A.5: Trend lines for F1-measures in epochs 10-100 for APa

The argument mining models trained with the representations produced by the RST encoders (*DEmb+ELMo+RSTEnc*) yield better performances, with gains of 0.03, 0.04 and 0.02 F_1 points for AFu, ATy and APa, respectively, over the models trained solely with the dependency-based and ELMo embeddings (*DEmb+ELMo*).

In order to determine whether the better performance of the RST encoders is due to the knowledge conveyed by the task-specific representations we conducted an additional experiment in which we concatenated 200-dimensional GloVe embeddings[12] (Pennington et al., 2014) ($\vec{g}$) obtaining 1524-dimension embeddings $[\vec{k}, \vec{e}, \vec{g}]$ used as input of each of the argument mining models. In this case, the results obtained are mixed, with an increase in performance of 0.02 F_1 points in average–for the epochs 10 to 100–for ATy, a worse performance of 0.01 F_1 points for AFu and no difference in performance for APa. The models with the GloVe embeddings (*DEmb+ELMo+GloVe*) have, therefore, worse performances in average of 0.04, 0.02 and 0.02 F_1 points for AFu, ATy and APa with respect to the models that include the embeddings obtained by means of the RST encoders.

Figures A.3, A.4 and A.5 show the trend-lines of F_1-measures obtained with the different models for the epochs 10 to 100 for the AFu, ATy and APa tasks, respectively. The graphs show that the models with information from the RST encoders not only learn better the argument mining tasks but they also do it in less time with respect to the other settings.

These results support out initial hypothesis in the sense that transferring discourse knowledge by means of representations learned in discourse parsing tasks can contribute to improve the performance of argument mining models trained with a rather small number of instances.

[12]We used the 6 billion tokens versions trained with Wikipedia 2014 and Gigaword 5 available at `https://nlp.stanford.edu/projects/glove/`

A.3 Acceptance prediction experiment

As a pilot application we explore the possibility of predicting the acceptance/rejection of papers in computer science conferences[13] based on the annotations generated by the best-performing argument mining models.

Quality assessment metrics that consider elements such as *clarity and simplicity, lack of redundancy and comprehensiveness* of scientific reporting have been developed for abstracts in other domains–in particular, in life sciences–(Timmer et al., 2003). These instruments were used in studies that show that abstracts with higher formal quality scores–as measured by human experts–are more frequently accepted for presentations in conferences (Timmer et al., 2001). We do not believe that these results can be directly extrapolated to the quality assessment of scientific abstracts in computer science, an area in which full manuscripts are most frequently considered for review and where abstracts have less fixed structures. Furthermore, clearer links between the formal quality of scientific reporting and the overall quality of research in computer science still need to be established. Considering all these limitations, we were interested in exploring whether the automatically identified argumentative structure of the abstracts could reflect some quality aspects of the full manuscripts and if this, in turn, could contribute to predict their acceptance in conferences in a specific research area in the field of computer science.

A.3.1 Dataset

As training set for the acceptance prediction experiment we use 117 abstracts of manuscripts submitted to the Compact Deep Neural Network Representation with Industrial Applications (CDNNRIA) and the Interpretability and Robustness for Audio, Speech and Language (IRASL) workshops held in the context of the Thirty-second Conference on Neural Information Processing Systems (NIPS 2018). As test set we use 30 abstracts of manuscripts submitted to the Sixth International Conference on Learning Representations (ICLR 2018). All of the abstracts were collected from the OpenReviews website (Soergel et al., 2013).[14]

[13] In particular, in the areas of neural-based systems and its applications to speech and language.
[14] `openreview.net`

The distribution of accepted/rejected papers in the training and test sets is shown in Table A.3

Set	Conference	Accepted	Rejected
Train	*CDNNRIA*	35	23
Train	*IRASL*	30	29
		55	52
Test	*ICLR*	15	15

Table A.3: Accepted/rejected papers in training and test sets

A.3.2 Experimental setup

The CDNNRIA, IRASL and ICLR abstracts are used as input to the AFu, ATy and APa models described in Section A.2 obtaining sequences of argumentative units, types and parent attachments. These sequences are then used as features to train and evaluate a binary classifier aimed at predicting the acceptance or rejection of the corresponding papers. Table A.4 shows sample training/test instances. As the number of argumentative units identified in each abstract might differ we use padding values (*nofunc*, *notype* and *100* for AFu, ATy and APa, respectively) to generate training and test instances with a fixed number of features (equal to three times the maximum number of argumentative units identified in the dataset).

Considering that we are dealing with a small set of features with a reduced number of potential values for each one, we use a decision tree algorithm for our pilot classification experiment. In addition to the training and evaluation speed of the algorithm we consider that the higher interpretability of the results–by examining the decision points–can also contribute to assess to what degree the different elements of the predicted argumentative structure are used in the classification. We use Weka's implementation of the C4.5 algorithm (Quinlan, 1993) (J48) with default parameters with the exception of the confidence factor used for pruning the tree, which was selected evaluating the different models obtained against a random split of 20% of the test set used for validation.[15]

[15] `weka.classifiers.trees.J48 -C0.6 -M2`

x_1	x_2	...	x_n
none	*additional*	...	*support*
support	*support*	...	*none*
...	...	...	...
support	*nofunc*	...	*nofunc*
proposal	*assertion*	...	*assertion*
result	*assertion*	...	*proposal*
...	...	...	...
observation	*notype*	...	*notype*
0	*1*	...	*1*
1	*1*	...	*0*
...	...	...	...
-5	*100*	...	*100*

y_1	y_2	...	y_n
REJECT	ACCEPT	â¦	ACCEPT

Table A.4: Example of input instances to the classifier

As the training set is not perfectly balanced, we pre-process the data with Weka's ClassBalancer algorithm, which assigns weights to each instance so that each class has the same total weight.

A.3.3 Results

The classifier trained with the argumentative units and relations extracted from the CDNNRIA/IRASL abstracts has a performance of 0.67 F_1-score when evaluated with the training set obtained from processing the ICLR abstracts,[16] 0.17 F_1 points above a random binary classification in a balanced set.

As expected, the main decision points in the tree correspond, broadly, to those attributes that are also ranked higher when measuring their contribution to reduce the entropy with respect to the class.[17]

[16] 20 of the abstracts were correctly classified and ten were mis-classified: five as false positives and five as false negatives

[17] As calculated by means of Weka's *InfoGainAttributeEval* algorithm.

Type		**Func.**		**Distance par.**	
proposal	210	*support*	381	*1*	242
assertion	522	*attack*	0	*2*	439
result	35	*detail*	69	*3*	69
observation	3	*additional*	73	*4*	24
means	15	*sequence*	3	*5*	9
description	1			*6*	0
				7	1

Table A.5: Statistics of predicted argumentative units and relations in the training set

Observing these features, we can see that the most relevant decision elements are the parent attachment of the first argumentative unit, the argumentative functions of the first two units and the argumentative type of the first unit. Also relevant are the features that mark the end of the sequences of argumentative types and functions for the majority of the instances. This means that the number of identified units also have a relevant role in the predictions. However, the number of units by itself is not a good predictor of the class. In fact, executing the same experiment but replacing the non-padding values for function, type and attachment for fixed values we obtain an F_1-measure of 0.59 due, in particular, to a higher number of false negatives (accepted papers classified as rejected).

Features	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F_1}$
Arg. units alone	0.67	0.53	0.59
Arg. units with types, functions and parents	0.67	0.67	**0.67**

Table A.6: Precision, recall and F1-measures for the acceptance prediction classifiers with and without fine-grained argumentative information