CONTEXT-AWARE ARGUMENT MINING AND ITS APPLICATIONS IN EDUCATION

by

Huy V. Nguyen

Bachelor of Engineering

Hanoi University of Sciences and Technologies, Vietnam

2007

Submitted to the Graduate Faculty of

the Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Huy V. Nguyen

It was defended on

04/14/2017

and approved by

Diane J. Litman, Department of Computer Science

Rebecca Hwa, Department of Computer Science

Adriana I. Kovashka, Department of Computer Science

Kevin D. Ashey, School of Law

Dissertation Director: Diane J. Litman, Department of Computer Science

ABSTRACT

CONTEXT-AWARE ARGUMENT MINING AND ITS APPLICATIONS IN EDUCATION

Huy V. Nguyen, PhD

University of Pittsburgh, 2017

Context is crucial for identifying arguments and argumentative relations in text, but existing argument studies have not addressed context dependence adequately. In this thesis, we propose *context-aware argument mining* that makes use of contextual features extracted from writing topics and context sentences to improve state-of-the-art argument component and argumentative relation classifications. The effectiveness as well as generality of our proposed contextual features is proven through its application in different argument mining tasks in student essays. We further evaluate the applicability of our proposed argument mining models in automated persuasive essay scoring tasks.

Keywords: argument mining, topic context, context segment, automated essay scoring.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

First of all, I would like to sincerely thank my advisor Professor Diane J. Litman for her boundless support, tremendous patience and motivation that she provided for my study from the start of my Ph.D. program at the University of Pittsburgh. Her valuable advices, insightful comments and critics helped me throughout my research and improved my writing. It was her courses which brought me to Artificial Intelligence and Natural Language Processing, and seeded my interest in the fields so I can achieve this success.

I would like to express my gratitude to Professor Kevin D. Ashley, Professor Christian Schunn and other members of the SWoRD's team at LRDC for all of the great collaboration that made my studies on educational data possible and practical. All meetings and discussions with the team helped grow my background to understand and appreciate Artificial Intelligence for Education. I would like to convey my thanks to professors Rebecca Hwa, Adriana I. Kovashka and Kevin D. Ashey for being part of my Ph.D. committee. Their comments, questions and encouragements helped me see other perspectives of my research and further improve my dissertation.

I greatly appreciate friends and members of ITSPOKE Lab and Computer Science Department who shared and helped me make the Ph.D. life less tough but more enjoyable. Pursuing a Ph.D. degree is a long term commitment and it will only be possible if one can find his lab and department the second home, as we did together.

Last but not least, I am deeply grateful to my parents Quy Nguyen and Loan Pham for their unconditional support, my wife Ngan Le who joined my life in the tough time and shared the hardship, my grand mother Nho Cao for her tender care, my sister Mai Nguyen for her encouragement, and my two little daughters Ha-My and Lam for bringing me the most precious gifts. I always knew that you all believed in me and wanted the best for me.

1.0 INTRODUCTION

Van Eemeren and Grootendorst defined argumentation as "a social, intellectual, verbal activity serving to justify or refute an opinion, consisting of a constellation of utterances which have a justifying or refuting function and being directed towards obtaining the agreement of a judge who is deemed to be reasonable" (Van Eemeren and Grootendorst, 1982). Originally proposed within the realms of Logic, Philosophy, and Law, computational argumentation has become an increasingly central core study within Artificial Intelligence (AI) through the connection with research in knowledge representation, non-monotonic reasoning and multi-agent systems (Bench-Capon and Dunne, 2007; Bentahar et al., 2010). In such areas, abstract argumentation in which each argument is regarded as atomic with no internal structure provides a formalism to model the reasoning process (Lippi and Torroni, 2015). While that high abstraction of argumentation facilitates modeling and analysis of attack relations and acceptability of arguments, it has no specification of what is an argument or an attack (Dung, 1995).

On the contrary, structured argumentation assumes a knowledge representation formalism to specify how arguments are constructed from components. Over the past decades, structured argumentation theories have gained an increasing interest as a vehicle for representing components of arguments and the interactions between components, evaluating arguments, and distinguishing legitimate from invalid arguments (Bench-Capon and Dunne, 2007). This, together with the rapid growth of textual data and tremendous advances in text mining, has brought the emergence of a new research area – argument (argumentation) mining in text[1] – to draw a bridge between formal argumentation theories and everyday life argumentative reasoning.

[1] Argument mining for short.

Aiming at automatically identifying argument components (e.g., premises, claims, conclusions) in natural language text, and the argumentative relations (e.g., support, attack) between components, argument mining is found to promise novel opportunities for opinion mining, automated essay evaluation as well as offers great improvement for current legal information systems or policy modeling platforms. Argument mining has been studied in a variety of text genres like legal documents (Moens et al., 2007; Mochales and Moens, 2008; Palau and Moens, 2009), scientific papers (Teufel and Moens, 2002; Teufel et al., 2009; Liakata et al., 2012), news articles (Palau and Moens, 2009; Goudas et al., 2014; Sardianos et al., 2015), user-generated online comments (Cabrio and Villata, 2012; Boltužić and Šnajder, 2014), and student essays (Burstein et al., 2003; Stab and Gurevych, 2014b; Rahimi et al., 2014; Ong et al., 2014). Problem formulations of argument mining have ranged from the separation of argumentative from non-argumentative text, the classification of argument components and argumentative relations, to the identification of argumentation structures/schemes.

To illustrate different tasks in argument mining, let us consider a sample student essay in Figure 1. The first sentence in the example is the writing prompt. The *MajorClaim* which states the author's stance towards the writing topic is placed at the first sentence of the essay's body, i.e., sentence 1. The student author used different *Claims* (controversial statements) to validate/support and attack the major claim, e.g., claims in sentences {2, 5, 8}. Validity of the claims are underpinned/rebutted by *Premises* (reasons provided by the author), e.g., premises in sentences {5, 6, 7}.

As the first task in argument mining, *Argument Component Identification* aims at recognizing argumentative portions in the text (Argumentative Discourse Units – ADUs (Peldszus and Stede, 2013)), e.g., a subordinate clause in sentence 1, or the whole sentence 2, and classifying those ADUs accordingly to their argumentative roles, e.g., *MajorClaim*, *Claim*, and *Premise*. The two sub-tasks are often combined into a multi-way classification problem by introducing the *None* class. Thus, possible class labels for a candidate ADU are {*MajorClaim*, *Claim*, *Premise*, *None*}. However, determining boundaries of candidate ADUs to prepare input for argument mining models is a nontrivial preprocessing task. In order to simplify the main argument mining task, sentences are usually taken as primary units (Moens et al., 2007), or the gold-standard boundaries are assumed available (Stab and Gurevych, 2014b).

Essay 75: $^{(0)}$Do arts and music improve the quality of life?
$^{(1)}$My view is that the [*government should give priorities to invest more money on the basic social welfares such as education and housing instead of subsidizing arts relative programs*]$_{MajorClaim}$.
$^{(2)}$[*Art is not the key determination of quality of life, but education is*]$_{Claim}$. $^{(3)}$[*In order to make people better off, it is more urgent for governments to commit money to some fundamental help such as setting more scholarships in education section for all citizens*]$_{Premise}$. $^{(4)}$This is simply because [*knowledge and wisdom is the guarantee of the enhancement of the quality of people's lives for a well-rounded social system*]$_{Premise}$.
$^{(5)}$Admittedly, [*art, to some extent, serve a valuable function about enriching one's daily lives*]$_{Claim}$, for example, [*it could bring release one's heavy burden of study pressure and refresh human bodies through a hard day from work*]$_{Premise}$. $^{(6)}$However, [*it is unrealistic to pursuit of this high standard of life in many developing countries, in which the basic housing supply has still been a huge problem with plenty of lower income family have squeezed in a small tight room*]$_{Premise}$. $^{(7)}$By comparison to these issues, [*the pursuit of art seems unimportant at all*]$_{Premise}$.
$^{(8)}$To conclude, [*art could play an active role in improving the quality of people's lives*]$_{Premise}$, but I think that [*governments should attach heavier weight to other social issues such as education and housing needs*]$_{Claim}$ because [*those are the most essential ways enable to make people a decent life*]$_{Premise}$.

Figure 1: A sample student essay taken from the persuasive essay corpus (Stab and Gurevych, 2014a). The essay has sentences numbered and argument components enclosed in tags for easy look-up.

The second task, *Argumentative Relation Classification*(Stab and Gurevych, 2014b), considers possible pairs of argument components in a definite scope, e.g., paragraph or pairs of argument component and argument topic. For each pair, the task is to determine if a component supports or attacks the other component. The definite scope is necessary to make the distribution less skewed. In fact, the number of pairs that hold an argumentative relation is far smaller than the total number of possible pairs. In the example essay, the *Claim* in sentence 2 supports the *MajorClaim* in sentence 1: $Support(Claim^{(2)}, MajorClaim^{(1)})$. We also have $Attack(Claim^{(5)}, MajorClaim^{(1)})$, $Support(Premise^{(5)}, Claim^{(5)})$. Given the direct relations as in the examples, one can infer $Attack(Premise^{(5)}, MajorClaim^{(1)})$ and so on.

While in argumentative relation classification one does not differentiate direct and inferred relations, *Argumentation Structure Identification* (Mochales and Moens, 2011) aims

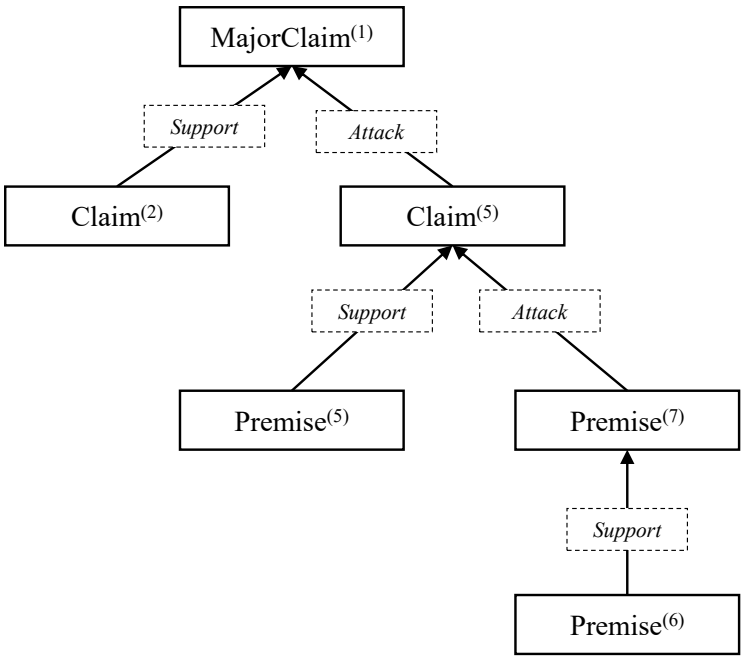

Figure 2: Graphical representation of a part of argumentation structure in the example essay. Argumentative relations are illustrated based on annotation by (Stab and Gurevych, 2014a).

at constructing the graphical representation of argumentation in which edges are direct attachments between argument components. Attachment is an abstraction of support/attack relations, and is illustrated as arrowhead connectors in Figure 2. Attachment between argument components does not necessarily correspond to the components' relative positions in the text. For example, $Premise^{(6)}$ is placed between $Claim^{(5)}$ and $Premise^{(7)}$ in the essay, but $Premise^{(7)}$ is the direct premise of $Claim^{(5)}$ as shown in the figure.

1.1 AN OVERVIEW OF MY THESIS WORK

In education, teaching argumentation and argumentative writing to students are in particular need of attention (Newell et al., 2011; Barstow et al., 2015). Automated essay scoring (AES) systems have been proven effective to reduce teachers' workload and facilitate writing

practices, especially in large-scale (Shermis and Burstein, 2013). AES research has recently shown interest in automated assessment of different aspects of written arguments, e.g., evidence (Rahimi et al., 2014), thesis and argument strength (Persing and Ng, 2013, 2015). However, the application of argument mining in automated argumentative essay scoring has been studied limitedly (Ong et al., 2014; Song et al., 2014). Motivated by promising results of argument mining as well as a desire of automated support for argumentative writings in school, this research aims at building models that automatically mine arguments in natural language text, and applying argument mining outcome to automatically score argumentative essays.

In particular, this thesis proposes *context-aware argument mining models* to improve state-of-the-art argument component and argumentative relation classifications. In order to make the proposed approaches more applicable to the educational context, the current research conducts both intrinsic and extrinsic evaluations when comparing the proposed models to the prior work. Regarding intrinsic evaluation, the current research performs both random folding cross validation and cross-topic validation to assess robustness of the models. For extrinsic evaluation, this thesis investigates the uses of argument mining for automated essay scoring. Overall, our research on argument mining can be divided into three components with respect to their functional aspects.

1.1.1 Context-aware Argument Mining Models

The main focus of the current research is to build models for argument component identification and argumentative relation classification. Context is crucial for identifying argument components and argumentation structures (Stab and Gurevych, 2014a). However, context dependence has not been addressed adequately in prior work (Stab et al., 2014). Most argument mining studies built prediction models that process each candidate ADU in argument component identification, or pair of argument components in argumentative relation classification, isolatedly from the surrounding text. To enrich the feature space of such models, history features such as argumentative roles of one or more preceding components, and features extracted separately from preceding and/or following text spans have been usually used

(Teufel and Moens, 2002; Hirohata et al., 2008; Palau and Moens, 2009; Guo et al., 2010; Stab and Gurevych, 2014b). However, the idea of using surrounding text as a context-rich representation of the prediction input for feature extraction was studied limitedly in prior research (Biran and Rambow, 2011).

In many writing genres, e.g., debates, student essays, scientific articles, the availability of writing topics provides valuable information to help identify argumentative text as well as classify their argumentative roles (Teufel and Moens, 2002; Levy et al., 2014). Especially, Levy et al. (2014) defined the term Context Dependent Claim to emphasize the role of discussion topic in distinguishing claims relevant to the topic from the irrelevant statements. The idea of using topic and discourse information to help resolve ambiguities are commonly used in word sense disambiguation and sentiment analysis (Navigli, 2009; Liu, 2012).

Based on the above observations, we hypothesize that argument component identification and argumentative relation classification can be improved with respect to prediction performance by considering contextual information at both local and global levels when developing prediction features. This thesis differentiates between global context and local context. While global context refers to the main topic/thesis of the document, the local context is instantiated by the actual text segment covering the textual unit of interest, e.g., preceding and following sentences.

Instead of building prediction models that process each textual input isolatedly, the proposed context-aware approach considers the input within its *context window* to enable advanced contextual features for argumentative relation classification.

Definition 1. *The context window of a textual unit is a text segment formed by neighboring sentences and the unit itself. The neighboring sentences are called context sentences, and must be in the same paragraph with the textual unit.*

The term "context sentences" was used by Qazvinian and Radev (2010) to refer to sentences surrounding a citation, that contain information about the cited source but do not explicitly cite it. In this thesis, we place no other constraint to context sentences than requiring them to be adjacent to the textual unit. Our approach aims at extracting discourse relations within the context window to better characterize the rhetorical function of the unit

in the entire text. In addition, the context windows instead of their units will be fed to textual entailment and semantic similarity scoring functions to extract semantic relation features. We expect that the aggregated semantic score (e.g., entailment and semantic similarity) computed from possible pairs extracted from two windows better represents the semantic relations of the two input units than their single score. As defining the context and identifying boundaries of context window are not a focus of this thesis, this thesis proposes to use different heuristics, e.g., window-size and text segmentation, to approximate the context window given a textual unit, and evaluate the contribution of such techniques to the final argument mining performance.

As for a global context, this thesis proposes an approach that uses writing topics to guide a semi-supervised process for separating *argument words* from *domain words*.

Definition 2. *Argument words are words that signal the argumentative content and are commonly used across different argument topics, e.g., 'believe', 'opinion'. In contrast, domain words are specific terminologies commonly used within the topic, e.g., 'art', 'education'. Domain words are a subset of content words that form the argumentative content.*

The above definition of argument and domain words shares similarities with the idea of shell language and content in (Madnani et al., 2012) in that it aims to model the lexical signals of argumentative content. The extracted vocabularies of argument words and domain words are then used to derive novel features and constraints for an argument component identification model.

1.1.2 Intrinsic Evaluation: Cross validation

In educational settings, students can have writing assignments in a wide range of topics. Therefore a desired argument mining model that has practical application in student essays is the one that can yield good performance for new essays of different topic domains than those of the training essays. As a consequence, features which are less topic-specific will be more predictive when cross-topic evaluated. Given this inherent requirement to the argument mining tasks for student essays, this research emphasizes the evaluation of the robustness of argument mining models. In addition to k-fold cross-validation (i.e., training and testing data

are randomly split from the corpus), the current research also conduct cross-topic validation (i.e., training and testing data are from essays of different writing topics) when comparing the proposed approaches with prior studies (Burstein et al., 2003).

For both cross-fold and cross-topic validations, we use different corpora to evaluate the effectiveness of the proposed approaches. The first corpus consists of 90 persuasive essays and the associated coding scheme specifying three different types of argument components: Major Claim, Claim, and Premise (Stab and Gurevych, 2014a). The coding scheme was then revised for use in a more expensive annotation study which yielded 420 annotated persuasive essays (Stab and Gurevych, 2017). The third corpus are academic writings collected from college Psychology classes and has sentences classified based on their argumentative roles: hypothesis, support finding, opposition finding, or non-argumentative (Barstow et al., 2015). We directly compare the proposed argument mining approaches to state-of-the-art models (Stab and Gurevych, 2014b, 2017).

1.1.3 Extrinsic Evaluation: Automated Essay Scoring

Aiming at high performance and robust models of argument mining, the second goal of this thesis is to apply argument mining in automated argumentative essay evaluation. As proposed in the literature, a direct approach would be using prediction outcome (e.g., arguments identified by prediction models) to call students' attention to not only the organization of their writings but also the plausibility of the provided arguments in the text (Burstein et al., 2004; Falakmasir et al., 2014). Such feedback information also helps teachers quickly evaluate writing performance of their students for better instructions. However, deploying an argument mining model to an existing computer-supported writing service, and evaluating its benefit to student learning would require a great amount of time and effort. Thus, it is set up as the long-term goal of our research. In the course of this thesis, we instead look for answers to the question of whether the outcome of automated argument mining can predict essay scores (Ghosh et al., 2016; Klebanov et al., 2016; Wachsmuth et al., 2016).

In a recent study, Ghosh et al. (2016) annotated a set of persuasive essays following the coding scheme in (Stab and Gurevych, 2014a) and evaluated a set of coarse-grained

argumentation features for persuasive essay scoring. In a similar vein of study, Klebanov et al. (2016) investigated a relationship between argumentation content and structure with essay quality using TOEFL11 corpus (Blanchard et al., 2013). We follow their settings to conduct AES experiments but further leverage the extrinsic evaluation of argument mining models to reveal how their accuracy impacts the performance of automated persuasive essay scoring. By argument mining accuracy, we mean the classification performance of each basic argument mining task.

Moreover, this thesis explores the value of argument mining in AES by investigating how much argumentation content and structure contribute to AES performance in comparison with other frequently used features of essay such as word-count, lexicon. We both proposed a compatitive base model for AES as well as use the Enhanced AI Scoring Engine (EASE) library[2] to extract features from essays for base AES models.

When an AES system is trained using essays from a specific writing prompt, it usually suffers from low performance when used on essays of different prompts. Because obtaining a large number of manually graded essays each time a new prompt is introduced is costly, domain adaptation is highly desired but yet challenging when designing AES systems. Regarding this matter, we evaluate how well argumentation content and structure perform in AES when training and test essays are of different prompts. Argumentative essays for evaluation are collected from the Automated Student Assessment Prize (ASAP) Competition[3] sponsored by the Hewlett Foundation in 2012.

1.2 THESIS STATEMENTS

Motivated by the benefit of using contextual information in writing topics and context windows in argument mining, this thesis proposes context-aware argument mining approaches that make use of additional context features derived from such contextual information. This thesis aims to support the following hypotheses of *the effectiveness of the proposed context*

[2]https://github.com/edx/ease
[3]http://www.kaggle.com/c/asap-aes

features:

- **H1**. The proposed contextual features improve the argument component identification in student essays.
- **H2**. The proposed contextual features improve the argumentative relation classification in student essays.
- **H3**. Prediction output of end-to-end argument mining provides effective features for automated argumentative essay scoring.

1.3 CONTRIBUTIONS

Through supporting the above research hypotheses, the contributions of the thesis are revealed. The first contribution is an argument mining system that offers state-of-the-art performance.

- A novel algorithm to extract argument and domain words from text. As shown in subsequent studies, the extracted lexicons are essential to improve argument mining performance, especially in cross-topic validation. Although different approaches were proposed to learn different aspects of argumentative languages, e.g., language expressing claims vs. language organizing these claims, in argumentative text, this research is the first time that language aspects separation is brought into an application in argument mining.
- Innovative local-context features by exploiting context windows. While argument and domain words enable abstractions of topic-dependent information and thus make use of the global context of the topic domain, a context window captures the local relations between the input argument component and surrounding sentences. Our experiments show that local and global context information represent complementary aspects of the relation between two argument components, and combining the two sets of contextual features achieves the best performance.

- While the main focus of this thesis is to solve the two classification tasks in argument mining, i.e., argument component and argumentative relation classifications, we also develop a sequence labeling model to segment sentences into argumentative vs. non-argumentative phrases for the AES studies. As a consequence, an end-to-end argument mining system that performs the argument parsing pipeline is developed. Given a free text as input, the argument mining system can parse its sentences to identify different types of argument components, and determine argumentative relations among those components.

The second contribution of this thesis is a comprehensive study on the impact of argument content and structure on AES performance.

- The current research is the first to perform an extrinsic comparison of argument mining models for persuasive essay scoring. We evaluate argument mining models in two extreme cases where argument components were segmented manually versus automatically.
- We also study a larger set of argumentation features for persuasive essay scoring than prior studies. Our study not only compares argumentation features with word-count and sentence-count, but also more advanced features extracted by an existing AES program.
- Finally, we are the first to evaluate the generality of argumentation features in AES in both in-domain and cross-domain evaluations. Research has explored a wide variety of domain adaptation techniques for AES depending on whether annotated data from a target domain is available or not. Our study does not solve domain adaptation but uses argumentation features to capture off-topic argumentation strategies in persuasive essays, and thus are domain-independent.

1.4 THESIS OUTLINE

1.4.1 Background and Data

In the Chapter 2, we discuss argument mining from its theoretical fundamentals to existing computational studies in different domains, and briefly introduce recent research on argument

mining for automated essay scoring.

Chapter 3 presents in detail the three corpora for argument mining research. Our research utilizes two annotated corpora of persuasive essays, and a corpus of academic essays to prove the generality of the proposed approaches.

1.4.2 Argument Component Classification

Chapters 4 and 5 present the work on argument component classification and support the first hypothesis H1. In Chapter 4 we develop an algorithm for extracting argument and domain words to use as novel topic-context features and feature constraints. Chapter 5 presents the improved model which achieves state-of-the-art performance in two argument mining corpora.

1.4.3 Argumentative Relation Mining

Chapters 6 and 7 supports the second hypothesis H2 through presenting context-aware argumentative relation mining approaches that make use of topic and window-context features. From the idea of context-window, we not only introduce new discourse relation features but also leverage textual relation features to improve argumentative relation mining in different corpora. We also experiment with different heuristics for forming context-windows of argument components.

1.4.4 End-to-end Argument Mining

Given the improvements made to argument mining tasks, Chapter 8 compares the proposed end-to-end argument mining system with the state-of-the-art models. The proposed system significantly outperforms a pipeline argument mining system, and achieves performance close to a joint-prediction model.

1.4.5 Automated Score Prediction for Persuasive Essays

Our first extrinsic evaluation of argument mining is presented in Chapter 9 where two end-to-end argument mining systems in Chapter 8 are compared in terms of using argumentation features derived from argument mining output to predict essay scores. Argument mining systems are evaluated in two extreme cases that their inputs are manually or automatically identified argument components.

While the first extrinsic evaluation only considers argumentation features, Chapters 10 and 11 further leverage the evaluation by putting argumentation features in context of base models for automated essay scoring. Moreover, Chapter 11 emphasizes the value of argumentation features in cross-domain essay scoring. Findings in these three chapters support the third hypothesis H3.

1.4.6 Appendixes

- Appendix A lists two set of argument words that are extracted from unlabeled data and used in the proposed argument mining models.
- Appendix B summarizes the coding manual for academic essays used in our studies (see Chapters 5, 7).
- Appendix C gives an output example of the Bayesian segmentation algorithm. This algorithm is utilized to create segment contexts for our proposed argumentative relation mining models (Chapter 7).
- Appendix D presents the results of using argumentation features for predicting peer ratings in academic essays.
- Appendix E presents a preliminary study on essay score explanation with argumentation features. This is the first step towards an envisioned intelligent feedback system for argumentative writings.
- Appendix F investigates the cross-domain essay scoring problem from a new perspective of score scaling.

2.0 BACKGROUND

2.1 ARGUMENTATION THEORIES

From dialectics and philosophy, models of argumentation have spread to core areas of AI including knowledge representation, non-monotonic reasoning, and multi-agent system research (Bench-Capon and Dunne, 2007). This has given rise to computational argumentation with two main approaches which are abstract argumentation and structured argumentation (Lippi and Torroni, 2015). Abstract argumentation considers each argument as a primary element without internal structure, and focuses on the relation between arguments, or sets of them. In contrast, structured argumentation studies internal structure (i.e., argument components and their interaction) of argument that is described in terms of some knowledge representation formalism. While abstract argumentation which is also called macro argumentation considers argumentation as a process, structured argumentation considers argumentation as a product and is also called micro argumentation (Mochales and Moens, 2011; Stab et al., 2014). Structured argumentation models are those typically employed in argument mining where the goal is to extract argument components from natural language texts. In this section, we describe two notable structured argumentation theories which are *Macro-structure of Argument* by Freeman (1991), and *Argumentation Scheme* by Walton et al. (2008). From the provided description of argumentation theories, we expect to give a concise yet sufficient introduction of related argument mining studies from a theoretical perspective.

Among a vast amount of structured argumentation theories (Bentahar et al., 2010; Besnard et al., 2014), the *premise-conclusion* models of argument structure (Freeman, 1991; Walton et al., 2008) are the most commonly used in argument mining studies. In fact, the

three corpora of argumentative writings that are studied in this thesis have coding schemes derived from the premise-conclusion structure of argument. Walton et al. (2008) gave a simple and intuitive description of argument which specifies an argument as a set of statements consisting a conclusion, a set of premises, and an inference from the premises to the conclusion. In the literature, claims are sometimes used as a replacement of conclusion, and premises are mentioned as evidences or reasons (Freeley and Steinberg, 2008). The conclusion is the central component of the argument, and is what "we seek to establish by our argument" (Freeley and Steinberg, 2008). The conclusion statement should not be accepted without additional reasons provided in premises.

The second component of argument, i.e., premise, is therefore necessary to underpin the plausibility of the conclusion. Premises are "connected series of sentences, statements or propositions that are intended to give reason" for the conclusion (Freeley and Steinberg, 2008). In a more general representation, premise can either support or attack the conclusion (i.e., giving reason or refutation) (Besnard and Hunter, 2008; Peldszus and Stede, 2013; Besnard et al., 2014). Based on the premise-conclusion standard, argument mining studies have proposed different argumentative relation schemes to cope with the great diversity of argumentation in natural language texts, for instances claim justification (Biran and Rambow, 2011), claim support vs. attack (Stab and Gurevych, 2014b), verifiability of support (Park and Cardie, 2014).

While most premise-conclusion models do not differentiate functions of different premises,[1] they enable the *macro-structure* of arguments which specifies the different ways that premises and conclusions combine to form larger complexes (Freeman, 1991). In the *Macro-structure of Argument Theory* the term 'argument' is thus not for premises, but for the complex of one or more premises put forward in favor of the conclusion. For example, Freeman (1991) identified four main macro-structures of arguments: linked, serial, convergent, and divergent, to represent whether different premises contribute together, in sequence, or independently to one or multiple conclusions. An example of a complex macro-structures of argument is shown in Figure 3. Based on Freeman's theory, Peldszus and Stede (2013) expanded the macro-

[1]Toulmin's argument structure theory distinguishes the role of different types of premise, i.e., data, warrant, and backing, in the argument (Toulmin, 1958).

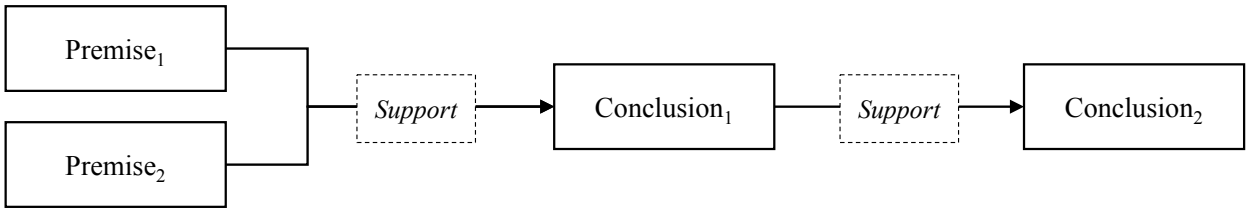

Figure 3: A complex macro-structure of argument consisting of linked structure (i.e., the support of Premise$_1$ and Premise$_2$ to Conclusion$_1$), and serial structure (i.e., the support of the two premises to Conclusion$_2$).

structure to cover more complex attack and counter-attack relations. In argument mining, the argumentation structure identification task aims at identifying the macro-structure of arguments in text (Palau and Moens, 2009; Peldszus and Stede, 2015; Persing and Ng, 2016; Stab and Gurevych, 2017).

Another notable construct of premise-conclusion abstraction is the *Argumentation Scheme Theory* (Walton et al., 2008). The authors used the argumentation scheme notion to identify and evaluate reasoning patterns commonly used in everyday conversational argumentation, and other contexts, notably legal and scientific argumentation. In Argumentation Scheme Theory, arguments are instances of abstract argumentation schemes each of which requires premises, the assumption implicitly holding, and the exceptions that may undercut the argument. Each scheme has a set of critical questions matching the scheme and corresponding to its premises, assumptions and exceptions, and such a set represents standard ways of critically probing into an argument to find aspects of it that are open criticism. Figure 4 illustrates the Argument-from-Cause-to-Effect scheme consisting of two premises and a conclusion. As we can see, argument schemes are distinguished by their content templates rather than their premise-conclusion structures. Identifying the argumentation scheme in the written argument has been considered to help recover implicit premises and re-construct the full argument (Feng and Hirst, 2011). On the other hand, research was also conducted to analyze the similarity and difference between argumentation schemes and discourse relations

Argument from cause to effect

- *Major premise*: Generally, if A occurs, then B will (might) occur.
- *Minor premise*: In this case, A occurs (might occur).
- *Conclusion*: Therefore, in this case, B will (might) occur.

Critical questions

1. *Critique the major premise*: How strong is the causal generalization (if it is true at all)?
2. *Critique the minor premise*: Is the evidence cited (if there is any) strong enough to warrant to the generalization as stated?
3. *Critique the production*: Are there other factors that would or will interfere with or counteract the production of the effect in this case?

Figure 4: Argumentation scheme: Argument from Cause to Effect.

(i.e., Penn Discourse Treebank discourse relations (Prasad et al., 2008)) which is considered a fruitful support of automated argument classification and process (Cabrio et al., 2013).

This thesis utilizes the annotated corpora compiled by Stab and Gurevych (2014a, 2017), whose annotation scheme is a simplification of the premise-conclusion model (see the first section of Chapter 1). The annotation scheme defines three types of argument components and considers only relations in argument component pairs. Thus, it ignores the more complex structures that may involve interaction, e.g., linked vs. convergent, between premises. We think such a simplification is reasonable for the purpose of a wide application to different text genres and the ease to develop prediction models with limited data. Argumentation schemes are also skipped in the annotation so that the focus is on direct support and opposition between argumentative content.

2.2 ARGUMENT MINING IN DIFFERENT DOMAINS

Argument mining is a relatively new research domain so its problem formulation is not well-defined but rather is considered potentially relevant to any text mining application

that targets to argumentative text (Mochales and Moens, 2011; Peldszus and Stede, 2013; Lippi and Torroni, 2015). Moreover, there is no consensus yet on an annotation scheme for argument components, or on the minimal textual units to be annotated. For these reasons, we follow Peldszus and Stede (2013) and consider in this study "argument mining as the automatic discovery of an argumentative text portion, and the identification of the relevant components of the argument presented there." We also borrow the term "argumentative discourse unit" to refer to the textual unit, e.g., text segment, sentences, clauses, which are considered as argument components (Peldszus and Stede, 2013).

In scientific domains, research has been long focusing on identifying the rhetorical status (i.e., the contribution to the overall text function of the article) of text segments, i.e., zone, to support summarization and information extraction of scientific publications (Teufel and Moens, 2002). Different zone mining studies were also conducted for different scientific domains, e.g., chemistry, biology, and proposed different zone annotation schemes that targets the full-text or only abstract section of the articles (Lin et al., 2006; Hirohata et al., 2008; Teufel et al., 2009; Guo et al., 2010; Liakata et al., 2012). However, none of the zone mining models described local interactions across segments and thus the embedded argument structures in text are totally ignored. Despite this mismatch between zone mining and argument mining, the two areas solve a similar core problem which is text classification, which makes zone mining an inspiration for argument mining models.

Two other domains that have argument mining intensively studied are legal documents and user-generated comments. In the legal domain, researchers seek for applications of automated recognition of arguments and argumentation structures in legal documents to support information retrieval, visualizing and qualifying arguments (Grabmair et al., 2015; Mochales and Moens, 2011). A wide range of argument mining tasks have been studied including argumentative text identification (Moens et al., 2007), sentence role identification in legal arguments (Grabmair et al., 2015; Bansal et al., 2016), argument component classification (i.e., premise vs. conclusion), and argumentation structure identification (Mochales and Moens, 2008; Palau and Moens, 2009). While the computational models for such argument mining tasks were evaluated using legal document corpora, those studies all employed the genre-independent premise-conclusion framework to represent the argument structure.

Therefore many prediction features used in argument mining models for legal text, e.g., indicative keywords for argumentation, discourse connectives, are generally applicable to other argumentative text genres, e.g., student essays. In fact, studies on argument mining in student essays including ours have taken advantage of solid work for scientific publications and legal documents to develop prediction features.

In user-generated comments, argument mining has been studied as a natural extension to opinion mining. While opinion mining answers what people think about for instance a product (Somasundaran and Wiebe, 2009), argument mining identifies reasons that explain the opinion. Among the first research on argument in user comments, Cabrio and Villata (2012) studied the acceptability of arguments in online debates by first determining whether two user comments support each other or not. In their study, arguments are users' pros and cons comments of the debate topic and were manually selected. Boltužić and Šnajder (2014) extended the work by mining user comments for more fine-grained relations, i.e., {explicit, implicit} $\times$ {support, attack}. Park and Cardie (2014) addressed a different aspect of argumentative relation which is the verifiability of argumentative propositions in user comments. While the task does not solve whether the given proposition is a support or opposition of the debate topic, it provides a mean to analyze the arguments in terms of the adequacy of their support assuming support/attack propositions are labeled already. From another aspect, predicting argumentative relations between user comments usually has multiple-sentence texts as input while argument mining in legal and scientific domains usually work at sentence/clause levels. For our research on argument mining in student essays, while the prediction problems are formulated as sentence/clause classification, our window-context features are inspired by prior work on argumentative relations of user comments.

Argument mining in student essays is rooted in argumentative discourse analysis for automated essay scoring (Burstein et al., 2003). In argumentative[2] writing assignments, students are given a topic and asked to propose a thesis statement and justify support for the thesis. Oppositions are sometime required to make the thesis risky and nontrivial (Barstow et al., 2015). Classifying argumentative elements in student essays has been used to support automated essay grading (Ong et al., 2014), peer review assistance (Falakmasir

[2]The term "persuasive" was also used as an equivalent (Burstein et al., 2003; Stab and Gurevych, 2014a).

et al., 2014), and providing writing feedback (Burstein et al., 2004). Burstein et al. (2003) built a discourse analyzer for persuasive essays that aimed at identifying different discourse elements (i.e., sentence) such as thesis, supporting idea, conclusion. Similarly, Falakmasir et al. (2014) aimed at identifying thesis and conclusion statements in student writings, and used the prediction outcome to scaffold peer reviewers of an online peer review system. Stab and Gurevych (2014a) annotated persuasive essays using a domain-independent scheme specifying three types of argument components (major claim, claim, and premise) and two types of argumentative relations (support and attack). Stab and Gurevych (2014b) utilized the corpus for automated argument component and argumentative relation identification. Ong et al. (2014) developed a rule-based system that labels each sentence in student writings in psychology classes with an argumentative role, e.g., hypothesis, support, opposition, and found a strong relation between the presence of argumentative elements and essay scores. Our context-aware argument mining models are developed and evaluated using the persuasive corpora developed by Stab and Gurevych (2014a, 2017), which have been used widely for argument mining studies. This allows us to not only compare our proposed models with the state-of-the-art, but also apply argument mining to student essay scoring.

2.3 ARGUMENT MINING TASKS AND FEATURES

2.3.1 Argument Component Identification

Argument component identification aims at determining the boundaries (i.e., begin and end tokens) of argument components in a sentence. While this is usually considered the first step in end-to-end argument mining systems, it is not always needed for some text genres. For example, our Academic Essay Corpus applies argumentative label, i.e., Hypothesis vs. Finding, to the whole sentence, so that does not require an identification of argument component (Barstow et al., 2015). In contrast, the Persuasive Essay Corpora have argument components, e.g., claim and premise, identified both identical or internal to sentences (Stab and Gurevych, 2014a, 2017), and there exist multiple-component sentences like the following:

> I think that [*governments should attach heavier weight to other social issues such as education and housing needs*]$_{Claim}$ because [*those are the most essential ways enable to make people a decent life*]$_{Premise}$.

Madnani et al. (2012) were among the first to address the problem of identifying the organizational elements, which they called "shell", in argumentative discourse. In the above sentence, the shell is detected as "I think that" and "because". Their study annotated a set of student essays and developed a supervised sequence model using the Conditional Random Field algorithm (Lafferty et al., 2001) to label each word in the sequence as shell or not. Similar sequence labeling approaches were also proposed in later studies on argument component, e.g., claim and premise, extraction in social media and persuasive essays (Goudas et al., 2014; Stab and Gurevych, 2017). Among a great variety of features, raw token, cue words, term frequency and term likelihood given label are commonly used and the most effective.

Argument component identification was also cast as a text classification problem. Levy et al. (2014) proposed a pipeline approach in which the first step detects topic-relevant sentences and the second step detects boundaries of claim in such sentences. They, however, extracted claims from a set of sub-sentence candidates, i.e., consecutive sequence of three tokens or more. Solving a classification on claim candidates gives them a flexibility to rank candidates and retrieve top instances, which is usually helpful for later information retrieval tasks. Persing and Ng (2015) utilized a persuasive essay corpus (Stab and Gurevych, 2014a) to develop a heuristic for extracting phrases from sentences. While their method did not directly solve the argument component identification, the extracted phrases were claimed to have a high coverage of argument components and fed as input to an argument component classification model.

In this thesis, we implement the sequence model proposed in (Stab and Gurevych, 2017) for argument component identification in our end-to-end argument mining system.

2.3.2 Argument Component Classification

To solve the argumentative label classification tasks (e.g., argumentative vs. not, premise vs. conclusion, rhetorical status of sentence), a wide variety of machine learning models

have been applied ranging from classification models, e.g., Naive Bayes, Logistic Regression, Support Vector Machine (SVM), to sequence labeling models such as Hidden Markov Model (HMM), Conditional Random Field (CRF). Especially for zone mining in scientific articles, sequence labeling is a more natural approach given an observation that the flow of scientific writing exposes typical moves of rhetorical roles across sentences. Studies have been conducted to explore both HMM and CRF for automatically labeling rhetorical status of sentences in scientific publications using features derived from language models and relative sentence position (Lin et al., 2006; Hirohata et al., 2008; Liakata et al., 2012).

In the realm of argument mining, argument component identification studies have been focusing on deriving features that represent the argumentative discourse while being loyal to traditional classifiers such as SVM, Logistic Regression. Sequence labeling models were not used mostly due to the loose organization of natural language texts, e.g., student essays, user comments, that are studied here. Prior studies have often used seed lexicons, e.g., indicative phrases for argumentation (Knott and Dale, 1994), discourse connectives (Prasad et al., 2008), to represent the organizational shell of argumentative content (Burstein et al., 2003; Palau and Moens, 2009; Stab and Gurevych, 2014b; Peldszus, 2014). While the use of such lexicons was shown to improve prediction output, their coverage is far from efficient given the great diversity of argumentative writing in terms of both topic and style.

Given the fact that the argumentative discourse consists of a language used to express claims, evidences and another language used to organize them, researchers have explored both supervised and unsupervised approaches to mine the organizational elements of argumentative text. Madnani et al. (2012) used CRF to train a supervised sequence model using simple features like word frequency, word position, regular expression patterns. To leverage the availability of large amount of unprocessed data, Séaghdha and Teufel (2014) and Du et al. (2014) built topic models based on LDA (Blei et al., 2003) to learn two language models: topic language and shell language (rhetorical language, cf. (Séaghdha and Teufel, 2014)). While Madnani et al. (2012) and Du et al. (2014) used data which were annotated for shell boundaries to evaluate how well the proposed model separates shell from content, Séaghdha and Teufel (2014) showed that features extracted from the learned language models help improves a supervised zone mining model. In a similar vein, we post-process LDA output

to extract argument and domain words which are used to improve the argument component identification.

In addition, contextual features were also applied to represent the dependency nature of argument components. The most popular are history features that indicate the argumentative label of preceding one or more components, and features extracted from preceding and following components (Teufel and Moens, 2002; Palau and Moens, 2009; Liakata et al., 2012; Stab and Gurevych, 2014b). In many writing genres, e.g., debate, essay, scientific article, the availability of argumentative topics provide valuable information to help identify argumentative portions in text as well as classify their argumentative roles. Levy et al. (2014) proposed the context-dependent claim detection task in which a claim is determined with respect to a given context - i.e., the input topic. To represent the contextual dependency, the authors made use of cosine similarity between the candidate sentence and the topic as a feature. For scientific writings, genre-specific contextual features were also considered including common words with headlines, section order (Teufel and Moens, 2002; Liakata et al., 2012). As for context features, we use writing topic to guide the separation of argument words from domain words. We also use common words with surrounding sentences and with writing topic as features.

2.3.3 Argumentative Relation Classification

The next step of identifying argument components is to determine the argumentative relations, e.g., attack and support, between those components, or between arguments formed by those components. Researchers have explored different argumentative relation schemes that can be applied to a pair of components, e.g., support vs. not (Biran and Rambow, 2011; Cabrio and Villata, 2012; Stab and Gurevych, 2014b), implicit and explicit support and attack (Boltužić and Šnajder, 2014). Because the instances being classified are pairs of textual units, features usually involve information from both elements (i.e., source and target) of the pair (e.g., word pair, discourse indicators in source and target) and the relative position between them (Stab and Gurevych, 2014b). Beyond features from superficial level, features were also extracted from semantic level of the relation including textual entailment

and semantic similarity (Cabrio and Villata, 2012; Boltužić and Šnajder, 2014). Based on those ideas, our research combines semantic relations with window segments to leverage the use of contextual features for argumentative relation mining.

Unlike argument component identification where textual units are sentences or clauses, textual units in argumentative relation classification vary from clauses (Stab and Gurevych, 2014b) to multiple sentences (Biran and Rambow, 2011; Cabrio and Villata, 2012; Boltužić and Šnajder, 2014). However, only little research has investigated the use of discourse relations within the text fragment to support the argumentative relation prediction. Biran and Rambow (2011) proposed that justifications of claims usually contain discourse structure which characterizes the argumentation provided in the justification in support of the claim. On the other hand, Cabrio et al. (2013) studied the similarities and differences between Penn Discourse Treebank (Prasad et al., 2008) discourse relations and argumentation schemes (Walton et al., 2008) and showed that some PDTB discourse relations can be appropriate interpretations of particular argumentation schemes. Inspired by these pioneering studies, our thesis proposes to consider each argumentative unit in its relation with other surrounding text to enable advanced features extracted from the discourse context of the unit.

2.3.4 Argumentation Structure Identification

In contrast to the argumentative relation task, argumentation structure task emphasizes the attachment identification that is to determine if two argument components directly attach to each other, based on their rhetorical functions for the persuasion purpose of the text. Attachment is considered a generic argumentative relationship that abstracts both support and attack and is restricted to tree-structures in that a node attaches (has out-going edge) to only one other node, while can be attached (has in-coming edge) from one or more other nodes. Palau and Moens (2009) viewed legal argumentation as rooted at a final decision that is attached by conclusions which are further attached by premises. They manually examined a set of legal texts and defined a context-free argumentative grammar to show a possibility of argumentative parsing for case law argumentation. Peldszus and Stede (2015) similarly assumed the tree-like representation of argumentation that has central claim be the root

node to which claims point (i.e., support or attack). Their data-driven approach took a fully-connected graph of all argument components as input and determined the edge weights based on features extracted from each component such as lemma, part-of-speech, dependency, as well the relative distance between the components. The minimum spanning tree of such weighted graph is returned as the output argumentation structure of the text. Assuming that premises, conclusions and their attachment were already identified, Feng and Hirst (2011) aimed at determining the argumentation scheme (Walton et al., 2008) of the argument with the ultimate goal of recovering the implicit premises (enthymemes) of arguments. Besides the general features (relative position between conclusion and premises, number of premises) the study included scheme-specific features which are different for each target scheme (in one-vs-others classification) and based on pre-defined keywords and phrases.

A challenge to our context-aware argument mining model is to determine the right context window given the argument component. An ideal context window is the minimal segment that expresses a complete justification in a support of the argument component. Thus, identifying the ideal context window of an argument component requires identifying the argumentation structure. To make the context-aware argument mining idea more practical and easier to implement, our research does not require sentences in a context window to be semantically or topically related while some kind of relatedness among those sentences might be useful for the final argument mining tasks. In the course of this thesis, context windows are determined using simple heuristics such as window-size and text segmentation output. In the future, argument structure identification for determining context windows is worth an investigation.

2.3.5 End-to-End Argument Mining

Despite the fact that argument mining is a new research field, its tasks all find relevance in long-history research such as discourse parsing, sequence model, text classification. This advantage certainly boosts-up the development of argument mining and the creation of end-to-end argument mining systems (Palau and Moens, 2009; Persing and Ng, 2016; Stab and Gurevych, 2017). Given an unannotated text, an end-to-end argument mining system first

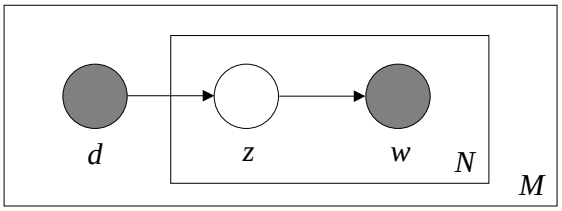

Figure 5: Probabilistic Latent Semantic Analysis

identifies argument components (or argumentative sentences) to prepare input for the argumentation structure tasks: argument component and argumentative relation classifications.

While classifying argument components and argumentative relations can be solved independently, the mutual information between them, e.g., argumentative relations are only allowed between certain types of argument components, suggests that exploiting prediction output of one task can improve the other. For example, predicted label of argument components has been used as an effective feature in argumentative relation mining (Stab and Gurevych, 2014b; Nguyen and Litman, 2016a). This suggests pipeline-based argument mining in which argument component and argumentative relation classifications are resolved in sequence. To better utilize the benefit of mutual information between argument components and argumentative relations, research also proposed to solve the two tasks jointly. The idea is that each task is first solved individually by base classifiers. The base classifiers assign labels to components and component pairs along with optional confidence scores. Then, a constrained optimization problem is formed to determine the best label assignment by resolving conflicts in the current assignments. In the approach proposed by Peldszus and Stede (2015), a complete argument graph is created where weights of edges between argument component are determined from output of base classifiers. The authors then solved a minimal spanning tree (MST) problem from the argument graph, which returned an argumentation structure in tree-like form.

In a different approach, Stab and Gurevych (2017) and Persing and Ng (2016) proposed Integer Linear Programming (ILP) frameworks to solve the argumentation structure tasks jointly. The proposed ILP frameworks used binary variables to represent labels of argument

component and argumentative relation, and prediction output of base classifiers are incorporated into objective functions. Both MST and ILP frameworks are generic and have no specific requirement on the base classifiers. This thesis, however, focuses on improving base classifiers for argument components and argumentative relations and developing a pipeline end-to-end argument mining. However, joint prediction frameworks should be easily applied to our system. We believe that by offering more accurate stand-alone models for argument component and argumentative relation classifications, we will improve the joint prediction.

2.4 TOPIC MODELS AND APPLICATIONS IN ARGUMENT MINING

2.4.1 Latent Dirichlet Allocation Topic Model

The principle idea in topic models is that documents are mixtures of topics, where a topic is a probability distribution over words (Blei et al., 2003; Hofmann, 1999, 2001; Steyvers and Griffiths, 2007; Griffiths and Steyvers, 2004; Blei, 2012). Hofmann (1999, 2001) introduced Probabilistic Latent Semantic Analysis (PLSA) that decomposes the joint probability of observing a term w and a document $\mathbf{d}$ with the use of a latent variable z which represent latent topics, where w and $\mathbf{d}$ are independent given z, and

$$P(w, \mathbf{d}) = P(\mathbf{d})P(w|\mathbf{d})$$
$$P(w|\mathbf{d}) = \sum_{z} P(w|z)P(z|\mathbf{d})$$

Figure 5 illustrates the plate digram of PLSA. Document $\mathbf{d}$ and word w are observed so they are represented by shaded nodes. Plates indicate repetition. The outer plate represents documents and the inner plate represents the repeated choices of topics and words within a document. PLSA assumes that a topic z is a distribution over a fixed size of vocabulary V, but does not explicitly specify this distribution. The model also assumes that a document $\mathbf{d}$ consists of multiple topics, but the distribution over that fix number of topics is not specified either. Therefore, in PLSA both topics and documents are represented as generic multinomial distributions, i.e., lists of numbers. Because PLSA does not define a generative

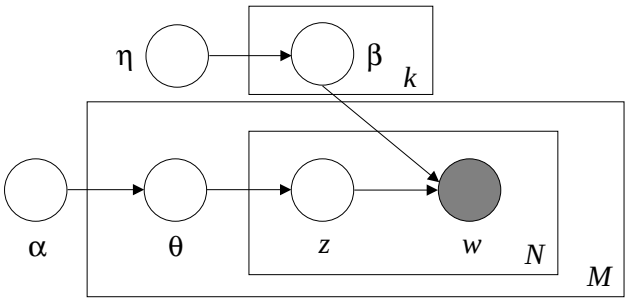

Figure 6: Latent Dirichlet Allocation

process for its topic distribution, the model exposes several problems: number of parameters increases linearly with the size of the training corpus and no way to assign probability to unseen documents (Blei et al., 2003).

Blei et al. (2003) extended PLSA by introducing a Dirichlet prior over the topic distribution and named the resulting generative model Latent Dirichlet Allocation. The model's graphical representation is shown in Figure 6. The generative process of each document d in a corpus D is as follows (Blei et al., 2003):

1. Decide on the number of words N the document will have: $N \sim \text{Poisson}(\xi)$.
2. Choose a topic mixture, i.e., multinomial distribution, θ for the document according to a Dirichlet distribution over a fixed set of k topics: $\theta = \text{Dir}(\alpha)$
3. Generate each word w_i in the document by:
   a. Picking a topic according to the multinomial distribution that was sampled above: $z_i = \text{Multinomial}(\theta)$.
   b. Choose a word w_i from $p(w_i|z_i, \beta)$

In this setting, the dimensionality k of of Dirichlet distribution (i.e., dimension of topic variable z) is provided and fixed. Alpha is a k-dimensional parameter vector with components $\alpha_i > 0$. Beta is a $k \times V$ matrix of word probability given topic, where $\beta_{ij} = p(w^j = 1|z^i = 1)$ and V is the vocabulary size. Each row of β is drawn independently from a Dirichlet distribution with a symmetric parameter vector, i.e., vector components are all equal to η.

Along with number of topics k, two hyper-parameters α, i.e., document-topic prior parameter, and β, i.e., word-topic prior parameter, need to set-up before run LDA. A simple implementation of LDA is to have symmetric Dirichlet priors when components in the parameter are the same. However, it has been shown that asymmetric α performs better than a symmetric prior, while an asymmetric β is largely not more helpful than a symmetric prior (Wallach et al., 2009). Also, a general intuition on the magnitude of α and β is that higher α values mean documents contain more similar topic contents, and a high β will result in topics with more similar word contents.

Given a set of documents, different learning algorithms were proposed to learn the document-topic and word-topic probabilities including variational expectation maximization (Blei et al., 2003) and collapsed Gibbs sampling (Griffiths and Steyvers, 2004). Extensions to LDA has been proposed including hierarchical LDA (Teh et al., 2005), supervised LDA (Mcauliffe and Blei, 2008).

2.4.2 LDA Topic Modes in Argument Mining

LDA topic models have been recognized as a useful tool for analyzing large collections of free-text documents. Applications of LDA to natural language processing can be found in a wide variety of areas such as entity analysis (Newman et al., 2006), multi-document summarization (Haghighi and Vanderwende, 2009), word-sense disambiguation (Boyd-Graber et al., 2007). In opinion mining and sentiment analysis, LDA topic models were successfully used to separate topic and opinion words (Mei et al., 2007; Lin and He, 2009; Zhao et al., 2010; Jo and Oh, 2011). However, LDA has been studied limitedly in argument mining.

Madnani et al. (2012) were the first who proposed the idea of separating shell language, e.g., "The argument states that", from the language that specifies claims and evidences, e.g., "based on the result of the recent research, there probably were grizzly bears in Labrador." Du et al. (2014) based on the idea of HMM-LDA (Griffiths et al., 2005) and developed an unsupervised topic model, called Shell Topic Model, to separate shell phrases from topical contents. Their idea based on two assumptions. The first was that each word in the document is associated with a status variable which tells if the word has a shell, topic or function status.

Each status generates word using a multinomial distribution which in turn is sampled from a Dirichlet prior. Then, the authors assumed that there are transition probabilities between statuses, which follow a multinomial distribution.

In document zoning, the problem is to recognize the information structure of documents to help assist information extraction and organize factual information from the documents (Teufel and Moens, 2002). Varga et al. (2012) adapted LDA topic model to document zone classification (e.g., *introduction*, *method*, *results* ...) with assumptions that a document is a mixture of zones and a zone is a probability distribution over words. The authors also proposed a special zone, i.e., background zone, which contains common words of different zone types, e.g., "use", "determine". Thus, the generative process involves a decision of whether a word is sampled from the background zone or other regular zones.

While also adapting LDA topic model to document zoning, Séaghdha and Teufel (2014) replied on the intuition that rhetorical language used in a document is independent of the topic. Their proposed model assumes that each word is generated either from an LDA-style topic model (captures topic matter of the document) or from a distribution associated with the rhetorical category, i.e., zone type, of the sentence (captures conventional language). The resulting model combines Hidden Markov process and "switching variable" mechanism with original LDA. Their experiments showed that features from output of the topic model, e.g., zone index, yielded significant improvement to a feature-based model.

In this thesis, we hypothesize that argumentative text can be separated into argument words and domain words, and the extracted vocabularies of argument and domain words can be used to improve argument mining models. However, we do not modify but use the original LDA topic model to parse the texts and then process the output to extract argument and domain words.

2.5 ARGUMENT MINING FOR AUTOMATED ESSAY SCORING

Automated essay scoring (AES) is advancing greatly with the success of many commercial and open-source systems in real-world applications (Shermis and Burstein, 2013). With

argumentation and argumentative writing as a key focus of Common Core Standards, a natural need for AES systems is the ability to consider argumentation in writings. Research on AES has recently investigated possibilities of grading essays on argument aspects, e.g., evidence (Rahimi et al., 2014), thesis clarity (Persing and Ng, 2013), and argument strength (Persing and Ng, 2015).

Targeting to identifying the argumentation structure in argumentative writings, argument mining offers the complete solution for argumentation-aware AES systems (Klebanov et al., 2016). In a preliminary study, Song et al. (2014) proposed to annotate argument analysis essays to identify responses of critical questions to judge the argument in writing prompts. The annotation was then used as features to improve an existing essay scoring model. Ong et al. (2014) were one of the first who investigate the relation between argument statistics and essay scores. However, their model used hand-crafted rules to extract different type of argumentative discourse units.

Argument mining has recently gained much interest in enabling automated argumentation feature extraction for AES and shows promising results. Ghosh et al. (2016) proposed a wide range of statistical features based on types of argument components and argumentative relations. Their study showed that automatically generated argumentation features yield a high correlation with human scores, and only 7% lower than using true values of argumentation features. However, their implementation of argument mining considered true argument components as inputs and solved a simplified argumentative relation classification problem. Therefore, the results did not reflect the capability of argumentation features on scoring unannotated essays.

Klebanov et al. (2016) were the first to use end-to-end argument mining to parse persuasive essays for argumentation features. Their results reveal that adding argumentation features yielded improvement to AES in comparison to a length-only model. Our study further investigates application of argument mining for AES on different perspectives including impact of argument mining accuracy, cross-domain essay scoring, and more advanced AES baseline models.

To deal with error-propagation in end-to-end argument mining, Wachsmuth et al. (2016) made two simplifications: (1) each sentence corresponds to an argumentative discourse unit

(AUD), (2) each paragraph corresponds to an argument. With the first simplification the authors avoided the need for argument component identification. The second simplification implicitly assumes that argument components within each paragraph support each other, thus argumentative relation classification can be skipped. The argumentative structure of an essay is represented as a sequence of arguments and each argument as a sequence of ADU types. Their argument flow features were shown to be effective for scoring essay organization and to gain improvement for scoring argument strength. Our study also confirms that argument flow features disregarding granularity of ADU are effective for predicting holistic score of essays.

3.0 DATA SETS FOR ARGUMENT MINING TASKS

With the concentration on application of argument mining in student essays, our argument mining models are mainly evaluated using different corpora of argumentative writings by students and test takers. Despite such a fact, this thesis aims for demonstrating the generality of proposed approaches because of the data diversity in our study. In particular, we employ three annotated corpora that are different in terms of writing styles, argumentative labels, and coding manuals.

3.1 FIRST CORPUS OF PERSUASIVE ESSAYS

The first dataset for our study (referred to as Persuasive1) is a corpus of persuasive essays which were annotated in accordance with the initial coding manual proposed by Stab and Gurevych (2014a). The corpus consists of 90 persuasive essays which were posted to an online forum (www.essayforum.com). Those essays are practice writings in response to sample test questions of standardized English tests for ESL learners. Essays were posted to the forum by users for feedback from the community. In the essays, the writers state their opinions (labeled as *MajorClaim*) towards the writing topics and validate those opinions with convincing arguments consisting of controversial statements (i.e., *Claim*) that support or attack the Major Claims, and evidences (i.e., *Premise*) that underpin the validity of the Claims. Three experts were asked to identify possible argument components, i.e., *Major Claim*, *Claim*, *Premise*, within each sentence, and connect the argument components using argumentative relations: *Support* and *Attack*. An argumentative relation is a directed connection that specifies source and target components. The coding manual only allows

argumentative relations to be held between Premises, from Premises to Claims or Major Claims, and from Claims to Major Claims. Except for the argumentative relation between Claim and Major Claim, other argumentative relations do not cross paragraph boundaries. According to the coding manual, Major Claim, Claim and Premise are different types of Argumentative Discourse Unit (ADU), and an argument is formed by a complex of a sequence of such ADUs along with specific relations between them. A paragraph may contain one or more complete arguments.

An example of a persuasive essay in the corpus is given in the excerpt below. Essay sentences are numbered and argument components are enclosed in tags which show their argumentative labels.

Example essay 1: (0)Effects of Globalization (Decrease in Global Tension)
(1)During the history of the world, every change has its own positive and negative sides. (2)Globalization as a gradual change affecting all over the world is not an exception. (3)Although it has undeniable effects on the economics of the world; it has side effects which make it a controversial issue.
(4)[*Some people prefer to recognize globalization as a threat to ethnic and religious values of people of their country*]$_{Claim}$. (5)They think that [*the idea of globalization put their inherited culture in danger of uncontrolled change and make them vulnerable against the attack of imperialistic governments*]$_{Premise}$.
(6)Those who disagree, believe that [*globalization contribute effectively to the global improvement of the world in many aspects*]$_{Claim}$. (7)[*Developing globalization, people can have more access to many natural resources of the world*]$_{Premise}$ and [*it leads to increasing the pace of scientific and economic promotions of the entire world*]$_{Premise}$. (8)In addition, they admit that [*globalization can be considered a chance for people of each country to promote their lifestyle through the stuffs and services imported from other countries*]$_{Premise}$.
(9)Moreover, [*the proponents of globalization idea point out globalization results in considerable decrease in global tension*]$_{Claim}$ due to [*convergence of benefits of people of the world which is a natural consequence of globalization*]$_{Premise}$.
(10)In conclusion, [*I would rather classify myself in the proponents of globalization as a speeding factor of global progress*]$_{MajorClaim}$. (11)I think [*it is more likely to solve the problems of the world rather than intensifying them*]$_{Premise}$.

According to the coding manual, each essay has one and only one Major Claim. An essay sentence (e.g., sentence 9) can simultaneously have multiple argument components which are clauses of the sentence (Argumentative spans), and text spans that do not belong to any argument components (None spans). None spans can be as short as a single punctuation. An argument component can be either a clause or a whole sentence (e.g., sentence 4). Sentences that do not contain any argument component are labeled *Non-argumentative* (e.g., sentences

{1, 2, 3}). The three experts achieved inter-rater accuracy of 0.88 for argument component labels, Krippendorff (2004) α_U of 0.72 for argument component boundaries, and Krippendorff (1980) α of 0.81 for argumentative relations.

Forming prediction inputs for argument component classification from the corpus is complicated due to the multiple-component sentences. For an illustration, consider sentence 9 in the sample essay. We have the following text spans with their respective labels:

Text span	Label
Moreover,	None
the proponents of globalization idea point out globalization results in considerable decrease in global tension	Claim
due to	None
convergence of benefits of people of the world which is a natural consequence of globalization	Premise
.	None

As described in (Stab and Gurevych, 2014b), the None spans are not considered as prediction inputs. Stab and Gurevych (2014b) defined a proper input of their prediction model as either a Non-argumentative sentence or an Argumentative span. Overall, the Persuasive Essay Corpus has 327 Non-argumentative sentences and 1346 Argumentative sentences with 1552 argument components. The distribution of argumentative labels is shown in Table 1. With regards to argumentative relations, Table 2 reports numbers of Support and Attack relations with different constraints. It is notable that Premise and Support are the dominant classes which characterizes the style of persuasive essays that writers usually support each of their claims by several premises.

3.2 SECOND CORPUS OF PERSUASIVE ESSAYS

Stab and Gurevych (2017) compiled the second corpus of persuasive essays (Persuasive2) with 402 essays to address the small size of the first corpus. The essays were again persua-

Argumentative label	First corpus	Second corpus
Major Claim	90	751
Claim	429	1506
Premise	1033	3832
Non-argumentative	327	1631
Total	1879	7720

Table 1: Counts of argument components in two persuasive essay corpora.

Argumentative relation	First corpus	Second corpus
With paragraph constraint		
Support	989	3613
Attack	103	219
Between Claim – Major Claim		
Support	365	1228
Attack	64	278

Table 2: Counts of argumentative relations in two persuasive essay corpora.

sive writings selected from www.essayforum.com with similar criteria. However, the coding manual was revised with significant differences. First, the authors removed the restriction that each essay has only one major claim. Allowing multiple instances of major claims yielded less confusion when formulating major claims and claims.

Second, the argumentative relations are defined by level in which the first level is between claim and major claim. Because the major claim may have more than one appearance, support and attack relations from claim to major claim contract into the stance attribute of claims, which can take values *for* or *against*. The second and third levels of argumentative relation are from premise to claim, and premise to premise, respectively.

Third, the new coding manual defines that each argument consists of only one claim (viewed as its conclusion) and one or more premises (as reasons given justifying or refuting another argument components). As a consequence, argumentative relation is not allowed between claims, or from premise to major claim. While the annotation scheme of the second persuasive essay corpus consists of the same class label set as those of the first corpus, the new annotation scheme specifies argument component and argumentative relation more consistently. Three experts annotated 80 essays and obtained Krippendorff $\alpha_U = 0.77$ for argument component boundaries, Fleiss (1971) $\kappa = 0.71$ and 0.74 for support and attack relations, respectively. Data statistics of this corpus is reported in Tables 1 and 2.

3.3 ACADEMIC ESSAY CORPUS

The third corpus for our study consists of 115 student essays collected from a writing assignment of University Introductory Psychology classes in 2014 (Barstow et al., 2015). The assignment requires each student to write an introduction of an observational study that the student conducted. With regard to the observational study, each student proposes one or two hypotheses about effects of different observational variables to a dependent variable, e.g., effect of gender to politeness. Students are asked to use relevant studies/theories to justify support for the hypotheses, and to present at least one theoretical opposition with a hypothesis. Students are also required to write their introductions in the form of an argumentative essay and follow the APA guideline that use citations whenever students refer to prior studies. Comparing to the Persuasive Essay Copora, while claims in the persuasive essays are mostly substantiated by personal experience, hypotheses in the academic essays are elaborated by findings from the literature. This makes the most distinguished difference between the two types of student writing.

Two experts labeled each sentence of the essays as to whether it contains *Hypothesis* statement, *Support* finding, or *Opposition* finding. If so it is an *argumentative sentence*, and the experts highlighted the argumentative parts of the sentence. Because an essay can address more than one hypothesis, annotators were required to number hypothesis state-

Argumentative label	#sentences
Hypothesis	185
Finding	130
– *Support finding*	50 (46)
– *Opposition finding*	83 (79)
Non-argumentative	2999
Total	3314

Table 3: Counts of argumentative sentences in Academic Essay Corpus.

ments. If a sentence is identified as a Support or Opposition, it will be linked to the relevant hypothesis statement. Direct relation between finding sentences are not considered. The detailed coding manual is provided in Appendix B.

For the argument component classification problem, *Support* and *Opposition* sentences were grouped into *Finding* category to make data less skewed and shift the focus to argumentative roles as claim (hypothesis) and premise (finding). The argumentative relation mining problem then classifies each possible pair of argumentative sentences as support, opposition or no-relation.

The two annotators achieved inter-rater Cohen's kappa 0.79 for the agreement on sentence labels for the coding scheme *Hypothesis-Finding*. Inter-rater kappa is 0.67 for coding scheme *Hypothesis-Support-Opposition*.

As an example, two last paragraphs of an academic essay are given below. The essay's topic is "Amount of Bystanders Effect on Helping Behavior".[1]

Example essay 2:
(1)Several studies have been done in the past that also examine the ideas of the bystander effect and diffusion of responsibility, and their roles in social situations. (2)[Daniel M. Wegner conducted a study in 1978 that demonstrated the bystander effect on a college campus by comparing the ratio of bystanders to victim, which showed that *the more bystanders in comparison to the victims led to less people helping* (Wegner, 1983).]$_{Support}$ (3)[Another

[1]Topic sentence and content of the essay are shown as they were written by the student.

supporting study was conducted Rutkowski in 1983 that also demonstrated that *with larger groups comes less help for victims in non-emergency situations due to less social pressure* (Rutkowski, 1983).]$_{Support}$ (4)Although these studies demonstrate the bystander effect and diffusion of responsibility, other studies oppose these ideas. (5)[One strong study that opposes the bystander effect was done in 1980 by Junji Harada that showed that *increase in group size, even in a face to face proximity, did not decrease the likelihood of being helped* (Harada, 1980).]$_{Opposition}$
(6)In order to find out specifically the effects that the bystander effect has in diverse settings, this study focuses on a non-emergency situation on a college campus. (7)[The hypothesis, based on the bystander effect demonstrated in Wegner's study (1978), is that *with more people around, less people will take the time to help the girl pick up her papers.*]$_{Hypothesis}$

In the example, the main content of argumentative sentences that express the argumentative role of the sentences (e.g., *hypothesis*, *support*, or *opposition*) are italicized. Given the annotation, *Finding* sentences are {2, 3, 5}.

While the coding manual allows essay sentences to have multiple labels, annotators were not required to split each sentence into smaller ADUs. The reason was that no sentence has both hypothesis and finding content, and the number of multiple-label sentences is small (9 out of total 3314 sentences). In particular, two sentences contain more than one hypothesis, and seven sentences contain different support and/or opposition findings. Therefore, maintaining sentence as the primitive ADU does not cause trouble for argument component identification.

Table 3 shows the label distribution in the corpus. Because of multiple-label sentences, number of *Finding* sentences is smaller than total of sentences that contain *Support* or *Opposition*. Among 50 Support sentences, 46 sentences are single-label. There are 79 single-label sentence out of 83 Opposition sentences. As we can see, the dataset is very skewed with Non-argumentative sentences as more than 90% of the data. Also while each essay has at least one Hypothesis statement, not all essays have Support and Opposition sentences.

Argumentative relations in academic essays are defined from a finding sentence to its linked hypothesis sentence. There are cases that a finding sentence supports and/or opposes different hypotheses. However, there exist three tricky sentences that each contains findings that support and oppose the same hypothesis. Thus, each of those sentences will create both support and opposition relation to a hypothesis sentence, and violate the class label consistency. Because of the small number of argumentative support relations, we re-label

those three sentences as support findings, which creates three support relations and discards four opposition relations. Our final data after that adjustment contains 50 support and 82 opposition relations. Because of this adjustment, the number of opposition relations (i.e., pairs of opposition finding and relevant hypothesis) does not match number of opposition findings shown in the Table 3.

3.4 SUMMARY

In this chapter, we present three annotated corpora for argument mining in which one consists of academic writings by college students and the other two are persuasive essays by ESL learners. The data sets expose great differences in writing style, fluency, and annotation scheme. While the persuasive essay corpora have ADUs at clause level, the academic essay corpus works at sentence level. The argumentation structure of academic essays are simplified as a flat tree to capture only support and opposition relations between findings and hypotheses. Persuasive essays were annotated for more complex argument structures in which argumentation relations are determined by layers, i.e., premise to premise, premise to claim, and claim to major claims. These data diversity gives us a good opportunity to demonstrate the generality of our proposed approaches that we will present in the next chapters.

To avoid distraction to readers, we decided to not introduce the data sets for persuasive essay score prediction tasks in this chapter. Instead, we will present essay score data within each of chapters 9, 10, and 11 which are about our studies on applying argument mining for automated essay scoring.

4.0 EXTRACTING ARGUMENT AND DOMAIN WORDS FOR IDENTIFYING ARGUMENT COMPONENTS IN TEXTS

4.1 INTRODUCTION

Argument component identification studies often use lexical (e.g., n-grams) and syntactic (e.g., grammatical production rules) features with all possible values (Burstein et al., 2003; Stab and Gurevych, 2014b). However, such large and sparse feature spaces can cause difficulty for feature selection. In our study (Nguyen and Litman, 2015), we propose an innovative algorithm that post-processes the output of an LDA topic model (Blei et al., 2003) to extract *argument words* (argument indicators, e.g. '*hypothesis*', '*reason*', '*think*') and *domain words* (specific terms commonly used within the topic's domain, e.g. '*bystander*', '*education*') which are used as novel features and constraints to improve the feature space. Particularly, we keep only argument words from unigram features and remove higher order n-gram features (e.g., bigrams, trigrams). Instead of production rules, we derive features from dependency parses which enables us to both retain syntactic structures and incorporate abstracted lexical constraints. Our lexicon extraction algorithm is semi-supervised in that we use manually-selected argument seed words to guide the process.

Different data-driven approaches have been proposed to identify aspects of argumentative language (e.g., organizational content vs. topical content), such as supervised sequence modeling (Madnani et al., 2012), probabilistic topic models (Séaghdha and Teufel, 2014; Du et al., 2014). Post-processing LDA (Blei et al., 2003) output was studied to identify topics of visual words (Louis and Nenkova, 2013) and representative words of topics (Brody and Elhadad, 2010; Funatsu et al., 2014). Our algorithm has a similarity with (Louis and Nenkova, 2013) in that we use seed words to guide the separation.

Our argument component identification model with novel features enabled by argument and domain lexicons is evaluated using the first persuasive essay corpus that we have introduced. Stab and Gurevych (2014b) were the first to utilize the corpus for developing an argument mining model for persuasive essays. Given a candidate argument component, the problem is to classify its argumentative label, i.e., Major Claim, Claim, Premise, or None. We re-implement the argument component classifier described in (Stab and Gurevych, 2014b) as a baseline to evaluate our approach in different experimental settings. We also follow experiments conducted in (Stab and Gurevych, 2014b) to directly compare our results with those reported in the prior study.

4.2 ARGUMENT AND DOMAIN WORD EXTRACTION

In this section we describe our algorithm to extract argument and domain words from a development dataset using predefined argument keywords (Nguyen and Litman, 2015). We recall that argument words are those playing a role of argument indicators and commonly used in different argument topics, e.g. '*reason*', '*opinion*', '*think*'. In contrast, domain words are specific terminologies commonly used within the topic, e.g. '*art*', '*education*'. Our notions of argument and domain languages share a similarity with the idea of shell language and content in (Madnani et al., 2012) in that we aim to model the lexical signals of argumentative content. However while Madnani et al. (2012) emphasized the boundaries between argument shell and content, we emphasize more the lexical signals themselves and allow argument words to occur in the argument content. For example, the MajorClaim in Figure 1 has two argument words '*should*' and '*instead*' which make the statement controversial.

The development data for the persuasive essay corpus are 6794 unlabeled essays (*Persuasive Set*) with titles collected from *www.essayforum.com*. We manually select 10 argument keywords/seeds that are the 10 most frequent words in the titles that seemed argument related: *agree, disagree, reason, support, advantage, disadvantage, think, conclusion, result, opinion*. We extract seeds of domain words as those in the titles but not argument keywords or stop words, and obtain 3077 domain seeds (with 136482 occurrences). Each domain seed

Topic 1 *reason exampl support agre think becaus disagre statement opinion believe therefor idea conclus ...*
Topic 2 *citi live big hous place area small apart town build communiti factori urban ...*
Topic 3 *children parent school educ teach kid adult grow childhood behavior taught ...*

Table 4: Samples of top argument words (topic 1), and top domain words (topics 2 and 3) extracted from persuasive development set. Words are stemmed.

is associated with an in-title occurrence frequency f.

All words in the development set including seed words are stemmed, and named entities are replaced with the corresponding NER labels by the Stanford parser. We run GibbsLDA++[1] implementation of LDA (Phan and Nguyen, 2007) on the development set, and assign each identified LDA topic three weights: domain weight (DW) is the sum of domain seed frequencies; argument weight (AW) is the number of argument keywords; and combined weight $CW = AW - DW$. Argument keywords are weighted more than domain seeds to reduce the size disparity of the two seed sets. For an example of these weights, topic 2 in the LDA's output of Persuasive Set in Table 4 has $AW = 5$ (five argument keywords not shown in the table are: *more, conclusion, advantage, who, which*), $DW = 0.15$, $CW = 4.85$. The in-title frequency of the stem *citi* is $f(citi) = 381/136482 = 0.0028$ given its 381 occurrences in the 136482 domain seed occurrences in the titles.

LDA topics are then ranked by CW with the top topic has highest CW value, and we calculate the ratio of CW of top-2 topics. We vary number of LDA topics k and select the k with the highest CW ratio ($k = 36$). The argument word list is the LDA topic with the largest combined weight given the best k. Domain words are the top words of other LDA topics but not argument or stop words.

[1] http://gibbslda.sourceforge.net

For the persuasive development set, our algorithm found $k = 36$ as the best number of LDA topics. Given 10 argument keywords, our algorithm returned a list of 263 argument words which is a mixture of keyword variants (e.g. *think, believe, viewpoint, opinion, argument, claim*), connectives (e.g. *therefore, however, despite*), and other stop words. The complete set of argument words extracted from the persuasive development set is presented in Appendix A.1. 1806 domain words are extracted by the algorithm. We note that domain seeds are not necessarily present in the extracted domain words partially because words with occurrence less than 3 are removed from LDA topics. On the other hand, the domain word list of Persuasive Set has 6% not in the domain seed set.

4.3 PREDICTION MODELS

4.3.1 Stab & Gurevych 2014

We first describe in detail the model developed by Stab and Gurevych (2014b) because many of the features proposed here are used in our model. The model in (Stab and Gurevych, 2014b) (referred to as Stab14 hereafter) uses the following features extracted from persuasive essays:

- *Structural features:* #tokens and #punctuations in argument component (AC), in covering sentence, and preceding/following the AC in sentence; token ratio between covering sentence and AC. Two binary features indicate if the token ratio is 1 and if the sentence ends with a question mark. Five position features are covering sentence's position in essay, whether the AC is in the first/last paragraph, the first/last sentence of a paragraph.
- *Lexical features:* all n-grams of length 1-3 extracted from the text span that include the AC and its preceding text which is not covered by other AC's in sentence; verbs like '*believe*'; adverbs like '*also*'; and whether the AC has a modal verb.
- *Syntactic features:* #sub-clauses and depth of syntactic parse tree of the covering sentence of the AC; tense of main verb and grammatical production rules (VP $\rightarrow$ VBG NP) from the sub-tree that represent the AC.

- *Discourse markers:* discourse connectives of 3 relations: Comparison, Contingency, and Expansion are extracted by the *addDiscourse* program (Pitler et al., 2009). A binary feature indicates if the corresponding discourse connective precedes the AC.
- *First person pronouns:* Five binary features indicate whether each of *I*, *me*, *my*, *mine*, and *myself* is present in the covering sentence. An additional binary feature indicates if one of five first person pronouns is present in the covering sentence.
- *Contextual features:* #tokens, #punctuations, #sub-clauses, and presence of modal verb in preceding and following sentences of the AC.

Their study assumes that gold-standard boundaries of argument components are available, and the main focus is predicting the argumentative labels of those components. To develop discourse marker features, the authors manually collected 55 Penn Discourse Treebank markers after removing those that do not indicate argumentative discourse, e.g. markers of Temporal relations. Because the list of 55 discourse markers was not publicly available, we used a program to extract discourse connectives.

4.3.2 Nguyen & Litman 2015

Our proposed model (referred to as Nguyen15) improves Stab14 by using extracted argument and domain words as novel features and constraints to replace its n-gram and production rule features (Nguyen and Litman, 2015). Compared to n-grams in *lexical aspect*, argument words are believed to provide a much more compact representation of the argument indicators. As for the *structural aspect*, instead of production rules, e.g. "$S \rightarrow NP\ VP$", we use dependency parses to extract pairs of subject and main verb of sentences, e.g. "*I.think*", "*view.be*". Dependency relations are minimal syntactic structures compared to production rules. To further make the features topic-independent, we keep only dependency pairs that do not include domain words.

In summary, our proposed model takes all features from the baseline except n-grams and production rules, and adds the following features: *argument words* as unigrams; *filtered dependency pairs* which are argumentative subject–verb pairs are used as skipped bigrams; and *numbers* of argument and domain words (see Figure 7). Our proposed model is compact

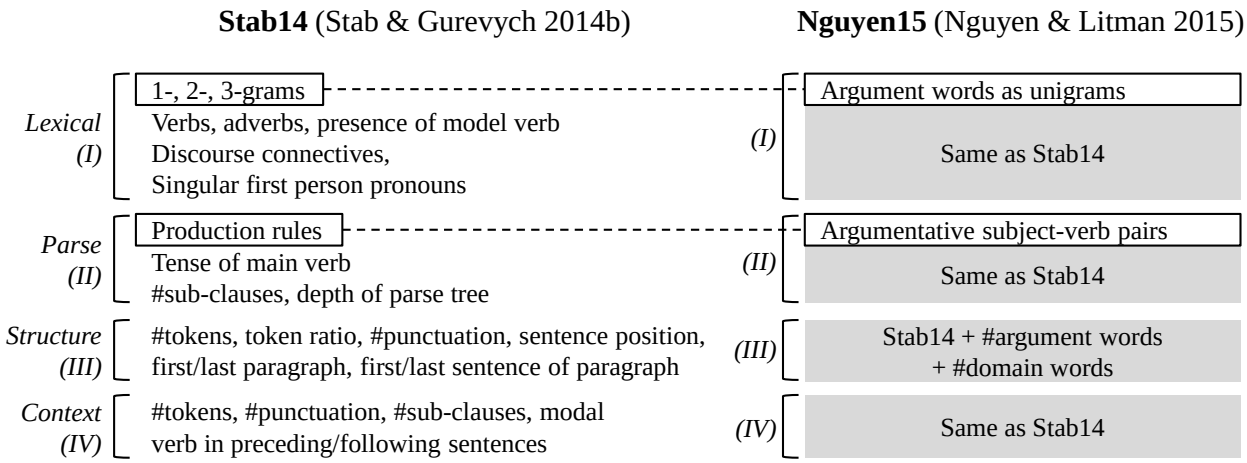

Figure 7: Feature illustration of Stab14 and Nguyen15. N-grams and production rules in Stab14 are replaced by argument words and argumentative subject–verb pairs in Nguyen15.

with 956 original features compared to more than 5000 features in our implementation of the baseline model. In fact, because our implementation removes n-grams with less than 3 occurrences, it should not have larger feature space than the original model in (Stab and Gurevych, 2014b).

4.4 EXPERIMENTAL RESULTS

4.4.1 Proposed vs. Baseline Models

This experiment replicates what was conducted in (Stab and Gurevych, 2014b). We perform 10-fold cross validations and report the average results. In each run models are trained using LibLINEAR (Fan et al., 2008) algorithm with top 100 features returned by the InfoGain feature selection algorithm performed in the training folds. We use LightSIDE[2] to extract

[2]http://ankara.lti.cs.cmu.edu/side

	Reported	Stab14	Nguyen15	Stab14	Nguyen15
#features	100	100	100	130	70
Accuracy	0.77	0.783	**0.794**+	0.803	**0.828***
Kappa	NA	0.626	**0.649***	0.640	**0.692***
Precision	**0.77**	0.760	0.756	0.763	**0.793**
Recall	0.68	0.687	**0.697**	0.680	**0.735**+

Table 5: Argument component classification performances with top 100 features (left) and best number of features (right). Corpus: Persuasive1.

n-grams and production rules, the Stanford Parser[3] (Klein and Manning, 2003) to parse the texts, and Weka[4] (Hall et al., 2009) to conduct the machine learning experiments.

Table 5 (left) shows the performances of three models: *Reported* and *Stab14* are respectively the reported performance and our implementation of Stab14, and *Nguyen15* is our proposed model. Because of the skewed label distribution, all reported precision and recall are un-weighted average values from by-class performances. In the table, symbols + and * indicate trending and significant difference ($p < 0.1$ and $p < 0.05$) in Stab14 vs. Nguyen15 comparison, respectively. Best values are highlighted in bold.

We note that there are performance disparities between Stab14 (our implementation), and reported performance (Stab and Gurevych, 2014b). The differences may mostly be due to dissimilar feature extraction methods and NLP/ML toolkits. Comparing Stab14 and Nguyen15 shows that our proposed model Nguyen15 yields higher Kappa (significantly) and accuracy (trending).

To further analyze performance improvement by Nguyen15 model, we use 75 randomly-selected essays to train and estimate the best numbers of features of Stab14 and Nguyen15 (w.r.t F1 score) through a 9-fold cross validation, then test on 15 remaining essays. As shown in Table 5 (right), Nguyen15's test performance is consistently better with far smaller number

[3]https://nlp.stanford.edu/software/lex-parser.shtml
[4]https://www.cs.waikato.ac.nz/ml/weka

of top features (70) than Stab14 (130). Nguyen15 has 6 of 31 argument words not present in Stab14's 34 unigrams: *analyze, controversial, could, debate, discuss, ordinal*. Nguyen15 keeps only 5 dependency pairs: *I.agree, I.believe, I.conclude, I.think* and *people.believe* while Stab14 keeps up to 31 bigrams and 13 trigrams in the top features. These indicate the dominance of our proposed features over generic n-grams and syntactic features.

4.4.2 Alternative Argument Word List

In this experiment, we study the prediction transfer of argument words when the development data to extract them is of a different genre than the test data. To create an alternative argument word list, we utilize 254 unannotated essays (*Academic Set*) with titles from Psychology classes in years 2011 and 2013 as the development data. We select 5 argument keywords which were specified in the writing assignments: *hypothesis, support, opposition, finding, study*. Filtering out argument keywords and stop words in essay titles of the academic set, we obtain 264 domain seeds (with 1588 occurrences), and their in-title occurrence frequency f.

With regard to this development set, the argument and domain word extraction algorithm returns 11 LDA topics, 315 (stemmed) argument words, and 1582 (stemmed) domain words. The learned argument words consist of keyword variants (e.g. *research, result, predict*), methodology terms (e.g. *effect, observe, variable, experiment, interact*), connectives (e.g. *also, however, therefor*), and other stop words. The set of learned domain words has 86% not in the domain seed set. Table 6 shows examples of top argument and domain words (stemmed) returned by the algorithm. The complete list of argument words extracted from the development set of academic writings is reported in Appendix A.2.

To build a model based on the alternative argument word list (referred to as AltAD), we replace the argument words in Nguyen15 with those 315 argument words, re-filter the dependency pairs and update the number of argument words. We follow the same setting in the experiment above to train Nguyen15 and AltAD using top 100 features. As shown in Table 7, AltAD performs worse than Nguyen15, with significantly lower accuracy and Kappa.

Topic 1 *studi research observ result hypothesi time find howev predict support expect oppos ...*
Topic 2 *respons stranger group greet confeder individu verbal social size peopl sneez ...*
Topic 3 *more gender women polit femal male men behavior differ prosoci express gratitud ...*

Table 6: Samples of top argument words (topic 1), and top domain words (topics 2 and 3) extracted from academic development set. Words are stemmed.

	AltAD	Nguyen15
Accuracy	0.770	**0.794***
Kappa	0.623	**0.649***
Precision	0.748	**0.756**
Recall	0.688	**0.697**

Table 7: Argument component classification performance with different argument word lists. Corpus: Persuasive1.

Comparing the two argument word lists gives interesting insights. The two lists have 119 common words with 9 discourse connectives (e.g. '*therefore*', '*although*'), 52 content words (e.g. '*result*', '*support*'), and 58 stop words. 28 of the common argument words appear in top 100 features of AltAD, but only 9 are content words (e.g., '*believe*', '*conclude*', '*example*', '*topic*', '*tendency*', '*conclusion*', '*instance*', '*analyze*', and '*final*'). This shows that while the two argument word lists have a fair number of words in common, the transferable part is mostly limited to function words, e.g. discourse connectives, stop words. In contrast, 188 of the 196 unique words to AltAD are not selected for top 100 features, and most of those are popular terms in academic writings, e.g. '*research*', '*hypothesis*', '*variable*'. Moreover,

Nguyen15's top 100 features have 12 argument words unique to the model, and 11 of those are content words, e.g. '*believe*', '*agree*', '*discuss*', '*view*'. These non-transferable parts suggest that argument words should be learned from appropriate seeds and development sets for best performance.

4.5 SUMMARY

Our proposed features are shown to efficiently replace generic n-grams and production rules in argument component classification tasks for significantly better performance. The core component of our feature extraction is a novel algorithm that post-processes LDA output to learn argument and domain words with a minimal seeding. These results support the first main hypothesis H1 (§1.2) about the effectiveness of topic-context features enabled by argument and domain word lexicons in argument component identification. Moreover, our analysis gives insights into the lexical signals of argumentative content. While argument word lists extracted for different data can have parts in common, there are non-transferable parts which are genre-dependent and necessary for the best performance.

5.0 IMPROVING ARGUMENT MINING IN STUDENT ESSAYS USING ARGUMENT INDICATORS AND ESSAY TOPICS

5.1 INTRODUCTION

Argument mining systems for student essays need to be able to reliably identify argument components independently of particular writing topics. Prior argument mining studies have explored linguistic indicators of argument such as pre-defined indicative phrases for argumentation (Mochales and Moens, 2008), syntactic structures, discourse markers, first person pronouns (Burstein et al., 2003; Stab and Gurevych, 2014b), and words and linguistic constructs that express rhetorical function (Séaghdha and Teufel, 2014). However only a few studies have attempted to abstract over the lexical items specific to argument topics for new features, e.g., common words with title (Teufel and Moens, 2002), cosine similarity with the topic (Levy et al., 2014), or to perform cross-topic evaluations (Burstein et al., 2003). In a classroom, students can have writing assignments in a wide range of topics, thus features that work well when trained and tested on different topics (i.e., writing-topic independent features) are more desirable.

Stab and Gurevych (2014b) studied the argument component identification problem in persuasive essays, and used linguistic features like ngrams and production rules (e.g., *VP→VBG NP*, *NN→sign*) in their argument mining system. While their features were effective, their feature space was large and sparse. Our prior work (Chapter 4) addressed that issue by replacing n-grams with a set of argument words learned in a semi-supervised manner, and using dependency rather than constituent-based parsers, which were then filtered based on the learned argument versus domain word distinctions (Nguyen and Litman, 2015). While our new features were derived from a semi-automatically learned lexicon of argument and

domain words, the role of using such a lexicon was not quantitatively evaluated. Moreover, neither (Stab and Gurevych, 2014b) nor we used features that abstracted over topic lexicons, nor performed cross-topic evaluation.

In this chapter, we present our new study that addresses the above limitations in four ways (Nguyen and Litman, 2016b). First, we run all of our studies using the first corpus of persuasive essays and the academic essay corpus (§3). Second, we present new features to model not only indicators of argument language but also to abstract over essay topics. Third, we build ablated models that do not use the extracted argument and domain words to derive new features and feature filters, so we can quantitatively evaluate the utility of extracting such word lists. Finally, in addition to 10-fold cross validation, we conduct cross-topic validation to evaluate model robustness when trained and tested on different writing topics.

Through experiments on two different corpora, we aim to provide support for the following three model-robustness hypotheses: *models enhanced with our new features will outperform baseline models* when evaluated using (h1) 10-fold cross validation and (h2) cross-topic validation; *our new models will demonstrate topic-robustness* in that (h3) their cross-topic and 10-fold cross validation performance levels will be comparable.

5.2 PREDICTION MODELS

5.2.1 Stab14

As described in §4.3.1, the Stab14 model was developed using the first version of the Persuasive Essay Corpus. Despite the differences between persuasive essays and academic essays, the Stab14 model is also applicable to the Academic Essay Corpus. First, the two corpora share certain similarities in writing styles and coding schemes. Both corpora consist of student writings whose content is developed to elaborate a main hypothesis for a persuasion purpose. Regarding coding schemes, MajorClaims in persuasive essays correspond to Hypothesis statements in academic essays, and Claims match Support and Opposition

findings. Premises in persuasive essays can be considered student writer's elaborations of previous studies in academic essays. Second, most of prediction features proposed in their study are generic, e.g., n-grams, grammatical production rules, and discourse connectives, which are expected to work for student writings in general. Therefore, we adapt the Stab14 model to the Academic Essay Corpus for a baseline model to evaluate our approach.

As the Academic Essay Corpus has annotation done at sentence-level and contains no information of argument component boundaries, all features of Stab14 that involve boundary information are not applicable to the Academic Essay Corpus. Therefore, the Stab14 model is adapted to the Academic Essay Corpus by simply extracting all features from the sentences, and removing features that require both argument component and covering sentence, e.g., token ratio.

5.2.2 Nguyen15v2

We implement two modified versions of the Nguyen15 model (§4.3.2) as the second baselines (referred to as Nguyen15v2), one for each corpus. Additional experiments with the Persuasive Essay Corpus showed that argument and domain word count features were not effective, so we decided to remove these two features from Nguyen15. For each version we re-implement the argument and domain word extraction algorithm to extract argument and domain words from a development dataset (§4.2).

5.2.3 Proposed Model

Our proposed model of this study, ADW4, is Nguyen15v2 (with the argument- and domain-word based features) expanded with 4 new feature sets extracted from the sentences of the associated argument components, i.e., covering sentences. A summary of features used in this model is given in Figure 8. To model the topic cohesion of essays, we include two contextual features that count words in common:

1. *Numbers of common words* of the given sentence with the preceding one and with the essay title.

We also proposed new lexical features for better indicators of argument language. We observe that in argumentative essays students usually use comparison language to compare and contrast ideas. However not all comparison words are independent of the essay topics. For example, while adverbs (e.g., '*more*') are commonly used across essays, adjectives (e.g., '*cheaper*', '*richer*') seem specific to the particular topics. Thus, we introduce the following comparison features:

2. *Comparison words*: comparative and superlative adverbs. *Comparison POS*: two binary features indicating the presences of *RBR* and *RBS* part-of-speech tags.

We also see that student authors may use plural first person pronouns (*we*, *us*, *our*, *ours*, and *ourselves*) as a rhetorical device to make their statement sound more objective/persuasive, for instance "*we always find that we need the cooperation.*" We supplement the first person pronoun set in the baseline models with 5 plural first person pronouns:

3. Five binary features indicating whether each of 5 *plural first person pronouns* is present.

We notice that many discourse connectives used in baseline models are duplicates of our extracted argument words, e.g., '*however*'. Thus using both argument words and discourse connectives may inefficiently enlarge the feature space. To emphasize the discourse information, we include discourse relations as identified by the addDiscourse program (Pitler et al., 2009) as new features:

4. Three binary features showing if each of *Comparison*, *Contingency*, *Expansion* discourse relations is present. Stab and Gurevych (2014b) did not use temporal discourse relations so we ignored those relations in this study.

5.2.4 Ablated models

We propose two simple alternatives to ADw4 to examine the role of argument and domain word lists in our argument mining task:

• **woAD**: we disable the argument/domain-word based features and constraints in ADw4 so that woAD does not include argument words, but uses all possible subject–verb pairs. All other features of ADw4 are unaffectedly applied to woAD. Comparing woAD to ADw4

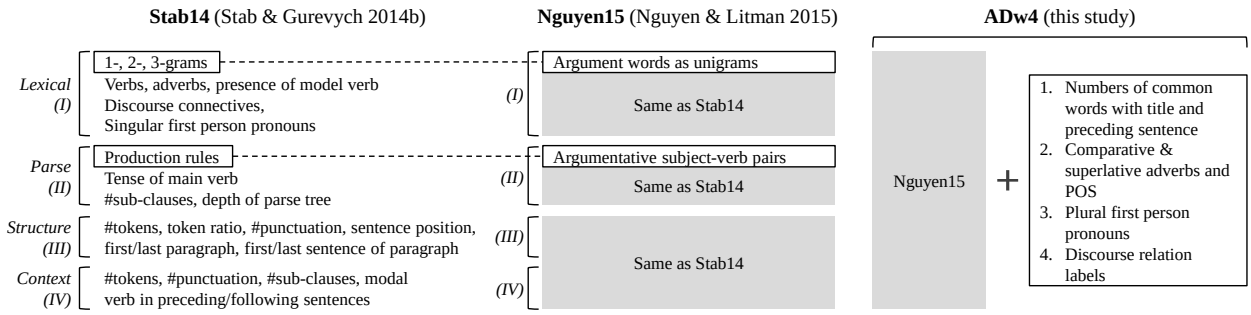

Figure 8: Feature illustration of Stab14, Nguyen15v2 and ADw4. 1-, 2-, 3-grams and production rules in Stab14 are replaced by argument words and argumentative subject–verb pairs in Nguyen15v2. ADw4 extends Nguyen15v2 with 4 new feature sets.

will show the contribution of the extracted argument and domain words to the model performance.

- **Seed**: extracted argument and domain word lists are replaced with only the seeds that were used to start the semi-supervised argument and domain word learning process (see next section). Comparing Seed to ADw4 will show whether it is necessary to use the semi-supervised approach for expanding the seeds to construct the larger/more comprehensive argument and domain word lexicons.

5.3 EXPERIMENTAL RESULTS

5.3.1 10-fold Cross Validation

We first conduct 10-fold cross validations to evaluate our proposed model and the baseline models. All models are trained using the SMO (as in (Stab and Gurevych, 2014b)) implementation of SVM in Weka (Hall et al., 2009). LightSIDE[1] and Stanford Parser (Klein and

[1]http://ankara.lti.cs.cmu.edu/side

Manning, 2003) are used to extract n-grams, parse trees and named entities. We follow (Stab and Gurevych, 2014b) and use top 100 features ranked by InfoGain algorithm on training folds to train the models.

To obtain enough samples for a significance test when comparing model performance in 10-fold cross validation to cross-topic validation, we perform 10 runs of 10-fold cross validations (10×10 cross-validation) and report the average results over 10 runs. From our prior study, and additional experiments, we also noticed that the skewed distributions and small sizes of our corpora make stratified 10-fold cross validation performance notably affected by the random seeds. Thus, we decided to conduct multiple cross validations in this experiment to reduce any effect of random folding. We use T-tests to compare performance of models given that each model evaluation returns 10 samples of 10-fold cross validation performance.

As the two corpora are very class-skewed, we report unweighted precision and recall. Also while accuracy is a common metric, kappa is a more meaningful value given our imbalanced data. Model performances are reported in Table 8. Best values are highlighted in bold. Symbols + and * indicate trending and significant difference ($p < 0.1$ and $p < 0.05$) by T-test when comparing with ADw4, respectively.

Our first analysis is about the performance improvement of our proposed model over the two baselines. We see that our model ADw4 significantly outperforms Stab14 in all reported metrics across both corpora. However comparing ADw4 and Nguyen15v2 reveals inconsistent patterns. While ADw4 yields a significantly higher performances than Nguyen15v2 when evaluated in the persuasive corpus, our proposed model performs worse than that baseline in the academic corpus. Looking at individual metrics of these two models we see that Nguyen15v2 has trending higher accuracy ($p = 0.05$) and also trending higher precision ($p = 0.09$) than ADw4 in academic corpus. The differences on kappa and recall between the two models are not significant. These results partially support our first model-robustness hypothesis (h1) in that our proposed features improve over both baselines using 10-fold cross validation in the persuasive corpus only.

We now turn to our feature ablation results. Removing the argument/domain-word based features from ADw4, we see that woAD's performance figures are all significantly worse than

	Persuasive Essay Corpus				
Metric	Stab14	Nguyen15v2	woAD	Seed	ADw4
Accuracy	0.787*	0.792*	0.780*	0.781*	**0.805**
Kappa	0.639*	0.649*	0.629*	0.632*	**0.673**
Precision	0.741*	0.745*	0.746*	0.740*	**0.763**
Recall	0.694*	0.698*	0.695*	0.695*	**0.720**
	Academic Essay Corpus				
Metric	Stab14	Nguyen15v2	woAD	Seed	ADw4
Accuracy	0.934*	**0.942**+	0.933*	0.935*	0.941
Kappa	0.558*	**0.635**	0.528*	0.564*	0.629
Precision	0.804*	**0.830**+	0.829	0.826	0.825
Recall	0.628*	**0.695**	0.594*	0.637*	**0.695**

Table 8: Argument component classification performance. Corpora: Persuasive1, Academic.

ADw4 except for precision in the academic corpus. Furthermore, we find that argument keywords and domain seeds are poor substitutes for the full argument and domain word lists learned from these seeds. This is shown by the significantly lower performances of Seed compared to ADw4, except for precision in the academic corpus. Nonetheless, adding the features computed from just argument keywords and domain seeds still helps Seed perform better than woAD (with higher accuracy, kappa and recall in both persuasive and academic corpora).

5.3.2 Cross-topic Validation

To better evaluate the models when predicting essays of unseen topics we conduct cross-topic validations where training and testing essays are from different topics (Burstein et al., 2003). We examined 90 persuasive essays and categorized them into 12 groups including 11 single-topic groups, each corresponding to a major topic (groups have 4 to 11 essays). The

	Persuasive Essay Corpus				
Metric	Stab14	Nguyen15v2	woAD	Seed	ADw4
Accuracy	0.780*	0.796	0.774*	0.776*	**0.807**
Kappa	0.623*	0.654+	0.618*	0.623*	**0.675**
Precision	0.722*	0.757*	0.751	0.734	**0.771**
Recall	0.670*	0.695*	0.681*	0.686*	**0.722**
	Academic Essay Corpus				
Metric	Stab14	Nguyen15v2	woAD	Seed	ADw4
Accuracy	0.928*	0.939+	0.931*	0.935*	**0.944**
Kappa	0.491*	0.598+	0.474*	0.547*	**0.630**
Precision	0.768	0.832	**0.866**	0.839*	0.851
Recall	0.565*	0.664	0.551*	0.617*	**0.686**

Table 9: Argument component classification with cross topic performance. Corpora: Persuasive1, Academic

twelfth group (*Other*) is a mixture of 17 essays of minor topics (each has less than 3 essays), e.g., 3 essays about Languages, 2 essays about Prepared Food.

> *Technologies* (11 essays), *National Issues* (10), *School* (8), *Policies* (7), *Advertisement* (6), *International Relations* (6), *Learning* (6), *Art* (5), *Gender* (5), *Animal* (5), *Living Abroad* (4), *Other* (17).

We manually split 115 academic essays into 5 topics accordingly to the studied variables.

- *Attractiveness* as a function of clothing color (20 essays)
- *Email-response rate* as a function of recipient size (22)
- *Helping-behavior* with effects of gender and group size (31)
- *Politeness* as a function of gender (23)
- *Self-description* and word choices with influences of gender and self-esteem (19)

Again all models are trained using the top 100 features selected in training folds. In each folding, we use essays of one topic for evaluation and all other essays to train the model.

T-test is used to compare each of two sets of by-fold performances.

We first evaluate the performance improvement of our model compared to the baselines. As shown in Table 9, ADw4 again yields higher performance than Stab14 in all metrics of both corpora, and the improvements are significant except for precision in the academic essays. Moreover we generally observe a larger performance gap between ADw4 and Stab14 in cross-topic validation than in 10-fold cross validation. More importantly, with cross-topic validation, ADw4 now yields better performance than Nguyen15v2 for all metrics in both persuasive and academic corpora. Especially, our proposed model now even has trending higher accuracy and kappa than Nguyen15v2 in academic corpus. This shows a clear contribution of our new features in the overall performance, and supports our second model-robustness hypothesis (h2) that *our new features improve the cross-topic performance in both corpora compared to the baselines.*

With respect to feature ablation results, our findings are consistent with the prior cross-fold results in that woAD and Seed both have lower performance (often significantly) than ADw4 (with one exception). Seed again generally outperforms woAD, indicating that deriving features from even impoverished argument and domain word lists is better than not using such lexicons at all.

Next, we compare ADw4 performance across the cross-fold and cross-topic experimental settings (using a T-test to compare the mean of 10 samples of 10-fold cross validation performance versus the mean of cross-topic validation performance). In both corpora we see that ADw4 yields higher performance for all metrics in cross-topic versus 10-fold cross validation, except for recall in the academic corpus. Of these cross-topic performance figures, ADw4 has significantly higher precision and trending higher accuracy in the persuasive corpus. In academic corpus, ADw4's cross-topic accuracy, precision and recall are all significantly better than the corresponding figures for 10-fold cross validation. These results support strongly our third model-robustness hypothesis (h3) that *our proposed model's cross-topic performance is as high as 10-fold cross validation performance.*

In contrast, Nguyen15v2's performance difference between cross-topic and random-folding validations does not hold a consistent direction. Stab14 returns significantly higher results in 10-fold cross validation than cross-topic validation in both persuasive and academic corpora.

Also woAD and Seed's cross-topic performances are largely worse than those of 10-fold cross validation. Overall, the cross-topic validation shows the ability of our proposed model to perform reliably when the testing essays are from new topics, and the essential contribution of our new features to this high performance.

To conclude this section, we give a qualitative analysis of the top features selected in our proposed model. In each folding we record the top 100 features with associated ranks. By the end of cross-topic validation, we have a pool of top features (≈200 for each corpus), with an average rank for each. First we see that the proportion of argument words is about 49% of pooled features in both corpora, and the proportion of argumentative subject–verb pairs varies from 8% (in persuasive corpus) to 15% (in academic corpus). The new features introduced in ADw4 that are present in the top features include: two common word counts; *RBR* part-of-speech; person pronouns *We* and *Our*; discourse labels *Comparison, Expansion, Contingency*. All of those are in the top 50 except that *Comparison* label has average rank 79 in the persuasive corpus. This shows the utility of our new feature sets. Especially the effectiveness of common word counts encourages us to study advanced topic cohesion features in future work.

5.3.3 Performance on Held-out Test Sets

The experiments above used 10×10-fold cross-validation and cross-topic validation to investigate the robustness of prediction features. Note that this required us to re-implement both baselines as neither had previously been evaluated using cross-topic validation.[2] However, since both baselines were evaluated on single held-out test sets of the Persuasive Essay Corpora, that were available to us, our last experiment compares ADw4's performance with the best *reported* results for the original baseline implementations using their exact same training/test set splits (Stab and Gurevych, 2014b; Nguyen and Litman, 2015). That is, we train ADw4 using SMO classifier with top 100 features with the two training sets of 72 essays (Stab and Gurevych, 2014b) and 75 essays (Nguyen and Litman, 2015), and report the corresponding held-out test performances in Table 10.

[2]While Nguyen15v2 (but not Stab14) had been evaluated using 10-fold cross-validation, the random fold data cannot be replicated.

	Stab's test set		Nguyen's test set		
Metric	Stab best	Our SMO	Nguyen best	Our SMO	Our Lib-LINEAR
Accuracy	0.77	**0.816**	0.828	0.819	**0.837**
Kappa	–	**0.682**	0.692	0.679	**0.708**
Precision	0.77	**0.794**	0.793	0.762	**0.811**
Recall	0.68	**0.726**	0.735	0.703	**0.755**

Table 10: Argument component classification performance on held-out test sets. Corpora: Persuasive1, Academic.

While test performance of our model is higher than (Stab and Gurevych, 2014b), our model has worse test results than (Nguyen and Litman, 2015). This is reasonable as our model was trained following the same configuration as in (Stab and Gurevych, 2014b), but was not optimized as in (Nguyen and Litman, 2015). In fact, (Nguyen and Litman, 2015) obtained their best performing model using LibLINEAR classifier with top 70 features. If we keep our top 100 features but replace SMO with LibLINEAR, then ADw4 gains performance improvement with accuracy 0.84 and Kappa 0.71. With respect to the cross validations, while our chosen setting is in favor of Stab14, it still offers an acceptable evaluation as it is not the best configuration for either Nguyen15v2 or ADw4. Therefore, the conclusions from our new cross fold/topic experiments also hold when ADw4 is directly compared with published baseline test set results.

5.4 SUMMARY

Motivated by practical argument mining for student essays (where essays may be written in response to different assignments), we have presented new features that model argument indicators and abstract over essay topics, and introduced a new corpus of academic essays

to better evaluate the robustness of our models. Our proposed model in this study shows robustness in that it yields performance improvement with both *cross-topic* and *10-fold cross* validations for different types of student essays, i.e., *academic* and *persuasive*. Moreover, our model's cross-topic performance is even higher than cross-fold performances for almost all metrics.

Experimental results also show that while our model makes use of effective baseline features that are derived from extracted argument and domain words, the high performance of our model, especially in cross-topic validation, is also due to our new features which are generic and independent of essay topics. That is, to achieve the best performance, the new features are a necessary supplement to the learned and noisy argument and domain words.

6.0 EXTRACTING CONTEXTUAL INFORMATION FOR ARGUMENTATIVE RELATION CLASSIFICATION

6.1 INTRODUCTION

Given a pair of arguments or argument components with one referred to as the source and the other as the target, argumentative relation mining involves determining whether a relation holds from the source to the target, and classifying the argumentative function of the relation, e.g., support vs. attack. While some sort of heuristics may be useful to pre-determine source and target components, e.g., relative positions of the components, the general form of the argumentative relation mining problem considers two ordered pairs for each two argument components, i.e., each component is considered as the source source and target in turn. Argumentative relation mining – beyond argument component mining – is perceived as an essential steps towards more fully identifying the argumentative structure of a text (Peldszus and Stede, 2013; Sergeant, 2013; Stab and Gurevych, 2014b). Consider the second paragraph shown in Figure 9. Only detecting the argument components (a claim in sentence 2 and two premises in sentences 3 and 4) does not give a complete picture of the argumentation. By looking for relations between these components, one can also see that the two premises together justify the claim. The argumentation structure of the text in Figure 9 is illustrated in Figure 10 according to the annotation provided in the first corpus of persuasive essays.

Research on classifying argumentative relations between pairs of arguments or argument components has proposed a variety of features ranging from the superficial level, e.g., word pair, relative position, to the semantic level, e.g., semantic textual similarity, textual entailment. Cabrio and Villata (2012); Boltužić and Šnajder (2014) studied online debate corpora

Essay 73: Is image more powerful than the written word?

... [1]Hence, [*I agree only to certain degree that in today's world, image serves as a more effective means of communication*]$_{MajorClaim}$.

... [2][*pictures can influence the way people think*]$_{Claim}$. [3]For example, [*nowadays horrendous images are displayed on the cigarette boxes to illustrate the consequences of smoking*]$_{Premise}$. [4]As a result, [*statistics show a slight reduction in the number of smokers, indicating that they realize the effects of the negative habit*]$_{Premise}$...

Figure 9: Excerpt from a student persuasive essay. Sentences are numbered and argument components are tagged.

and aimed at identifying whether user comments support or attack the debate topic. They proposed to use content-rich features including semantic similarity and textual entailment. In principle, they expect the comment text (which is usually longer) to entail the topic phrase (which is usually shorter). Boltužić and Šnajder (2014) calculated semantic similarity between each comment sentence and the topic phrase, and returned the max and mean of sentence-level similarity scores. Despite the fact that user comments are usually long with multiple sentences, both Cabrio and Villata (2012) and Boltužić and Šnajder (2014) did not consider the discourse structure of the comment as auxiliary information to support the prediction. It has been proposed in (Biran and Rambow, 2011) that justifications (e.g., user comment) usually contain discourse structures that characterize argumentation. However, their study made use of only discourse indicators but not the discourse relations. We believe that identifying the discourse structures of justification will give insights to argumentation patterns used by writers to show their stances towards the argument topic.

To illustrate our idea, consider the following excerpt from a persuasive essay in the first corpus:

Essay 26: Prepared food

[1]In addition, cooking is one of arts humans create. [2]The more cooked food we chosen, the more cooking skills we lose. [3]At the increasing living pace, the majority of people tend to choose microwave as their unique cooker that help them prepare a dish in five minutes. [4]But rare people have been aware that this has contributed to a modification of cooking habits, which may cause the loss of our custom and culture about cooking.

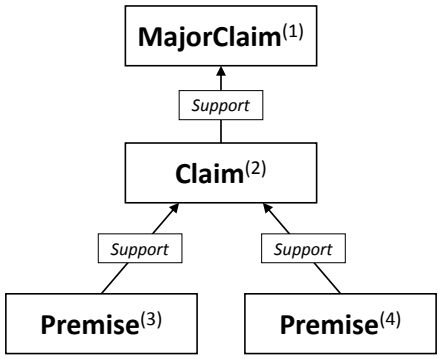

Figure 10: Structure of the argumentation in the excerpt in Figure 9. Premises 3 and 4 were annotated for separate relations to Claim 2. Our visualization should not mislead that the two premises are linked or convergent.

(5)In conclusion, although the invention of prepared foods definitely satisfies the demand of some people who are busy in their work, it is not a good thing.

The excerpt consists of a justification in sentences {1, 2, 3, 4} which supports a claim in sentence 5. Analyzing the discourse structure of the justification, we can see that the writer wanted to prove that "*losing cooking skills*" is a bad thing, which causes "*losing custom and culture*", which consequently shows a stance against the "*prepared foods*".

Another example can be taken from Figure 9. Without knowing the content "*horrendous images are displayed on the cigarette boxes*" in sentence 3, one cannot easily tell that "*reduction in the number of smokers*" in sentence 4 supports the "*pictures can influence*" claim in sentence 2. We expect that such content relatedness can be revealed from a discourse analysis, e.g., the appearance of a discourse connective "*As a result*".

Differently from (Cabrio and Villata, 2012; Boltužić and Šnajder, 2014), Stab and Gurevych (2014b) aimed at classifying the argumentative relations (i.e., support vs. non-support) between argument components. An argument component in (Stab and Gurevych, 2014b) is a sentence or a clause so it is less content-rich than user comments in (Cabrio and Villata, 2012; Boltužić and Šnajder, 2014). Stab and Gurevych (2014b) proposed a diverse feature set including features involving information from both components of the pair. e.g., word

pairs, common words, relative positions. However, a limitation of their model is the lack of contextual information as mentioned in their paper. For example, it is hard to determine the support relation between these two argument components: "*It helps relieve tension and stress*" and "*Exercising improves self-esteem and confidence*" without knowing that "*it*" refers to "*Exercising*". Although anaphora resolution may help in this case, other situations could require topic inference to determine the relatedness between texts. While topic information in many writing genres (e.g., scientific publications, Wikipedia articles, student essays) has been used to create features for argument component mining (Teufel and Moens, 2002; Levy et al., 2014; Nguyen and Litman, 2015), topic-based features have been less explored for argumentative relation mining. In the excerpts below, knowing that '*technology*' and '*weapons*' in essay 8, and '*online game*' and '*computer*' in essay 24 are topically related might help a model decide support relations between sentences.

Essay 8: Technology cannot solve all the world's problems
$^{(1)}$...[*there are some serious problems springing from modern technology*]$_{Claim}$. $^{(2)}$First, [*deadly and powerful weapons can be a huge threat to the world's peace*]$_{Premise}$.

Essay 24: Computer has negative effects to children
$^{(1)}$[*People who are addicted to games, especially online games, can eventually bear dangerous consequences*]$_{Claim}$. $^{(2)}$Although [*it is undeniable that computer is a crucial part of human life*]$_{Premise}$, [*it still has its bad side*]$_{MajorClaim}$.

Motivated by the discussion above, we propose *context-aware argumentative relation mining* – a novel approach that makes use of contextual features that are extracted by exploiting context sentence windows and writing topic to improve relation prediction.

6.2 CONTEXT-AWARE ARGUMENTATIVE RELATION MINING

Given these issues of existing work on argumentative relation classification, we propose a general framework that exploits contextual information to tackle the problems. First we derive features from *argument and domain word lexicons* which were automatically created by post-processing an essay's topic model. Besides using argument words as unigrams, we

also pair domain words that have the same or different LDA topic between source and target components.

Second, instead of considering argument components isolatedly as in (Stab and Gurevych, 2014b), our approach puts each argument component in its context window (Definition 1) to enrich the justification and enable contextual features. In particular, we derive features from *discourse relations* between argument components and windows of their surrounding sentences. We consider two discourse structure frameworks which are Penn Discourse Treebank (Prasad et al., 2008), and Rhetorical Structure Theory (Carlson et al., 2001) and use available toolkits for discourse relation extraction. Below we describe in detail the model developed by Stab and Gurevych (2014b) and how we improve it by our proposed contextual features for argumentative relation mining.

6.2.1 Baseline

We adapt Stab and Gurevych (2014b) to use as a baseline for evaluating our approach. Given a pair of argument components, we follow Stab and Gurevych (2014b) by first extracting 5 feature sets:

- *Structural features*: numbers of tokens and punctuations in source and target components, the absolute difference in numbers of tokens and punctuations between source and target. Positions of covering sentences of source and target components, sentence distance between source and target, whether source and target components are of the same sentence. Whether source and target components are in first or last sentences of a paragraph, whether target component occurs before source component.
- *Lexical features*: pairs of words from source and target components. The first word of argument component, pair of first words from source and target components. Whether source and target components contain modal verb, number of terms in common between two components.
- *Syntactic features*: grammatical production rules (e.g., $S \rightarrow NP, VP$) extracted from source and target components

- *Indicators*: whether source and target components start with a discourse connective from a set of 55 discourse connectives.
- *Predicted type*: the argumentative labels (e.g., Major Claim, Claim, Premise) of source and target components, which were identified by an argument component model.

We further improve the baseline model with additional features that were found helpful in prior studies.

- *Structural features*: Because a sentence may have more than one argument component, the relative component positions might provide useful information (Peldszus, 2014). We include 8 new component position features: whether the source and target components are the whole sentences or the beginning/end components of the sentences; whether the source is before or after the target component; and the absolute difference of their positions.
- *Indicators*: We expand discourse connective set by combining them with a 298-discourse marker set developed in Biran and Rambow (2011). We expect the expanded set of discourse connectives will represent better possible discourse relations in the texts.
- *Predicted type*: we use predicted labels returned by our argument component model which was shown to significantly outperform the corresponding model of Stab and Gurevych (Nguyen and Litman, 2016b).

For later presentation purposes, we name the set of all features from this section *except word pairs and production rules* as the *common features*. While word pairs and grammatical production rules were the most predictive features in (Stab and Gurevych, 2014b), we hypothesize that this large and sparse feature space may have a negative impact on model robustness (Nguyen and Litman, 2015). Most of our proposed models replace word pairs and production rules with different combinations of new contextual features.

6.2.2 Topic-context Model

Our first proposed model (TOPIC) makes use of topic-context features derived from the lexicon of argument and domain words for persuasive essays (Chapter 4). Using the lexicon,

we extract the following Topic-context features:

- *Argument word*: from all word pairs extracted from the source and target components, we remove those that have at least one word not in the argument word list. Each argument word pair defines a boolean feature indicating its presence in the argument component pair. We also include each argument word of the source and target components as a boolean feature which is true if the word is present in the corresponding component. We count number of common argument words, the absolute difference in number of argument words between source and target components.
- *Domain word count*: to measure the topic similarity between the source and target components, we calculate number of common domain words, number of pairs of two domain words that share an LDA topic, number of pairs that share no LDA topic, and the absolute difference in number of domain words between the two components.
- *Non-domain MainVerb-Subject dependency*: we extract MainVerb-Subject dependency triples, e.g., *nsubj(belive, I)*, from the source and target components, and filter out triples that involve domain words. In this case, the domain word lexicon is used as contextual constraints to keep our dependency features domain-independent. We model each extracted triple as a boolean feature which is true if the corresponding argument component has the triple.

Finally, we include the common feature set. To illustrate the topic-context features, consider the following source and target components. Argument words are in boldface, and domain words are in italic.

> **Essay 54**: museum and art gallery will disappear soon?
> Source: [**more** and **more people can** *watch exhibitions* through *television* **or** *internet* at *home* **due to** *modern technology*]$_{Premise}$
> Target: [**some people think** *museums* and *art galleries* **will** *disappear soon*]$_{Claim}$

An argument word pair is **people-think**. There are 35 pairs of domain words. A pair of two domain words that share an LDA topic is *exhibitions-art*. A pair of two domain words that do not share any LDA topic is *internet-galleries*.

6.2.3 Window-context Model

Our second proposed model (WINDOW) extracts features from discourse relations and common words between context sentences in the *context windows* (Definition 1) of the source and target components.

In this study, context windows are determined using *window-size* heuristics. Given a half-size n, we form a context window by grouping the covering sentence with at most n adjacently preceding and n adjacently following sentences that must be in the same paragraph. Thus, the context window has the size $2n$. To minimize noise in feature space, we require that context windows of the source and target components must be mutually exclusive. Biran and Rambow (2011) observed that the relation between a source argument and a target argument is usually instantiated by some elaboration/justification provided in a support of the source argument. Therefore we prioritize the context window of source component when it overlaps with the target context window. Particularly, we keep overlapping context sentences in the source window, and remove them from the target window. Due to the paragraph constraint and window overlapping as mentioned, half-size does not infer the actual context-window size. However, half-size infers the maximum size that a window can have.

For example, with half-size 1, context windows of the Claim in sentence 2 and the Premise2 in sentence 4 in Figure 11 overlap at sentence 3. When the Premise2 is set as a source component, its context window includes sentences {3, 4}, and the Claim as a target has context window with only sentence 2.

We extract three window-context feature sets from the context windows to use with the common feature set.

- *Common word*: as common word counts between adjacent sentences were shown useful for argument mining (Nguyen and Litman, 2016b), we count common words between the covering sentence with preceding context sentences, and with following context sentences, for source and target components.
- *Discourse relation*: for both source and target components, we extract discourse relations between context sentences, and within the covering sentence. We also extract discourse

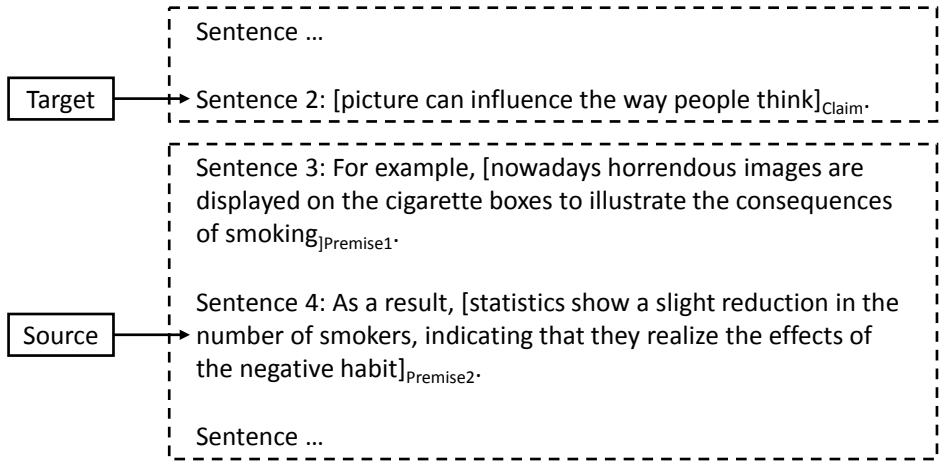

Figure 11: Context-windows for argument components in Figure 9 when sentence 4 is the source and sentence 2 is the target components.

relations between each pair of source context sentence and target context sentence. Each relation defines a boolean feature. We extract both Penn Discourse Treebank (PDTB) relations and Rhetorical Structure Theory Discourse Treebank (RST-DTB) relations using publicly available discourse parsers (Ji and Eisenstein, 2014; Wang and Lan, 2015).

Each PDTB relation has sense label defined in 3 layers (class, type, subtype), e.g., *CONTINGENCY.Cause.result*. While there are only four semantic class labels at the class-level which may not cover well different aspects of argumentative relation, subtype-level output is not available given the discourse parser we use. Thus, we use relations at type-level as features.

For RST-DTB relations, we use only relation labels, but ignore the nucleus and satellite labels of components as they do not provide more information given the component order in the pair. Because temporal relations were shown not helpful for argument mining tasks, we exclude them here (Biran and Rambow, 2011; Stab and Gurevych, 2014b).

- *Discourse marker*: while the baseline model only considers discourse markers within the argument components, we define a boolean feature for each discourse marker classifying whether the marker is present before the covering sentence of the source and target com-

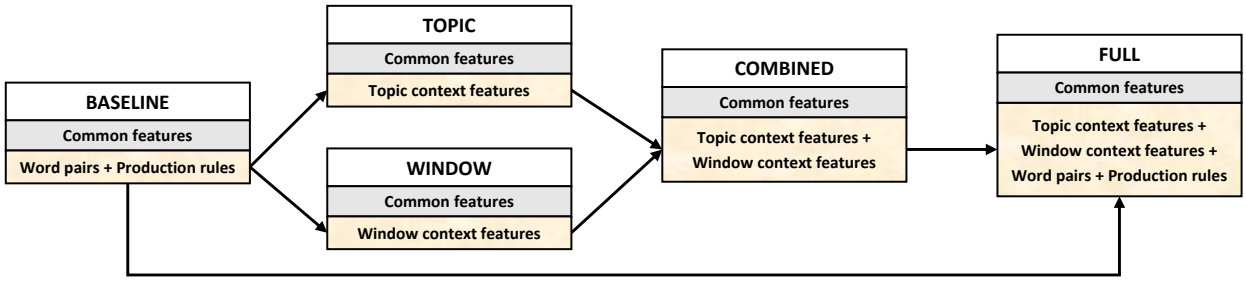

Figure 12: Features used in the baseline and our proposed models for argumentative relation mining. Feature change across models are denoted by connectors.

ponents or not. This implementation aims to characterize the discourse of the preceding and following text segments of each argument component separately.

6.2.4 Combined Model

While window-context features are extracted from surrounding text of the argument components, which exploits the local context, the topic-context features are an abstraction of topic-dependent information, e.g., domain words are defined within the context of topic domain (Nguyen and Litman, 2015), and thus make use of the global context of the topic domain. We believe that local and global context information represent complementary aspects of the relation between argument components. Thus, we expect to achieve the best performance by combining Window-context and Topic-context models.

6.2.5 Full Model

Finally, the Full model includes all features in Baseline and Combined models. That is, the Full model is the Combined model plus word pairs and production rules. A summary of all models is shown in Figure 12.

Label	#pairs
Within-paragraph constraint	
Support	989
Attack	103
No paragraph constraint	
Support	1312
Attack	161

Table 11: Argumentative relations with different constraints in corpus Persuasive1.

6.3 ARGUMENTATIVE RELATION TASKS

6.3.1 Task 1: Support vs. Non-support

We utilize the first corpus of persuasive essays to demonstrate our context-aware argumentative relation mining approaches. Our first task follows (Stab and Gurevych, 2014b): given a pair of source and target argument components, identify whether the source argumentatively supports the target or not. When a support relation does not hold, the source may attack or have no relation with the target component. For each of two argument components in the same paragraph, we form two pairs (i.e., reversing source and target). In total we obtain 6330 pairs in 90 essays, in which 989 (15.6%) have Support relation. Among 5341 Non-support pairs, 103 have Attack relation and 5238 are no-relation pairs (Table 11). Stab and Gurevych (2014b) split the corpus into an 80% training set and a 20% test set which have similar label distributions. We use this split to train and test our proposed models, and directly compare our models' performance to their reported results.

6.3.1.1 Tuning Half-size Parameter Because our WINDOW model uses a half-size parameter to form context windows of the source and target argument components, we investigate how the half-size of context window impacts the prediction performance of the

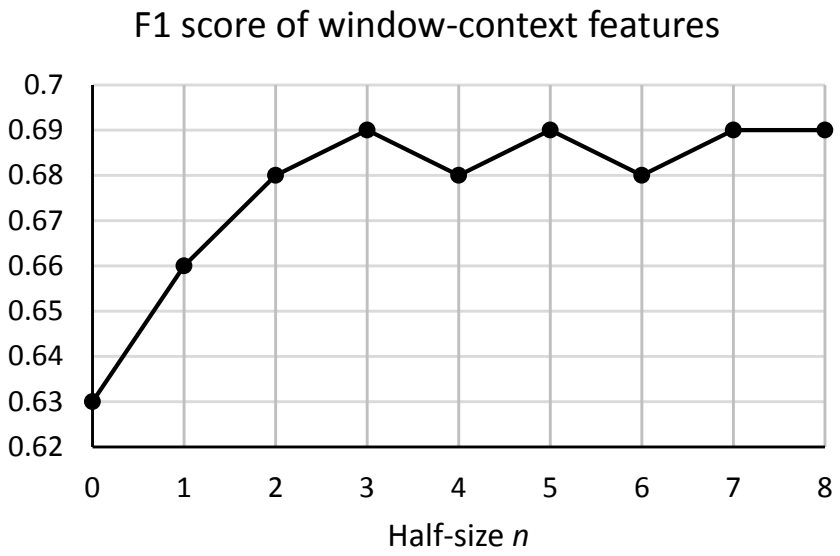

Figure 13: Performance of window-context features by half-size n. Corpus: Persuasive1.

window-context features. We set up a model with only window-context features (i.e., WINDOW model without common features) and determine the window-size in range [0, 8] that yields the best F1 score in 10-fold cross validation. Half-size 0 means covering sentence is the only context sentence. We experimented with not using context sentence at all and obtained worse performance. Our data does not have context window with half-size 9 or larger.[1]

We use the training set as determined in Stab and Gurevych (2014b) to cross-validate the model using LibLINEAR algorithm (Fan et al., 2008) without parameter or feature optimization. Cross-validations are conducted using Weka (Hall et al., 2009). We use Stanford Parser to perform text processing (Klein and Manning, 2003).

As shown in Figure 13, while increasing the half-size from 2 to 3 improves F1 score (significantly), using half-sizes greater than 3 does not gain further improvement. We hypothesize that after a certain limit, larger context windows will produce more noise than helpful information for the prediction. Therefore, we set the half-size to 3 in all of our experiments involving window-context features (all with a separate test set).

[1]Counting the whole corpus, the maximal paragraph has 10 sentences – see Table 16.

6.3.1.2 Performance on Test Set We train all models described above using the training set and report their performances on the test set in Table 12. Best values are highlighted in bold. Values smaller than baseline are underlined. Symbol * indicates significantly different from the baseline ($p < 0.05$). The learning algorithm with parameters are kept the same as in the window-size tuning experiment. Given the skewed class distribution of this data, Accuracy and F1 of Non-support (the major class) are less important than Kappa, F1, and F1 of Support (the minor class). To conduct T-tests for performance significance, we split the test data into subsets by essays' ID, and record prediction performance for individual essays. We also compare our baseline to the reported performance (REPORTED) for Support vs. Non-support classification in (Stab and Gurevych, 2014b).

We first notice that the performances of our baseline model are better than (or equal to) REPORTED, except the Macro Recall. We reason that these performance disparities may be due to the differences in feature extractions between our implementation and Stab and Gurevych's, and also due to the minor set of new features (e.g., new predicted labels, expanded marker set, component position) that we added in our implementation of the baseline model.

Comparing proposed models with BASELINE, we see that WINDOW, COMBINED, and FULL models outperform BASELINE in important metrics: Kappa, F1, Recall, but TOPIC yields worse performances than BASELINE. However, the fact that COMBINED outperforms BASELINE, especially with significantly higher Kappa, F1, Recall, and F1:Support, has shown the value of Topic-context features. While Topic-context features alone are not effective, they help improve WINDOW model which supports our hypothesis that Topic-context and Window-context features are complementary aspects of context, and they together obtain better performance.

Comparing our proposed TOPIC, WINDOW, COMBINED models with each other shows that COMBINED obtains the best performance while TOPIC performs the worst, which reveals that Topic-context feature set is less effective than Window-context set. While FULL model achieves the best Accuracy, Precision, and F1:Non-support, it has lower performance than COMBINED model in important metrics: Kappa, F1, F1:Support. We reason that the noise caused by word pairs and production rules even dominate the effectiveness of Topic-context

	Reported	Baseline	Topic	Window	Combined	Full
Accuracy	0.863	0.869	0.857	0.857	0.870	**0.877**
Kappa	–	0.445	0.407	0.449	**0.507***	0.481
Macro F1	0.722	0.722	0.703	0.724	**0.753***	0.739
Macro Precision	0.739	0.758	0.728	0.729	0.754	**0.777**
Macro Recall	0.705	0.699	0.685	0.720	**0.752***	0.715
F1:Support	0.519	0.519	0.488	0.533	**0.583***	0.550
F1:Non-support	0.920	0.925	0.917	0.916*	0.923	**0.929**

Table 12: Support vs. Non-support classification performances on held-out test set. Corpus: Persuasive1.

and Window-context features, which degrades the overall performance.

Overall, by combining Topic and Window models, we obtain the best performance. Most notably, we obtain the highest improvement in F1:Support, and have the best balance between Precision and Recall values among all models. These reveal that our contextual features not only dominate generic features like word pairs and production rules, but also are effective to predict minor positive class (i.e., Support).

6.3.2 Task 2: Support vs. Attack

To further evaluate the effectiveness of our approach, we conduct an additional task that classifies an argumentative relation as *Support* or *Attack*. For this task, we assume that the relation, i.e., attachment (Peldszus, 2014), between two components is given, and aim at identifying the argumentative function of the relation. Because we remove the paragraph constraint in this task, we obtain more Support relations than in Task 1. As shown in Table 11, of the total 1473 relations, we have 1312 (89%) Support and 161 (11%) Attack relations. Because this task was not studied in (Stab and Gurevych, 2014b), we conduct 5×10-fold cross validation and use our implementation of Stab and Gurevych's model as the

baseline. We do not optimize the window-size parameter of the WINDOW model, and use the value 3 as set up before. Prediction performance of all models are reported in Table 13. Symbol ** indicates significant difference with the baseline ($p < 0.01$). Because we perform multiple k-folds, we expect significance at lower p-value to capture the stability across runs.

Comparing our proposed models with the baseline shows that all of our proposed models significantly outperform the baseline in important metrics: Kappa, F1, F1:Attack. More notably than in the Support vs. Non-support classification, all of our proposed models predict the minor class (Attack) significantly more effectively than the baseline. The baseline achieves significantly higher F1:Support than WINDOW model. However, F1:Support of the baseline is virtually in a tie with TOPIC, COMBINED, and FULL.

Comparing our proposed models, we see that TOPIC and WINDOW models reveal different behaviors. TOPIC model has significantly higher Precision and F1:Support, and significantly lower Recall and F1:Attack than WINDOW. Moreover, WINDOW model has slightly higher Kappa, F1, but significantly lower Accuracy. These comparisons indicate that Topic-context and Window-context features are equally effective but impact differently to the prediction. The different nature between these two feature sets is clearer than in the prior experiment, as now the classification involves classes that are more semantically different, i.e., Support vs. Attack. We recall that TOPIC model performs worse than WINDOW model in Support vs. Non-support task.

Our FULL model performs significantly worse than all of TOPIC, WINDOW, and COMBINED in Kappa, F1, Recall, and F1:Attack. Along with results from Support vs. Non-support task, this further suggests that word pairs and production rules are less effective and cannot be combined well with our contextual features.

Despite the fact that the Support vs. Attack task (Task 2) has smaller and more imbalanced data than the Support vs. Non-support (Task 1), our proposed contextual features seem to add even more value in Task 2 compared to Task 1. Using Kappa to roughly compare prediction performance across the two tasks, we observe a greater performance improvement from Baseline to Combined model in Task 2 than in Task 1. This is an evidence that our proposed context-aware features work well even in a more imbalanced with smaller data classification task. The lower performance values of all models in Support vs. Attack than

	Baseline	Topic	Window	Combined	Full
Accuracy	0.885	0.886	0.872	0.885	**0.887**
Kappa	0.245	0.305**	0.306**	**0.342****	0.274**
Macro F1	0.618	0.651**	0.652**	**0.670****	0.634**
Macro Precision	0.680	0.692	0.663	**0.697**	0.693
Macro Recall	0.595	0.628**	0.644**	**0.652****	0.609**
F1:Support	0.937	0.937	0.928**	0.936	**0.938**
F1:Attack	0.300	0.365**	0.376**	**0.404****	0.330**

Table 13: Support vs. Attack classification performance in 5×10-fold cross validation. Corpus: Persuasive1.

in Support vs. Non-support indirectly suggest that Support vs. Attack classification is a more difficult task. We hypothesize that the difference between support and attack exposes a deeper semantic relation than that between support and no-relation. We extract textual text similarity and textual entailment features to investigate this hypothesis in the next chapter (Cabrio and Villata, 2012; Boltužić and Šnajder, 2014).

6.4 SUMMARY

In this study, we have presented *context-aware argumentative relation mining* that makes use of contextual features by exploiting information from topic and context sentences. We have explored different ways to incorporate our proposed features with baseline features used in a prior study, and obtained insightful results about feature effectiveness. The proposed contextual features are evaluated with two argumentative relation mining tasks: support vs. non-support and support vs. attack. Experimental results show that topic-context and window-context features are both effective but impact predictive performance measures dif-

ferently. In addition, predicting an argumentative relation will benefit most from combining these two set of features as they capture complementary aspects of context to better characterize the argumentation in justification. Overall, we have supported strongly our second main hypothesis H2 (§1.2) of the effectiveness of topic-context and window-context features in argumentative relation mining.

The results obtained in this study are promising and encourage us to explore more directions to enable contextual features. In Chapter 7, we investigate uses of topic segmentation to identify context sentences and compare this linguistically-motivated approach to our current window-size heuristic. While support vs. attack relation classification are commonly studied in argument mining because this relation scheme is widely applicable to different text genres, we experiment the capabilities of our proposed context features for the attachment problem in Chapter 8, and plan for more advanced schemes, e.g., types of support, in the futures.

7.0 IMPROVING ARGUMENTATIVE RELATION MINING IN STUDENT WRITINGS

In the prior study, we showed that features derived from topic-context (i.e., argument and domain word lexicons) and window-context (i.e., surrounding sentences) help improve significantly performance of argumentative relation classification tasks in persuasive essays. This chapter explores our proposed context-aware argumentative relation model in academic essays which expose different writing style and coding manual than the academic essays. We also propose new window-context features derived from textual similarity and textual entailment, and to use text segmentation for context window formation.

7.1 ACADEMIC ESSAY DATA

Our current study utilizes the corpus of 115 academic essays (Chapter 3). Recall that two experts labeled each sentence of the essays as to whether it is a *Hypothesis* statement, *Support* finding, or *Opposition* finding. If a sentence is identified as a Support or Opposition, it will be linked to the relevant Hypothesis statement. Differently from persuasive essays in which argumentative relations are identified between argument components in a paragraph, argumentative relations in academic essays are determined from findings to hypotheses regardless of paragraph boundaries. As described in §3.3, the Academic Essay Corpus contains 132 argumentative relations with 50 support and 82 opposition.

In persuasive essays, the argumentative relation mining problem assumes that argument components were minimally identified, which mean their positions are known but not necessarily their argumentative labels. We follow the same setting to formulate the argumentative

Label	#pairs
Support	50
Opposition	82
None	702
Total	834

Table 14: Number of argumentative relations in corpus Academic.

relation classification in academic essays: assuming all argumentative sentences are located in a given essay but not necessarily classified, determine argumentative relation of each ordered pair of argumentative sentences. Overall, we form 834 ordered pairs of argumentative sentences in the corpus. Class distribution of this data set is shown in Table 14.

7.2 PREDICTION MODELS

Similarly to our prior study in Chapter 6, we enhance Stab and Gurevych's model to use as a baseline along with Topic and Window models. Combined model has all features in Topic and Window, and Full model is the combination of Baseline, Topic, and Window models. A summary of all models is shown in Figure 12. Beside those prediction models that we have introduced in the previous chapter, we modify Combined in two ways.

7.2.1 Context Window from Text Segmentation Output

First, instead of using the window-size heuristic to form context windows of argumentative sentences, we employ text segmentation to determine context windows' boundaries. Given an essay split by a text segmentation algorithm, the context window of an argumentative sentence includes adjacent sentences in the same segment and same paragraph with the argumentative sentence. When context windows overlap, overlapping context sentences are

resolved by prioritizing the source context window as in Chapter 6. In this study, we experiment with the Bayesian Topic Segmentation algorithm by Eisenstein and Barzilay (2008).[1] The algorithm takes raw text as input and returns a list of positions of segment boundary sentences. An example of topic segmentation output for an academic essay is given in Appendix C. We will compare WINDOW-based models of different window sizes with the model based on text segmentation (referred to as ADwSEG).

7.2.2 Semantic Relation Features

Our other modified COMBINED models exploit semantic relations between short texts, e.g., sentences. Textual entailment and semantic textual similarity have been used in prior studies on identifying whether user comments support or attack a debate topic (Cabrio and Villata, 2012; Boltužić and Šnajder, 2014). For semantic similarity computation, we use the TakeLab STS library which was ranked in top 5 of the SemEval-2012: Semantic Evaluation Exercises to perform the Semantic Textual Similarity task (Sarić et al., 2012).[2] Given two sentences, the program returns a similarity score in range from 0 to 5 in which score 0 means two sentences are on different topics, and score 5 indicates the two sentences are completely equivalent as they mean the same thing. We use the Excitement Open Platform for textual entailment computation (Magnini et al., 2014).[3] This program also takes a pair of sentences as input, but one sentence as a source and the other as a target. The output includes entailment score and label with Entailment means the source sentence is predicted to entail the target sentence, and No-entailment indicates no relation.

We propose to derive features from textual entailment (TE) and semantic textual similarity (STS) between pair of sentences to support argumentative relation classification. In particular, we first simply calculate TE and STS scores between source and target argumentative sentences to use as numerical features. The group of these two semantic relation features is named R1.

We further utilize the context window of argumentative sentences to extract more TE and

[1] https://github.com/jacobeisenstein/bayes-seg
[2] http://takelab.fer.hr/sts/
[3] https://hltfbk.github.io/Excitement-Open-Platform/

STS features. Given a context window of the source argumentative sentence, we calculate STS score between each sentence in the source context window and the target argumentative sentence. Similarly, we calculate STS score between each sentence in the target context window and the source argumentative sentence. The maximum score value is then used as a numerical feature. We expect that the max from a set of STS scores better captures the topic similarity between source and target argumentative sentences than the single STS score.

Because textual entailment is a directed relation, we only consider TE scores from each sentence of source context window to the target argumentative sentences. The maximum TE score value is used as a feature. We calculate the entailment score from the source context window as a whole to the target argumentative sentence, and extract TE score as a feature.

We create a feature group RC by adding 4 semantic relation scores extracted from context windows to R1. While R1 only exploits semantic relations between source and target argumentative sentences, RC is expected to approximate also semantic relations between source sentence's justification and target sentence.

7.3 EXPERIMENT RESULTS

7.3.1 Performance on Academic Essay Corpus

Our first experiment conducts 10×5-fold cross validation on academic essays to compare different models which were proposed in Chapter 6. We do not split data into training and development sets to optimize half-size for context window-based models. In this experiment, we start with context windows with the smallest half-size $n = 1$, which contain at most 3 sentences. In the follow-up experiment, we will use the whole data to quantify the impact of the size of context windows to prediction performance.

Because of the small data with just more than 800 instances, this experiment performs 5-fold cross validation so that training and test folds should have reasonable numbers of instances for each class. We further run 10 times of cross validation to eliminate noise caused by minor classes (i.e., Support and Opposition). Models are trained using LibLINEAR

algorithm (Fan et al., 2008) and cross-validations are conducted using Weka library. Table 15 presents prediction performance on the Academic Essay Corpus using the 5 argumentative relation classification models studied in Chapter 6. Best values are highlighted in bold. Values smaller than baseline are underlined. Symbol ** indicates significant difference with the baseline ($p < 0.01$).

As shown in the table, three of our proposed models, i.e., TOPIC, WINDOW and COMBINED, significantly outperform BASELINE. While BASELINE yielded higher F1 for None class, it achieved lower F1 for positive classes, i.e., Support and Opposition, which are the classes of interest. FULL model performed not better than BASELINE even though it has our proposed context features. In fact, most performance measures of FULL model are significantly lower than those of our other proposed models. These results confirm the finding in Chapter 6 that our proposed topic-context and window-context features are much more effective than the n-gram and production rule features. However, the noise of n-gram and production rule features is dominant, and degrades performance when those features are combined with our proposed features.

While TOPIC model obtains significantly higher F1 for Support class than WINDOW model, it has significantly lower F1 for Opposition class. We hypothesize that topic-context features may help identify support relation more efficiently than opposition because it is more reliable to reason that two words are topically-related than unrelated. In contrast, it seems that discourse relations in context windows become an essential factor to characterize the opposition relations between argumentative sentences. Combining topic-context with window-context features yields the best model, except that COMBINE's F1 for Opposition is lower than that of WINDOW. This shows a sign of conflict between window-context and topic-context feature when predicting opposition relations. We expect to relieve this feature inhibition by adding semantic relations such as textual entailment and textual similarity.

7.3.2 Window-size Impact

In this experiment, we investigate the impact of window-size to the prediction performance of window-context features on academic essays. We vary the half-size parameter and report

	Baseline	Topic	Window	Combined	Full
Accuracy	0.828	0.823**	0.819**	**0.829**	0.827
Kappa	0.291	0.315**	0.315**	**0.342****	0.291
Macro F1	0.493	0.540**	0.521**	**0.553****	0.494
Macro Precision	0.529	0.560**	0.536**	**0.575****	0.528
Macro Recall	0.472	0.525**	0.510**	**0.536****	0.474
F1:Support	0.260	0.399**	0.300**	**0.405****	0.265
F1:Opposition	0.307	0.317	**0.360****	0.344**	0.305
F1:None	**0.912**	0.904**	0.904**	0.909**	0.911

Table 15: Argumentative relation classification performance in 10×5-fold cross validation. Corpus: Academic.

F1 scores of Combined model in Figure 14. In the chart, the X-axis indicates half-size n of context windows. F1 scores of Combined in Table 15 correspond to $n = 1$.[4]

As we can observe, the macro F1 line has two peaks at $n = 4$ and $n = 8$, and its values are stable after $n = 11$. Similarly, F1 scores of Support and Opposition vary much less with large n. This is reasonable that after a certain value, increasing n will introduce only a few number of larger context windows, which affects just a small portion of argumentative sentences. Thus, changes to prediction performance are getting negligible with larger n. These findings are similar to the results on persuasive essays as shown in Figure 13. The best half-size $n = 8$ for academic essays is larger than the best $n = 3$ for persuasive essays probably because academic essays have longer paragraphs in average than persuasive essays (see Table 16).

Looking at per-class F1 scores, we see that with very small n, i.e., $n = \{1, 2\}$, F1 of Support are larger than F1 of Opposition. However, from $n = 3$, F1 scores of Opposition increase and stay at high values when n increases. On the contrary, F1 scores of Support vary

[4]Recall that given a half-size n, the context window has at most n preceding and n following sentences adjacently to the argumentative sentence of interest, so the context window has size $2n + 1$ at the largest.

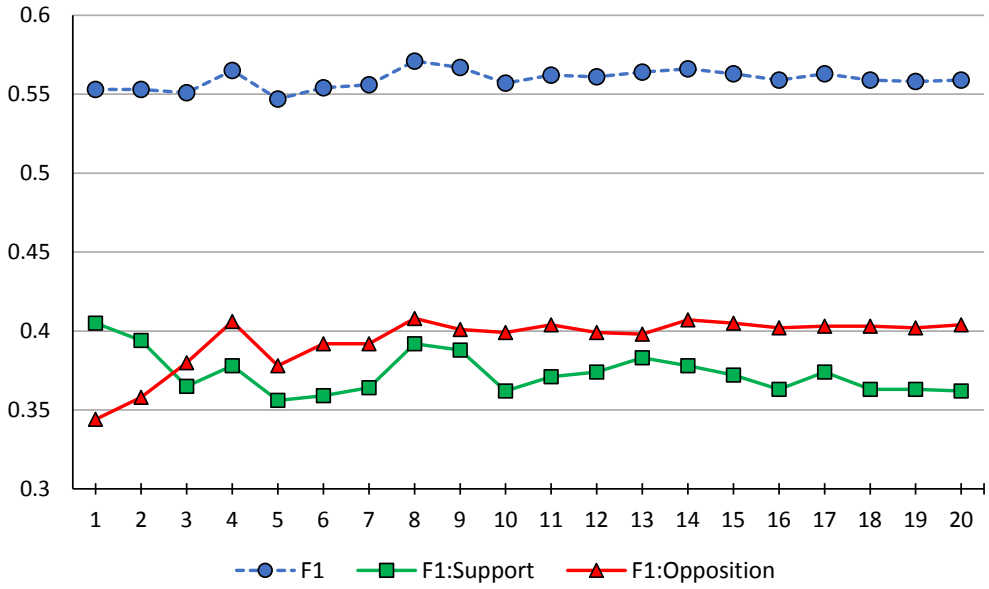

Figure 14: F1 scores of COMBINED model in academic essays by half-size n.

Paragraph length	Academic essays	Persuasive essays
Min	1	1
Max	28	10
Mean	5.28	3
Median	5	3
Std	3.63	1.94

Table 16: Paragraph length in persuasive and academic essays.

more greatly. We, however, observe the trend that large n degrades F1 of Support. With larger context windows, COMBINED can capture more local relations among context sentences and that seems to help identify opposition relation between argumentative sentences. We hypothesize that when developing opposition findings, writers may make argument switches back and forward which can be revealed on the usage of markers and an analysis of discourse relations. However, justification of support findings may expose no to very little reversal of

	Academic essays		Persuasive essays	
	COMBINED$_{n=8}$	ADWSEG	COMBINED$_{n=3}$	ADWSEG
Accuracy	**0.836***	0.829	0.871	**0.873***
Kappa	**0.377***	0.343	0.487	**0.493***
Macro F1	**0.571***	0.545	0.743	**0.746***
Macro Precision	**0.590***	0.567	0.758	**0.762***
Macro Recall	**0.556***	0.530	0.731	**0.734***

Table 17: Cross-validation performance of ADWSEG models in corpora Academic and Persuasive1.

argument flow so that expanding the search for local discourse relations does not gain more information to help prediction.

7.3.3 Text Segmentation-based Context Windows

An inherent problem with the window-size heuristic is that model performance is sensitive to the half-size parameter. For example, while our COMBINED model achieved high performance with the first trial half-size $n = 1$ (Table 15), it could be further improved if the best half-size could be estimated, e.g., with some development data. Therefore, we propose to use text segmentation to approximate context windows without a need to tune the window-size parameter.

Performance of ADWSEG model on academic and persuasive essays is shown in Table 17. For reference models in each corpus, we report COMBINED models with the best half-size n. This gives us upper-bound performance of argumentative relation model using windows-size heuristics. For persuasive essays, we conduct the Support vs. Not task, which classifies each pair of argument components in the same paragraph as holding a support relation or not. All experiments conduct 10×5-fold cross validation.

ADWSEG is shown to perform differently in the two corpora in comparison to the

reference models. In academic essays, ADwSEG performs worse than Combined with $n = 8$, and the differences on major measures are significant ($p < 0.05$). For per-class measures, ADwSEG also returned significantly lower F1 scores of Support and Opposition. Considering Figure 14, we see that ADwSEG even has worse F1 than Combined with other half-size n, although the disparities are insignificant for small $n \leq 3$.

On the contrary, in persuasive essays ADwSEG significantly ($p < 0.05$) outperforms Combined with best $n = 3$ even the performance disparities are quite small. ADwSEG's F1 score of Support is also significantly higher than that of Combined. This result is impressive as it shows a case that topic segmentation-based context window works better than window-size heuristic. However, the advantage of ADwSEG was not observed in academic essays. While we expected that topic segments naturally fits argument justification and thus offers a good alternative of window-size heuristic for context windows, results of ADwSEG in two corpora give both caution and promise on the benefit of text segmentation for argument mining. We believe an analysis on the segmentation quality with different corpora is necessary to explain the result conflict of ADwSEG in academic essays versus persuasive essays. However, such an analysis is out of scope of this thesis. In the course of this study, we report in Tables 18 and 19 average sizes of topic segments as well as source and target context windows as identified by text segmentation.

In academic essays, the average topic segment returned by the text segmentation algorithm has size of 4.13 sentences, while persuasive essays have average size of topic segment about 2.14 (Table 18). As a consequence, ADwSEG model has source and target context windows with average size 6.33 and 5.52 respectively for academic essays, which are twice larger than the average sizes 3.35 and 2.30 for source and target context windows in persuasive essays (Table 19). However, when we compare context windows of ADwSEG with Combined, we do not see any remarkable difference. In particular, Combined$_{n=8}$ has source and target context windows with average size 6.40 and 5.50 in academic essays, respectively. Combined$_{n=3}$ has average size 3.17 and 2.28 for source and target windows in persuasive essays.

We also count number of topic segments that span across paragraphs (Table 18). Assuming that each paragraph should contain completely one or more topics, a large portion

Segment statistic	Academic essays	Persuasive essays
Min size	1	1
Max size	24	13
Average size	4.13	2.14
Segments/essay	6	8
Cross-paragraph segments/essay	2.80	3.92
Cross-paragraph segment ratio	0.40	0.45

Table 18: Statistics on segmentation output in corpora Academic and Persuasive1.

of cross-paragraph topic segments may indicate a noisy output of the text segmentation algorithm. In academic essays, each essay has 6 topic segments in average, and 2.8 are cross-paragraph. With regard to persuasive essays, there are 8 topic segments per essay, and 3.9 of those span across paragraphs. Averaging over all essays, the Academic Essay Corpus has cross-paragraph segment ratio 0.4, and the ratio of Persuasive Essay Corpus is 0.45. Interestingly, persuasive essays have higher ratio of cross-paragraph segments, but ADwSEG performs better than it does in academic essays.

7.3.4 Impact of Semantic Relation Features

We evaluate the impact of semantic relation features, i.e., R1 and Rc, in different combinations with Combined and ADwSEG models. Because semantic relation features in Rc are extracted from context sentences, actual feature values of Rc highly depend on the context windows of source and target argumentative sentences. Thus we expect that impact of Rc features to Combined and ADwSEG are different. We keep the same experimental setting as in prior experiments. For academic essays, we solve the 3-way classification problem: Support vs. Opposition vs. Non-argumentative. Regarding persuasive essays, we perform the Support vs. Not task. Performances are calculated from 10×5-fold cross validation.

Results are shown in Tables 20 and 21. Symbol ** indicates significant difference

Corpus	Academic		Persuasive1	
Context window	Source	Target	Source	Target
ADwSEG	6.33	5.52	3.35	2.30
Combined (with best n)	6.40	5.50	3.17	2.28

Table 19: Average sizes of source and target context windows.

	Combined$_{n=8}$ +			ADwSEG +		
	$\emptyset$	R1	Rc	$\emptyset$	R1	Rc
Accuracy	0.836	0.835	**0.837**	0.829	0.829	**0.833****
Kappa	0.377	0.374	**0.380**	0.343	0.341	**0.359****
Macro F1	0.571	0.569	**0.573**	0.545	0.544	**0.556****
Macro Precision	0.590	0.587	**0.593**	0.567	0.565	**0.578****
Macro Recall	0.556	0.555	**0.558**	0.530	0.528	**0.540****
F1:Support	0.392	**0.393**	**0.393**	0.348	0.344	**0.358****
F1:Opposition	0.408	0.404	**0.413**	0.380	0.379	**0.399****
F1:None	**0.912**	0.911	**0.912**	0.908	0.908	**0.910****

Table 20: Performance of argumentative relation classification by adding semantic relation features. Corpus: Academic.

($p < 0.01$) with the model not using semantic relation features, denoted as $\emptyset$. As we can see adding semantic relation, i.e., textual entailment and semantic textual similarity, features Rc consistently helps improve argumentative relation mining problem across the two corpora. The improvement is significant for ADwSEG model on the Academic Essay Corpus. However, simply using TE and STS scores between source and target argument components does not gain performance increase but decrease. Both Combined + R1 and wSegment + R1 models perform worse than the corresponding Combined and ADwSEG

	COMBINED$_{n=3}$ +			ADWSEG +		
	∅	R1	RC	∅	R1	RC
Accuracy	0.871	0.871	**0.872**	**0.873**	0.872	**0.873**
Kappa	0.487	0.486	**0.490**	0.493	0.492	**0.495**
Macro F1	0.743	0.743	**0.745**	0.746	0.746	**0.747**
Macro Precision	0.758	0.758	**0.759**	**0.762**	0.761	**0.762**
Macro Recall	0.731	0.731	**0.733**	0.734	0.733	**0.735**
F1:Support	0.562	0.561	**0.564**	0.567	0.566	**0.569**
F1:Not-support	0.925	0.925	0.925	**0.926**	0.925	**0.926**

Table 21: Performance of argumentative relation classification by adding semantic relation features. Corpus: Persuasive1.

models. This result shown the advantage of aggregating semantic relation scores in context windows. While max score of STS was used in a prior study on classifying relation between multiple-sentence comments and topic, our proposed approach with context windows allows to incorporate the aggregated scores even when the relation of interest is between two single sentences and/or clauses.

7.4 SUMMARY

In this chapter, we explored different ways of improving argumentative relation mining and evaluate proposed approaches using two corpora of student writings. Our experiments showed a promising result that text segmentation can be used to outperform the window-size heuristic for context window-based models in persuasive essays. However, further analysis is needed to explain how quality of text segmentation affects argumentative relation mining. Furthermore, we proposed to extract textual entailment and semantic textual similarity

relation from context windows of argument components. While the simple TE and STS scores between argument components did not help, the aggregated scores, i.e., max scores of TE and STS, consistently improve prediction across data and models. In conclusion, our results here further support the second main hypothesis H2 (§1.2) of the effectiveness of topic-context and window-context features in argumentative relation mining.

8.0 END-TO-END ARGUMENT MINING IN STUDENT ESSAYS

This chapter describes our end-to-end argument mining system that can process unannotated essays for extracting argument component and identifying argumentative relations. Our main motivation is to have an automated argument parsing system for studying application of argument mining in automated essay scoring. The system makes use of our improved models for argument component and argumentative relation classifications. For argument component identification, we implement the supervised sequence model that is proposed in a study by Stab and Gurevych (2017).

In 2017, Stab and Gurevych released a second corpus of persuasive essays and developed a joint model for recognizing argumentation structure in essays. We train our argument mining system using this corpus to take advantage of the larger data set. To the best of our knowledge, this is the largest corpus with 402 annotated essays for argument mining research. We, however, only employ a pipeline paradigm for our argument mining system: argumentative relation classification can take prediction output from argument component classification but not vice versa. Experimental results show that our argument mining system can achieve a high performance close to the best system by Stab and Gurevych (2017) even without a joint prediction model.

8.1 PIPELINE ARGUMENT MINING

In general, an argument mining system involves three major basic tasks (Mochales and Moens, 2011; Peldszus and Stede, 2015; Stab and Gurevych, 2017). (1) *Argument compo-*

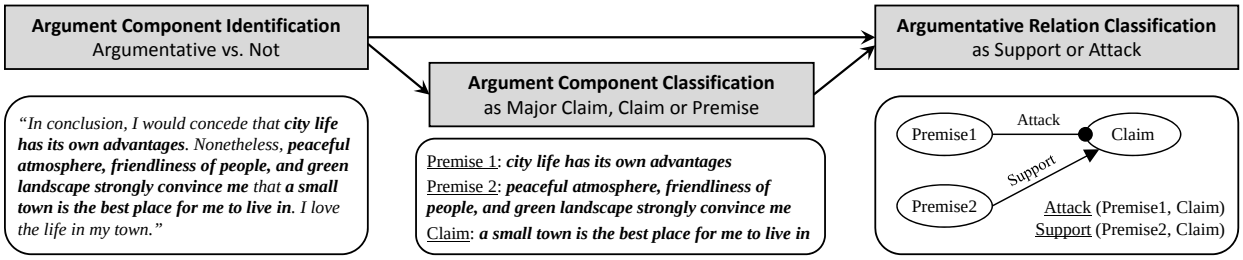

Figure 15: Pipeline argument mining. Each basic argument mining task is associated with the expected output from a given excerpt. In left text box, argument components are in bold face. Label of argument components may be passed to argumentative relation classification as features to improve performance.

nent identification aims at determining the boundaries of argumentative text units[1], i.e., argument components. (2) *Argument component classification* labels each component for its argumentative function, for example, Major Claim (author's stance toward a topic), Claim (controversial statement that argues for/against the stance), Premise (reason that underpins/rebuts the validity of claim) (Stab and Gurevych, 2014a). (3) *Argumentative relation classification* determines if an ordered pair of argument components has a relation, i.e., Support vs. Attack. Different approaches have been proposed to solve the second and third tasks in order, i.e., *pipeline argument mining* (Stab and Gurevych, 2014b), or jointly (Peldszus and Stede, 2015; Stab and Gurevych, 2017).

We follow Stab and Gurevych (2014b) and design a pipeline argument mining system that makes use of our context-aware argument mining models. Figure 15 depicts our pipeline argument mining which was tailored for persuasive essays. For the argument component identification task, we adapt the supervised sequence model proposed in their paper. This model is described in detail in the next section.

To solve the argument component classification, our system employs the wLDA+4

[1]Text portions (e.g., sentences, clauses) that have specific roles in forming the arguments in the text (Peldszus and Stede, 2013).

model (§5) with novel topic-context features derived from our lexicon of argument and domain words. Because our argument mining system will be applied for argumentative writings that share much similarity with the persuasive essays used in our studies, we use the argument and domain word lexicon that was learned from the development data of persuasive essays. Finally, with regard to argumentative relation mining, we employ the wSegment model which uses topic segmentation to identify context windows of argument components.

Stab and Gurevych (2017) split the corpus into training and test sets with 322 and 80 essays, respectively. We train our argument mining system using the training set, and compare the performances on the test set with the reported results by Stab and Gurevych (2017). Parameters of prediction models in our argument mining system are optimized through 10-fold cross validation within training set. Creation and statistics of this corpus were introduced in §3.2. Class distributions in training and test sets are shown in Table 22.

8.2 SUPERVISED SEQUENCE MODEL FOR ARGUMENT COMPONENT IDENTIFICATION

Stab and Gurevych (2017) encodes argument components using BIO tagset that every token in the essay has either B, I or O label depending on whether it is at the beginning, inside, or outside the argument component. Figure 16 shows an example argumentative sentence with BIO labels assigned to its tokens. The authors used Conditional Random Field algorithm (Lafferty et al., 2001) to learn a sequence labeling model. We adapt their model to use in our argument mining system. For each tokens, we extract following features as proposed in the prior study (Stab and Gurevych, 2017):

- *Structural features*
  - Token position: token present in first or last paragraph; token is first or last token in sentence; relative and absolute token position in document, paragraph and sentence.
  - Punctuation: token precedes or follows any punctuation, full stop, comma and semicolon; token is any punctuation or full stop.

Class	Training set	Test set
Argument Components		
Major Claim	598 (12%)	153 (12%)
Claim	1202 (25%)	304 (24%)
Premise	3023 (63%)	809 (64%)
In-paragraph Argument Component Pairs		
Linked	3023 (18%)	809 (16%)
Not-linked	14227 (82%)	4113 (84%)
In- and Cross-paragraph Argumentative Relations		
Support	3820 (90%)	1021 (92%)
Attack	405 (10%)	92 (8%)

Table 22: Class distributions in training and test sets of corpus Persuasive2.

  - Position of covering sentence: absolute and relative position of the token's covering sentence in the document and paragraph.

- *Syntactic features*
  - Part-of-speech: the token's part-of-speech.
  - Lowest common ancestor (LCA): length of the path to the LCA with the following and preceding token in the parse tree normalized by the depth of the tree.
  - LCA types: constituent types of two LCA of the current token with its preceding and following tokens.

- *Lexico-syntactic features*: for each token t, extract its uppermost node n in the parse tree with the lexical head t. A lexico-syntactic feature is defined as the combination of t and the constituent type of n. We also consider the child node of n in the path to t and its right sibling, and extract their lexico-syntactic features (Soricut and Marcu, 2003).

- *Probability feature*: we compute the maximal conditional probability of the current token

It	's	true	that	technology	and	computers	do	make	their	jobs	easier	but	it	cannot	definitely	replace	them	.
B	I	I	I	I	I	I	I	I	I	I	I	O	B	I	I	I	I	O

Figure 16: Tokens with BIO tagset.

t_i being the beginning of an argument component given its preceding tokens:

$$\max_{n \in \{1,2,3\}} P(\text{tagset}(t_i) = B | t_{i-n}, ..., t_{i-1})$$

The probability is estimated as a division of the number of times the preceding tokens precede a token t_i with tag B by the total number of occurrences of the preceding tokens in the training data.

We add six features derived from argument and domain word lexicon:

- AD features: two boolean features indicate whether the current token is an argument word or a domain word; four boolean features indicate whether the preceding or following tokens are argument or domain words.

Performances of argument component identification (ACI) on the test set are reported in Table 23. The first row shows reported results in Stab and Gurevych (2017). The two next rows present results of our implementation of Stab and Gurevych (2017), and our improved version with AD features (referred to as adACI). Our implementation of Stab & Gurevych's model obtained close results to what is reported in their paper. Our improved version with AD features yielded the best performance.

To further evaluate the impact of AD features in the task, we conduct 10-fold cross validation on training set and report results in Table 24. T-tests show that all performance improvement by AD features are significant with $p < 0.01$. Given this result, we integrate improved ACI model into our argument mining system to solve argument component identification.

Model	F1	Prec.	Recl.	F1:B	F1:I	F1:O
Human upper bound	*0.886*	*0.887*	*0.885*	*0.821*	*0.941*	*0.892*
ACI by Stab and Gurevych	0.867	0.873	0.861	0.809	0.934	0.857
ACI (our implementation)	0.865	0.868	0.862	0.799	0.938	0.859
adACI	**0.872**	**0.877**	**0.868**	**0.814**	**0.939**	**0.863**

Table 23: Argument component identification performance on the test set. Corpus: Persuasive2.

Model	F1	Prec.	Recl.	F1:B	F1:I	F1:O
ACI	0.850	0.853	0.848	0.778	0.931	0.841
adACI	**0.856***	**0.859***	**0.854***	**0.791***	**0.932***	**0.844***

Table 24: 10-fold cross validation performance of argument component identification in the training set. Corpus: Persuasive2.

8.3 ARGUMENT COMPONENT CLASSIFICATION

The next component in our argument mining system aims at classifying each argument component as *MajorClaim*, *Claim*, or *Premise*. This problem setting is more practical than the classification problem we solved in Chapters 4 and 5. While the old 4-way classification also considered non-argumentative sentences, this 3-way classification works on the output of the argument component identification step, and thus can skip non-argumentative sentences.

For this task, Stab and Gurevych (2017) had significantly improved their first model in 2014b with novel features. The most notable change that Stab and Gurevych made to their model was that they used dependency triples rather then production rules. It has been shown in our prior study that dependency triples are more effective then production rules for this classification task. Besides, they expanded their discourse marker set and included output

	ADw4	ADw4 + Prob.	ADw4 + Embedding
Accuracy	0.820	**0.845***	0.818
Kappa	0.656	**0.704***	0.650
Macro F1	0.792	**0.825***	0.788
Macro Precision	0.795	**0.829***	0.795
Macro Recall	0.790	**0.821***	0.783
F1:MajorClaim	0.866	**0.896***	0.862
F1:Claim	0.628	**0.681***	0.618
F1:Premise	0.883	**0.898***	0.884

Table 25: 10-fold cross validation performance of ACC models in the training set. Corpus: Persuasive2.

of a discourse parser (Lin et al., 2014). Finally, the authors proposed two novel feature sets.

The probability features are the conditional probabilities of the current component C having the argumentative label t in {MajorClaim, Claim, Premise} given the sequence of preceding tokens p:

$$P(\text{label}(C) = t|p)$$

Conditional probabilities P are estimated from the training data.

The second new feature set is based on the pre-trained Word2Vec word embedding (Mikolov et al., 2013). They summed vectors of words in the argument component and preceding tokens within the sentence. The summation vector was then added to the feature space.

While their new lexical features, e.g., dependency parse and discourse relations, overlap with our features, their probability and embedding features are novel. However, their experiments showed that adding probability and embedding features yielded very little performance improvement, i.e., less than 0.5%. Moreover, because the conditional probability of argumentative labels are estimated in the training set, we worry that relying on probabil-

ity features may over-fit with training data and degrade performance on unseen test data. Therefore, we experiment with adding these feature sets to our proposed model.

8.3.1 Experiment Results: Cross Validation in Training Set

For argument component classification, we employ the ADw4 model, and train the model using LibSVM (Chang and Lin, 2011). As shown in Table 25, adding probability features significantly improves performance for ADw4, but adding embedding features does not. In fact, ADw4 + Embedding performs worse than our original model. However, the high performance of ADw4 + Probability is expected because the probabilities are computed in the training set.

8.3.2 Experiment Results: Performance in Test Set

Argument component classification performances on the test set are shown in Table 26. Base column reports performance of the improved base classifier, and ILP column presents performance of the ILP-based joint prediction (Stab and Gurevych, 2017). ILP model exploits the mutual information between argument components and argumentative relations to optimize the prediction. As a result, ILP obtained remarkably better performance than Base.

Comparing our proposed models, adding probability or embedding features do not improve performance of ADw4. While ADw4 + Probability yielded the best performance in the training set, its performance is the lowest in the test set. This suggests that probability features might cause over-fitting. Although embedding features allow a much more efficient representation than bag-of-words, a simple usage like adding word vectors to feature space seems to not help. However, given all successes of word embeddings in many different NLP tasks, we plan to explore more advanced usage of word embeddings in argument mining in the future.

Comparing ADw4 with Stab and Gurevych's results, our model achieved significantly higher performance than their base classifier. Despite the fact that ADw4 does not have any information from argumentative relation prediction, its F1 score is comparable to ILP's F1, and it even could predict MajorClaim better than the ILP model did. This makes us

	Base	ILP	ADw4	ADw4+Prob.	ADw4+Embd.
Accuracy	-	-	0.848	0.839	0.841
Kappa	-	-	0.702	0.684	0.687
F1	0.794	**0.826**	0.825	0.814	0.816
Precision	-	-	0.831	0.825	0.825
Recall	-	-	0.822	0.805	0.810
F1:MajorClaim	0.891	0.891	**0.910**	0.901	0.906
F1:Claim	0.611	**0.682**	0.667	0.646	0.648
F1:Premise	0.879	**0.903**	0.900	0.896	0.896

Table 26: Test performance of ACC models. Corpus: Persuasive2.

believe that we can further improve state-of-the-art performance when implementing joint prediction from our base classifier.

Given the above results, we integrate ADw4 into our argument mining system to solve argument component classification.

8.4 ARGUMENTATIVE RELATION IDENTIFICATION

The last component in our end-to-end argument mining system aims at argumentative relation mining task. Differently from the previous tasks, argumentative relation mining can be cast to different classification problems depending on which representation of argumentative relation is of interest. In prior chapters, we have demonstrated our context-aware models by solving the argumentative relation mining problem in the forms of Support vs. Not and Support vs. Attack classifications. In this chapter, we experiment with the attachment problem that determines whether a pair of argument components with one as the source and the other as the target holds an argumentative relation (i.e., argumentative relation identi-

fication). If so, the two argument components are said to be linked in order (Peldszus and Stede, 2015; Stab and Gurevych, 2017). Support and attack relations then can be classified from linked pairs of argument components. For the argumentative relation mining task, we deploy ADwSEG2 model which is ADwSEG model with textual entailment and semantic textual similarity features.

Based on a prior model developed using corpus Persuasive1, Stab and Gurevych (2017) proposed an improved model for argumentative relation mining with new features and larger training data from corpus Persuasive2. The authors first limited numbers of n-grams and production rules to the 500 most frequent items in each set. This certainly is to address the large and sparse feature space generated by generic n-grams and production rules. Our proposed models have addressed this issue by eliminating domain words in n-grams and dependency triples.

The authors also introduced the pointwise mutual information feature that measures the dependency between a lemmatized token t of an argument component and the direction d of argumentative relation that attaches to the component:

$$\text{PMI}(t, d) = \log \frac{P(t, d)}{P(t)P(d)}$$

where $d \in \{\text{incoming, outgoing}\}$. $P(t, d)$ is the probability that token t occurs in an argument component with either incoming or outgoing relations. Probabilities are estimated from the training set of the corpus. Given the over-fitting issue with the probability feature for argument component classification, we do not include the PMI feature to our model for argumentative relation mining because our argument mining system will apply to different data sets of student essays that vary on topic domains and writing characteristics.

Finally, Stab and Gurevych added two shared-noun features that determine if the two argument components share a noun, and count number of shared nouns. These features are motivated by a fact that premises and claims in classical syllogisms share the same subjects (Govier, 2013). Our model has similar features that count shared argument and domain words. Our preliminary experiments showed that adding shared noun features does not help our model.

	Base	ILP	COMBINED$_{n=3}$	ADWSEG	ADWSEG2
Accuracy	-	-	0.861	0.865	0.866
Kappa	-	-	0.449	0.462	0.467
F1	0.717	**0.751**	0.724	0.730	0.733
Precision	-	-	0.751	0.759	0.760
Recall	-	-	0.705	0.711	0.713
F1:Linked	0.508	**0.585**	0.528	0.540	0.544
F1:Not-linked	0.917	0.918	0.919	0.921	**0.922**

Table 27: Test performance of models for attachment task. Corpus: Persuasive2.

8.4.1 Test Performance of Models

We compare performance of our three proposed models with the published results (Stab and Gurevych, 2017) and report in Table 27. COMBINED$_{n=3}$ model uses half-size $n = 3$ to identify context window of argument components (Chapter 6). ADWSEG takes output of a text segmentation algorithm to form context windows. ADWSEG2 adds textual entailment and semantic textual similarity features to ADWSEG (Chapter 7). Columns Base and ILP show performance of the base classifier and the joint model in the prior study (Stab and Gurevych, 2017).

As we can see in the table, ILP is the best model and has F1 score greatly improved in comparison with that of the base classifier. Among our proposed models, ADWSEG2 achieves the best performance. Also, ADWSEG2 has larger performance disparity with COMBINED$_{n=3}$ than with ADWSEG, which confirms that topic segmentation-based context windows yielded higher improvement than textual semantic relations. ADWSEG2's performance is higher than Base, which again demonstrates the effectiveness of our contextual features. However, the fact that ADWSEG2 performed worse than ILP clearly shows the advantage of joint prediction. We plan to implement a similar ILP framework and expect to further improve both argument component and argumentative relation classifications.

8.5 END-TO-END PERFORMANCE

In previous sections, we have described in detail our pipeline argument mining system and compared our proposed models with the baselines for different argument mining tasks. While prediction performance of each argument mining task was reported, those results do not reflect the true capability of the system because each task was performed using the input with true labels instead of output from the task before. In particular, both argument component classification and argumentative relation identification were fed with true argument components (AC). In this section, we test the end-to-end performance of our argument mining system.

Considering essays in the test set, argument components are first automatically extracted. Then, the extracted argument components are classified for their argumentative labels (i.e., MajorClaim, Claim, Premise) and pairs of components that hold a argumentative relation are identified. To measure the end-to-end performance, we first form an union set U of extracted argument components E and true argument components T which are missed at the identification task.

$$\mathrm{U} = E \cup T, \quad E \cap T = \emptyset$$

With the extracted argument components E, we assign true argumentative labels to those that have exact matches with true argument components. The other extracted argument components should have true non-argumentative labels (i.e., false positive). Because the true argument components in T are not given to later classification tasks, the creation of U is to assure that the missing argument components in T, and subsequently the argumentative relations among them, are taken into account when measuring performance. Thus, our performance measures for argument component classification and argumentative relation identification embed the performance of argument component identification.

The test set has 1266 true argument components (AC). Our argument component identification (ACI) model returned 3460 textual spans (i.e., sub-sentence portions) in which 1272 were identified as AC. Out of the extracted AC, 941 have exact matches with true AC (i.e., true positive). The confusion matrix is given in Table 28. Our union set U includes 1597 AC in which 1272 were returned by our model (set E) and 325 true AC were misidentified

	True argumentative	True non-argumentative
Predicted argumentative	941	331
Predicted non-argumentative	325	1868

Table 28: Confusion matrix of argument component identification on the test set. Corpus: Persuasive2.

	True MajorClaim	True Claim	True Premise	True Non
Predicted MajorClaim	81	10	1	64
Predicted Claim	15	138	50	95
Predicted Premise	0	77	569	172
Predicted Non	57	79	189	0

Table 29: Confusion matrix of argument component classification on the test set. Corpus: Persuasive2.

as non-argumentative (set T). We also wanted to mention that approximate match, i.e., two text spans are considered a match if their overlap portion is greater than some threshold (Persing and Ng, 2016), should be more favorable for the boundary extraction problem. We, however, use exact match in this study to give a sense of argument mining difficulty. Approximate match may make more sense when we are aware of how much flexibility an end-application allows for argument mining output.

8.5.1 Argument Component Classification

Given the set U, Table 29 presents the confusion matrix of argument component classification (ACC). The row Predicted Non does not reflect the misclassification by our ACC model, but shows errors carried over from ACI's results. Our ACC model achieves end-to-end F1 of 0.421 with F1:MajorClaim = 0.524, F1:Claim = 0.458, and F1:Premise = 0.699. Stab and

	True Linked	True Not-linked
Predicted Linked	252	369
Predicted Not-linked	449	3978

Table 30: Confusion matrix of argumentative relation identification on the test set. Corpus: Persuasive2.

Gurevych (2017) did not report the end-to-end performance of their models so we do not have a baseline for direct comparison. To give more intuition on the task difficulty, here we present the end-to-end measures reported in a study by Persing and Ng (2016). The authors developed a heuristic for argument component candidate extraction and an ILP framework for joint prediction. They conducted 5-fold cross validation in corpus Persuasive1. Essays in the corpus are of the same kind with those in Persuasive2 that we are using for this study (see Chapter 3). Their best system with exact matching returned F1:MajorClaim = 0.169, F1:Claim = 0.374, and F1:Premise = 0.534.

8.5.2 Argumentative Relation Identification

From 1272 argument components returned by our ACI model, our argumentative relation identification (ARI) model formed 4854 ordered pairs of AC in which 621 were predicted as Linked. With regard to 325 true AC which were missed by our ACI model, 189 Linked pairs of AC were not considered as input of the ARI model.

To have an end-to-end F1 for Linked pairs, we add 189 true Linked pairs to the cell [Predicted Not-linked, True Linked] in the confusion matrix. Thus, the confusion matrix in Table 30 has 189 more instances than the total number of pairs formed by our ARI model. With this adjustment, our ARI model obtained F1:Linked = 0.381. Persing and Ng (2016) achieved F1 = 0.136 using corpus Persuasive1, but their task was more difficult when it classified Support, Attack and No-relation.

Because all True Positive instances are included in our end-to-end measures, we can

roughly compare the end-to-end F1 scores with the results of individual tasks in previous sections. We observe a great reduction in performance with our end-to-end setting. For example, F1:Linked has decreased 30% while F1 of ACC has reduced nearly 50%. Despite the fact that argument component identification could obtain high performance (about 1.5% lower than human upper bound), the performance degradation in end results are remarkable which shows the essential value of a good ACI model.

8.6 SUMMARY

This section presents the end-to-end performance of our pipeline argument mining system in the corpus Persuasive2. The reported performances are promising but show need of improvement. Our plan for enhancing our argument mining system includes improving the ACI model and implementing joint prediction. We also suggest to use approximate match for ACI to increase model coverage when applying argument mining to a real task.

9.0 AUTOMATED ESSAY SCORING: AN EXTRINSIC EVALUATION OF ARGUMENT MINING MODELS

9.1 INTRODUCTION

Applications of argument mining in real-world tasks, e.g., automated writing evaluation, have gained an increasing interest. While prior studies proposed different approaches to improve argument mining, no study has investigated the impact of argument mining accuracy to the application tasks. In this research, we study argument mining for automated persuasive essay scoring and examine whether more accurate argument mining models help to predict essay scores more accurately. Our essay scoring study uses a larger set of features enabled by argument mining output compared to prior work, and performs argument mining at different levels of automation. The experimental results not only confirm that more accurate argument mining yields higher essay scoring performance, but also gives interesting insights on the contribution of different argumentation features.

In automated essay scoring (AES), argument mining offers new abilities for AES systems to consider argumentation aspects of persuasive essays beyond legacy essay dimensions, e.g., grammar, mechanics, discourse structure. Research has proposed different argumentation features for persuasive essay score prediction to improve automated scoring performance, e.g., numbers of claims and support relations, tree-form vs. chain-form arguments. In these studies, different levels of automation have been employed for argumentation feature extraction (Ghosh et al., 2016; Klebanov et al., 2016). However, no prior studies have investigated the impact of argument mining accuracy to the scoring performance. This issue is of particular interest given that argumentation features can be computed at different steps of the argument mining pipeline (Figure 15), and error propagation may degrade the

effectiveness of features computed at later steps of the pipeline (Chapter 8).

Our current study is the first time that argument mining models are extrinsically compared in terms of how their accuracy impacts the performance of automated persuasive essay scoring. By argument mining accuracy, we mean the classification performance of each basic argument mining task. We first adapt two argument mining models developed for persuasive essays: Stab and Gurevych (2014b), and our pipeline argument mining system (Chapter 8) which we name ArgS and ArgN respectively. We review prior studies and consider a large set of argumentation features for essay scoring. We hypothesize that *argumentation features computed by more accurate argument mining models will predict essay score more accurately.* Furthermore, we categorize the argumentation features into sets corresponding to the basic tasks where they are computed, and compare them for insights of their contributions to essay score prediction.

Most persuasive essay scoring tasks adopt a holistic scoring scheme in which argument convincingness is just a dimension of the overall essay quality (Song et al., 2014; Ong et al., 2014). Even when the argument quality could be an explicit criterion to evaluate the essay in some cases, predicting argument strength may require feature selection from argument mining output (Persing and Ng, 2015). Therefore, applying argument mining to automated essay scoring is usually taken as a feature engineering task. On the other hand, the literature on extrinsic evaluation of Natural Language Processing systems has shown that better intrinsic performance might not lead to better extrinsic performance (Belz and Gatt, 2008; Chiu et al., 2016). These bring-up the necessity of an empirical study on the effect of argument mining accuracy and argumentation features to AES performance.

9.2 ARGUMENT MINING SYSTEMS AND AES DATA

Our current study exploits different corpora for argument mining systems and essay scoring experiments. The ArgS system is our implementation of models proposed by Stab and Gurevych (2014b), and trained using the first corpus of persuasive essays (§3.1). Our end-to-end argument mining system ArgN was introduced in Chapter 8, which employs our

proposed argument mining models, and was trained using the second corpus of persuasive essays (§3.2). The two argument mining systems are implemented following the pipeline structure as depicted in Figure 15. Given an input essay, argument components are first extracted, then classified as Major Claim, Claim or Premise. Finally, for every ordered pairs of argument components in each paragraph, the systems determine if there exists a support relation or not. Both ArgS and ArgN systems are equipped with adACI model for argument component identification (§8.2).

For essay scoring experiments, we use essays of the TOEFL11 corpus (Blanchard et al., 2013). The corpus contains over 12 thousand TOEFL essays written by non-native test takers to argue for opinions towards issues stated in 8 writing prompts. Although the corpus was first introduced for a Native Language Identification shared task, the coarse-grained holistic scores (i.e., Low, Medium, and High) of essays were provided. Particularly, we use a set of essays from this corpus which has been used in a prior study on argumentation features for essay score prediction (Ghosh et al., 2016).

To evaluate a set of coarse-grained argumentation features for persuasive essay scoring, Ghosh et al. (2016) annotated 107 essays (Te107) from the TOEFL11 corpus using a similar annotation scheme as proposed in Stab and Gurevych (2014a) for the corpus Persuasive1. However, because our ArgN system is trained using the corpus Persuasive2 which were annotated with the improved scheme, there are certain types of argumentative relations in Te107 essays that cannot be identified by ArgN system, e.g., relations between claims. To better estimate ArgN's performance on Te107, we discard annotated relations between claims, from premises to major claims in Te107 essays. Te107 data includes 105 Major Claims, 468 Claims, and 603 Premises.[1] There are 4178 ordered pairs of argument components in which 507 pairs hold support relations (656 pairs before our adjustment). As this annotated dataset has true boundaries of argument components, we can evaluate ArgS and ArgN when the inputs are gold-standard argument components. Scores of Te107 essays are reported in Table 31. Because the essays were sampled in a way that keeps similar numbers of essays across scores, the score distribution does not match the distribution of

[1]The data was made available online at `github.com/debanjanghosh/argessay_ACL2016/`. We, however, observe a difference in number of Major Claims than reported in their paper.

#essays	107
Low score	31
Medium score	36
High score	40

Table 31: Statistics of TE107 data.

the TOEFL11 corpus.

The advantage of TE107 data is that its essays were both graded for writing quality and annotated for argumentation structures. This data is ideal for us to study the impact of argumentation features on predicting essay scores, and evaluate argument mining systems extrinsically on an automated essay scoring task. However, there are certain dissimilarities between TOEFL11 essays and those in the training corpora of the two argument mining systems used in this study. First, essays in our training corpora are practice writings which might be prepared without any limits of time or references. On the contrary, TOEFL11 essays were written in real tests with time limits and no reference material. Second, while persuasive essays in the training corpora were manually collected to assure that they are argument-rich (Stab and Gurevych, 2014a, 2017), TOEFL11 essays were sampled with an emphasis on variety to assure the inclusion of both high and low quality essays (Blanchard et al., 2013). Thus, TOEFL11 essays are expected to have lower quality as well as greater quality range compared to persuasive essays in our training corpora. Although both corpora are opinionated essays written by student authors, their quality disparity make TOEFL11 essays a challenging data set to evaluate our argument mining models.

9.3 INTRINSIC EVALUATION OF ARGUMENT MINING SYSTEMS

We first evaluate the performance of two argument mining pipelines, ARGS and ARGN, using true argument components provided in TE107 data. In this setting, we conduct both

System	F1:MajorClaim	F1:Claim	F1:Premise	F1:Support	F1:Not-support
	10-fold cross validation				
ArgS	0.570	0.549	0.732	0.226	0.906
ArgN	**0.604***	**0.606***	**0.753***	**0.320***	**0.915**
	Test performance				
ArgS	0.453	0.295	0.710	0.148	**0.917**
ArgN	**0.622***	**0.508***	**0.751***	**0.211***	0.915

Table 32: Argument mining performance in Te107 essays when inputs are true argument components.

System	Low-score Set		Medium-score Set		High-score Set	
	F1:AC	F1:Support	F1:AC	F1:Support	F1:AC	F1:Support
ArgS	0.400	0.234	0.482	0.115	0.501	0.050
ArgN	0.570	0.179	0.598	0.156	0.644	0.242

Table 33: Test performance in Te107 for different score sets. F1:AC reports macro average F1 score of argument component classification.

in-domain cross validation and out-of-domain validation. In-domain cross validation evaluates approaches in ArgS and ArgN through 10-fold cross validation. This experiment merely compares efficiency of prediction features in (Stab and Gurevych, 2014b), and features proposed in our studies. Out-of-domain validation evaluates the two argument mining systems in which prediction models are pre-trained using different argument mining corpora as mentioned above.

Table 32 reports prediction performance of the two argument mining pipelines when the argument components were manually identified.[2] Symbol * denotes difference with $p <$

[2]In 10-fold cross validation on Te107, we obtained lower argumentative relation performance than re-

0.05 when comparing performances of ArgS and ArgN. As we can see, ArgN system significantly outperformed ArgS models with all measures, except for the test F1 scores of Not-support. Not-support is the major class, and performance difference between the two systems are not significant. We, however, are more interested in F1 scores of the Support class, where ArgN yielded significantly higher scores. These results again confirm our prior findings regarding the effectiveness of our contextual features in ArgN. The results also show that the test performances of both systems on Te107 essays are lower than 10-fold performances for most measures. This is probably due to differences in writing quality and annotation between training essays and TOEFL11 essays that we have discussed.

To investigate the hypothesis whether essay quality affects argument mining performance, we report the test performance of ArgS and ArgN for differrent essay score sets in Table 33. Considering the argument component classification, we can see that the average F1 score increases when the prediction moves from low score to high score sets. Regarding argumentative relation classification, we also have ArgN obtained higher F1:Support in high-score essays than low-score essays. Although the score sets have different sizes and class distributions, these results roughly show that argument mining performs more accurately in high-quality essays than low-quality ones. However, the F1:Support of ArgS is higher in the low-score set and lower in the high-score set. We observe that the essays in high-score set are usually longer and produces more candidate pairs of components. The very skewed distribution in high-score essays might affect ArgS which caused its low performance.

Our next evaluation tests the two argument mining systems with automatically identified argument components. This evaluation follows exactly the same setting as in Chapter 8, and has only one difference in that test essays in this study (i.e., Te107) are from different data domain than the training data for argument mining. Model adACI is employed to extract argument components from the essays. Performance of argument component identification is shown in Table 34. Referring to results in Table 23, the model adACI performed much worse in Te107 data than in Persuasive2 corpus. The low performance in Te107 is expected

ported in (Ghosh et al., 2016). The reason was that we extracted all possible ordered pairs of argument components in each paragraph (Stab and Gurevych, 2014b), while Ghosh et al. (2016) only extracted certain pairs based on true labels of argument components. Thus, our setting is much more challenging and applicable to unannotated data.

F1	Prec.	Recl.	F1:B	F1:I	F1:O
0.578	0.575	0.591	0.436	0.757	0.540

Table 34: Argument component identification performance of adACI model in TE107 data.

System	F1:MajorClaim	F1:Claim	F1:Premise	F1:Support	F1:Not-support
ARGS	0.078	0.226	0.343	0.088	**0.962***
ARGN	**0.156***	**0.258***	**0.404***	**0.126***	0.947

Table 35: Argument mining performance in TE107 essays when inputs are automatically identified argument components.

because of differences in writing quality, topic domains and annotation.

Regarding end-to-end argument mining in TE107, as shown in Table 35, test performance on TE107 is much lower for both argument mining systems when argument components were automatically rather than manually identified. However, we still observe that ARGN performed better than ARGS, except for F1:Not-support. As we are more interested in detecting Support relations, F1:Not-support measure is less important. Overall, intrinsic comparative evaluations confirm that ARGN can predict argumentation structure more accurately than ARGS.

9.4 ARGUMENTATION FEATURES FOR PREDICTING ESSAY SCORES

This section describes different argumentation features that have been used in prior studies for persuasive essay scoring (Persing and Ng, 2015; Ghosh et al., 2016; Klebanov et al., 2016; Wachsmuth et al., 2016), and introduces new features for a more comprehensive evaluation. Because the argumentative relation models that we implemented for ARGS and ARGN only

classify pairs of components in the same paragraph as having support relation or not, we do not include argumentation features that involve attack relations or cross-paragraph argument component pairs. Table 36 lists 38 argumentation features in 5 sets that we study in our essay scoring experiments.

For argument component (AC) features, we use raw counts as well as the ratios of argument components and argumentative sentences (i.e., sentences that contain at least one argument component) over the total number of sentences in the essay. Numbers of argument components and argumentative sentences were widely used in prior studies on argument mining for essay score prediction (Ghosh et al., 2016; Klebanov et al., 2016). Our preliminary analysis found moderate correlations ($r > 0.7$) between number of argument components (also argumentative sentences) and essay length (i.e., word and sentence counts). Therefore, argument count features are expected to simulate the effect of essay length features.

Wachsmuth et al. (2016) hypothesized that essays largely argue sequentially, so they restricted to sequences of types (i.e., Thesis, Conclusion, Premise) of argumentative discourse units (i.e., argument flow) in paragraphs to mine reliable patterns of argumentation structure of persuasive essays. For example, argument flows (Conclusion, Premise) and (Conclusion, Premise, Premise) are found to be the most frequent in the International Corpus of Learner English (ICLE) (Granger et al., 2009). We adapt their idea to extract bigrams of types of argument components from paragraphs of essays to use as features. With three possible argumentative labels: MajorClaim, Claim and Premise, we have 9 possible typed bigrams of argument components. We do not consider the MajorClaim–MajorClaim bigrams which do not hold an argumentative relation, and retain 8 remaining typed bigrams. Also, we count number of paragraphs that have simultaneously MajorClaim and Claim, Claim and Premise, or MajorClaim and Premise.

For argumentative relation features, we count number of Claims that are supported by Premises, number of dangling Claims which are not supported by any Premises, number of Premises that support Claims.

Argumentation structure typology features (TS) were first proposed in (Ghosh et al., 2016). The authors constructed a directed acyclic graph of support relations for each paragraph, and defined three argumentation structure typologies: *Chain*-structure (i.e., Claim is

	Argument component features (AC)
1, 2	Number and fraction of argument components over total number of sentences in essay (Ghosh et al., 2016)
3, 4	Number and fraction of argumentative sentences (Ghosh et al., 2016)
5	Total number of words in argument components
6	Number of paragraphs containing argument components (Persing and Ng, 2015)
7	Whether the essay has paragraph without any argument component (Persing and Ng, 2015)
	Component label features (CL)
8	Number of Major Claims (*this study*)
9, 10	Number and fraction of Claims over total number of sentences (Persing and Ng, 2015; Ghosh et al., 2016)
11, 12	Number and fraction of Premises (Persing and Ng, 2015; Ghosh et al., 2016)
13	Average number of Premises per Claim (Klebanov et al., 2016)
	Argument flow features (AF)
14	Number of paragraphs that contain Major Claims and Claims (Persing and Ng, 2015)
15	Number of paragraphs that contain Major Claims and Premises (*this study*)
16	Number of paragraphs that contain Claims and Premises (*this study*)
17–24	Frequency of 8 typed bigrams of argument components (*this study*)
	Argumentative relation features (RL)
25	Number of supported Claims (Ghosh et al., 2016)
26	Number of dangling Claims (Ghosh et al., 2016)
27	Number of supporting Premises (*this study*)
28	Number of paragraphs that have support relations (*this study*)
	Argumentation structure typology features (TS)
29	Number of *Chain*-structures (Ghosh et al., 2016)
30	Number of *Tree*-structures (*this study*)
31	Number of *Tree*-structures with height = 1 (Ghosh et al., 2016)
32	Number of paragraphs that contain *Chain*-structures (*this study*)
33	Number of paragraphs that contain *Tree*-structures (*this study*)

Table 36: Argumentation features for essay score prediction

the root of single-brand tree), *Tree*-structure of height > 1 ($Tree_{h>1}$), and *Tree*-structure of height $= 1$ ($Tree_{h=1}$). Typology features are essentially different from argument flow features. While the former requires the existence of support relations, the other merely considers the appearance order of argument components. Due to the rare occurrence of $Tree$-structures in essays (Wachsmuth et al., 2016), we group $Tree_{h>1}$ and $Tree_{h>1}$-structures together.

9.5 ESSAY SCORE PREDICTION IN TE107 DATA

We evaluate two argument mining pipelines, i.e., ArgS and ArgN, with respect to how accurately their argumentation features predict essay scores in Te107 data to leverage the annotation. While Te107 has a small number of essays, and the score distribution does not truly represent the TOEFL11 corpus, its annotation allows us to derive argumentation features from true labels of argumentation structure.

Given a set of features, an essay score prediction model is trained using Logistic Regression algorithm and evaluated in 10×10-fold cross validation to obtain reliable performance estimation (Kohavi, 1995). The data is reshuffled and re-stratified before each 10-fold run. Reported performance figures include Cohen's kappa (κ) and quadratically-weighted kappa (qwk). While qwk is a standard measure in AES literature (Shermis and Burstein, 2013), κ is included because the prediction model is essentially a classifier. For each set of argumentation features, feature values are extracted in three ways: (1) from true argument components and argumentative relations (TrueLabel), (2) from output of ArgN, and (3) from output of ArgS.

9.5.1 AES Performance Based on Human-identified Argument Components

We first evaluate the argumentation features when their corresponding argument mining models work on human-identified argument components. This setting assumes true argument components are provided so effectiveness of argumentation features depends on the accuracy of argument component and argumentative relation classifications. Therefore, TrueLabel, ArgN and ArgS have identical values for AC features.

As reported in Table 37, argumentation features (except AF features) extracted by TrueLabel outperform those extracted from the output of ArgN and ArgS in score prediction. Symbols ** and † mean significantly higher and lower than the other two with $p < 0.01$ (because we perform multiple k-folds, we expect significance at lower p-value to capture the stability across runs), respectively. However, the performance disparity is larger for relation-based features (i.e., RL, TS) than component-based features (i.e., CL and AF). These could

	Component-based			Relation-based		
Feature set	AC	CL	AF	RL	TS	All
	κ					
TrueLabel	*0.583*	**0.591****	**0.449**	**0.466****	**0.384****	0.402
ArgN	*0.583*	0.504	0.440	0.318	0.031 †	**0.422**
ArgS	*0.583*	0.486	0.370 †	0.197 †	0.112	0.317 †
	qwk					
TrueLabel	*0.765*	**0.768****	0.686	**0.747****	**0.620****	**0.636**
ArgN	*0.765*	0.744	**0.695**	0.454	0.139	0.608
ArgS	*0.765*	0.729 †	0.577 †	0.423 †	0.165	0.559 †

Table 37: Essay score prediction performance in Te107 data. Argument components are manually identified.

be explained by the fact that argument component classification's output is more reliable than argumentative relation classification's output (see results in Table 32).

Comparing the two argument mining systems, we see that ArgN's argumentation features return significantly higher *qwk* and κ than ArgS, except for TS features. However, the absolute κ and *qwk* values of TS features are really small which makes us reason that neither argument mining systems derive reliable topology features. In fact, topology features involve multiple relations to form a structure, thus it is much more difficult for an argument mining system to approximate a topology feature as it is extracted from true label. In Ghosh et al. (2016), TS features were shown useful even they were computed from output of an argumentative relation model. This does not conflict with our finding here because their argumentative relation model solved a simplified problem and achieved high F1 scores (see footnote 2).

Comparing different sets of argumentation features, the general trend is that component-based features (AC, CL and AF) are more effective than relation-based features (RL and

Feature set	AC	CL	AF	RL	TS	All
	κ					
ArgN	*0.506*	0.367	**0.307****	0.171	0.018	**0.381****
ArgS	*0.506*	**0.383**	0.230	**0.175**	**0.094****	0.294
	qwk					
ArgN	*0.716*	**0.633**	**0.512****	**0.312***	0.057	**0.536**
ArgS	*0.716*	0.603	0.423	0.259	**0.189****	0.514

Table 38: Essay score prediction performance in Te107. Argument components are automatically identified.

TS). However, while RL features by TrueLabel are very competitive, those derived from argument mining output perform worse than component-based features. This may be due to poor results of argumentative relation classification. These facts suggest that argument component-based features are more favorable choices for AES tasks until we can have more reliable argumentative relation models.

When combining all argumentation features, all TrueLabel, ArgN and ArgS degrade performance compared to using only AC features, which reveals feature interaction and inference. Therefore, feature selection is necessary for the best performance.

9.5.2 AES Performance Based on Automatically Identified Argument Components

In this experiment, argumentation features are all extracted from output of the end-to-end argument mining process. Because both ArgN and ArgS are equipped with the same model for argument component identification, they have the identical values for AC features. Essay score prediction experiments are conducted following the same setting as above. Results are presented in Table 38.

First, essay score prediction performances are much lower when argument mining systems

have to take predicted argument component as inputs. The most effective features are still argument component statistics (AC). At the other end, TS features are the least reliable with very low κ and qwk. Combining all argumentation features yields significantly lower performance than using component-based features alone. These results confirm our findings from the previous experiments that complex relation-based features such as topology are not ready to be applied in AES tasks, and obtaining the best AES performance may require feature selection.

Comparing the two argument mining systems, we do not generally have ArgN's argumentation features perform better than ArgS. In particular, TS features of ArgN perform significantly worse than those of ArgS, which confirm the findings in Table 37. Moreover, performance disparity between ArgS and ArgN is not consistent across κ and qwk of CL and RL features. Similarly to AES performance based on true argument component, ArgN's all features returned higher qwk and κ than ArgS's all features.

To give a fair comparison between the two argument mining systems with respect to AES performance of their argumentation features, we conducted the experiments with a comprehensive set of argumentation features but did not apply any optimization such as feature selection. This is both an advantage and disadvantage of our analysis. At first, this analysis achieves the ultimate goal of the current study that is giving insights of impact of argument mining accuracy to AES performance. We compare argumentation features derived from not only different argument mining tasks, but also different argument mining systems.

However, in a different perspective the analysis has not answered the real question whether argumentation features from output of an end-to-end argument mining model eventually helps improve AES performance. By that we actually mean our evaluation of AES performance was not grounded on a base AES model. Let us consider a naive AES model that uses only word-count (WC) features and obtains 10-fold $\kappa = 0.552$ and $qwk = 0.743$ in Te107 data. Although word-count is much less descriptive than argumentation features, it alone can predict essay scores far better than any combination of argumentation features. From this fact, we do not expect argumentation features to be used as a replacement for baseline features in existing AES systems. In the next chapter we look for answers to the

research question of whether argumentation features can supplement base AES models with information of argumentation structure to improve persuasive essay score prediction.

9.6 SUMMARY

To the best of our knowledge, we are the first to perform an extrinsic comparison of argument mining systems for persuasive essay scoring. We evaluated argument mining models in two extreme cases where argument components were identified by human versus automatically. We also studied a larger set of argumentation features for persuasive essay score prediction than prior studies. Therefore, another contributions of our study are insights on the impact of argumentation features to essay score prediction. Among our results, notable findings include (1) features based on argument components can predict essay score better than features derived from argumentative relations; (2) argumentation features extracted by more accurate argument mining models predict essay scores more accurately. For the next study, we will extend the extrinsic evaluation by adding argumentation features to a base essay scoring system.

10.0 ARGUMENT MINING FOR IMPROVING PERSUASIVE ESSAY SCORE PREDICTION

10.1 INTRODUCTION

In Chapter 9, we showed that argumentation features derived by more accurate argument mining models can predict essay scores more accurately. However, the study also showed that the low performance of argumentative relation models make its argumentation features much less reliable in essay score prediction. In fact, although our proposed argument mining system has an improved argumentative relation model, its argument typology features did not predict essay scores better because the performance of argumentative relation classification is still low (Tables 32 and 35). Furthermore, adding all argumentation features significantly degraded AES performance compared to using only argument component features (Tables 37 and 38). These results seem to suggest that argument component features are more favorable choices for automated essay scoring while argumentative relation features are not ready for this task. Although such a finding is reasonable given experimental results, it indeed does not conclude about the true benefit of using argumentation features in automatically predicting persuasive essay scores. We hypothesize that while argumentation features may not effectively predict essay scores when used alone, they can help gain improvement when used with a base model for essay score prediction. This chapter seeks such a benefit of argumentation features when they are evaluated in the context of a base AES model.

Prior studies on argument mining for persuasive essay scoring have not considered the role of enhancing a base AES model adequately. Ghosh et al. (2016) was the first to study argumentation features but only compared against sentence-count feature. Klebanov et al. (2016) used word-count feature as the baseline to evaluate performance improvement when

adding proposed argumentation features. While their studies showed added values of argumentation features, such conclusions may not generally apply to real AES tasks when scoring models are usually tailored for the best performance. In a study by Wachsmuth et al. (2016), argument flow features gained improvement for state-of-the-art models. However, their study aimed at predicting trait scores of essays which are organization and argument strength. Our study aims for improving holistic score prediction by exploiting argumentation features.

In this study we evaluate AES models using both cross validation and held-out test sets. While the former minimizes bias in comparisons, the latter enables a direct comparison with a prior study.

10.2 DATA AND BASE MODEL FOR AUTOMATED ESSAY SCORING

In this study, we continue to utilize the TOEFL11 corpus for AES experiments. We, however, use the essay sets sampled by Klebanov et al. (2016), which have a larger size and their score distribution are similar to the original corpus. In particular, the authors compiled a training set of 6074 essays and a test set of 2023 essays. We did not experiment with this data set because it does not have human annotation. Numbers of essays with different scores are reported in Table 39. More than half of the total essays receive medium scores, and low-score essays have the smallest portion.

For the purpose of easy integration and evaluation, our current study implements a competitive base model for essay score prediction. We review the literature on AES and employ a variety of features that were found effective for essay scoring (Shermis and Burstein, 2013; Dikli, 2006). Our first group of features (LENGTH) include 5 numerical features that model fluency and readability of the writing. While we do not have a direct model for writing fluency, we use essay length features as an estimate because it is believed that a more fluent writer will be able to write more (Klebanov et al., 2016). Readability features are adapted from Automated Readability Index formula which involves average sentence length and average word length.[1]

[1] https://en.wikipedia.org/wiki/Automated_readability_index

	Training	Test
#essays	6074	2023
#prompts	8	
Low score	655	222
Medium score	3318	1101
High score	2101	700

Table 39: Essay score data description.

- Word count: number of tokens in the essay.
- Sentence count: number of sentences in the essay.
- Character count: number of characters not including white-space characters.
- Average sentence length: average number of words per sentence.
- Average word length: average number of characters per word.

Our second group of features (CONTENT) aim for modeling different aspects of writing mechanics including spelling errors, content-richness and sentence complexity:

- Spell: number and percentage of spelling errors in the essay. We use the Jazzy library with Ispell dictionary to detect incorrect words.[2]
- Stop-word: number and percentage of stop-words in the essays.
- Prompt: number and percentage of words found in the writing prompt.
- Vocabulary: number and percentage of words found in the SAT 5000-word list.[3]
- Comma: number of commas, semi-colons, and colons.
- Punctuation: numbers of question marks, exclamation marks and double quote symbols.

Word and POS n-grams are commonly used in AES research, but we found that adding these features makes our base model significantly less effective. While the utilized features are simple, their performance are shown competitive in our next experiments.

[2]http://jazzy.sourceforge.net/
[3]http://www.freevocabulary.com

Count	ArgS		ArgN	
Sentences	91349 (15.03±6.02)			
Argument components	85205 (14.02±5.44)			
Major-Claims	4119	(0.67±0.83)	8237	(1.35±0.96)
Claims	15460	(2.54±1.93)	24423	(4.02±2.31)
Premises	65626	(10.80±5.20)	52545	(8.65±5.10)
Support relations	12547	(2.06±2.38)	29322	(4.82±3.87)

Table 40: Statistics of argument mining output in train set. Mean and standard deviation are parenthesized.

10.3 IMPROVING ESSAY SCORING WITH ARGUMENTATION FEATURES

10.3.1 Cross Validation in Training Set

Our current study continues to use 38 argumentation features and compare two argument mining systems ArgS and ArgN as described in Chapter 9. 38 argumentation features are grouped in 5 sets: argument component (AC), argumentative label of components (CL), sequence of argument components (i.e., argument flow, AF), argumentative relation (RL), and argument typology (TS). Essays in the data set are first segmented into argument component by using adACI model (Chapter 8). The two argument mining systems are then employed to label argument components and identify support relations. Finally, argumentation features are extracted from argument mining output. Number of predicted argument components and support relations are shown in Table 40. T-test results show that statistical values of ArgS and ArgN are all significantly different with $p < 0.0001$.

Given two sets of baseline features and 5 sets of argumentation features, we evaluate different combinations. AES models are trained using Logistic Regression algorithm in Weka (Hall et al., 2009), and evaluated in 10-fold cross validation. Essay scoring performance are

Feature set	κ		qwk	
Length	0.440†		0.567†	
Content	0.453†		0.582†	
Length + Content (Base)	0.475		0.599	
	ArgS	ArgN	ArgS	ArgN
AC	0.341†		0.469†	
Base + AC	0.474		0.599	
CL	0.119†	0.118†	0.215†	0.211†
Base + CL	0.475	0.482*	0.599	0.605*
AF	0.058†	0.042†	0.141†	0.091†
Base + AF	0.477	0.476	0.602	0.601
RL	0.029†	0.054†	0.058†	0.092†
Base + RL	0.466†	0.478	0.592†	0.602
TS	0.015†	0.000†	0.029†	0.000†
Base + TS	0.475	0.470	0.600	0.595
ARG	0.346†	0.364†	0.481†	0.494†
ARG + Base (All)	0.480	0.486*	0.604	**0.611***
All – AC	0.475	**0.487***	0.599	0.610*
All – CL	0.477	0.484*	0.602	0.608*
All – AF	0.480	0.480	0.603	0.604
All – RL	0.481	0.481	0.605	0.606*
All – TS	0.485	**0.487***	0.608	**0.611***

Table 41: 10-fold cross validation performance of essay score prediction of base and argumentation features. ARG denotes all argumentation features.

shown in Table 41. Best values are highlighted in bold. Symbols * and † indicate significantly higher and lower than Base values ($p < 0.05$), respectively.

As shown in the middle part of the table, 4 of 5 argumentation feature sets derived from ArgS do not gain improvement for the base model when adding them individually to the base model. Only AF set is effective in that combining them with base features yielded higher κ and qwk.

With regard to ArgN, 3 of 5 argumentation feature sets (i.e., CL, AF and RL) helped improve the base model, and the improvement by CL features are significant. Significant improvements are obtained when combining argumentation features. The best combination includes base features with all argumentation features except argument structure typology (TS) features. This provides an evidence that argumentation features indeed help improve essay score prediction, and the performance increase is more significant when the argument mining output is more accurate.

Considering each set of argumentation features individually, we can see that they almost cannot predict essay score when used alone except AC features which yielded $\kappa = 0.341$ and $qwk = 0.469$. Interestingly, although AC features returned the highest performance among argumentation features, those do not help improve base performance at all. In fact, count features in AC correlate moderately to strongly with the corresponding Length features. For example, Pearson's correlation tests for number of argumentative sentences vs. number of sentences, number of words in argument components vs. number of words returned $r > 0.8, p = 0$. Thus, we hypothesize that AC features do not provide more predictive information than those captured in Length features.

The least effective argumentation features are TS features in that ArgN's TS features had κ and qwk almost zeros. Adding TS features extracted by either ArgS or ArgN to the baseline both degraded performance of the base model. As shown in the bottom part of the table, we achieved the best κ and qwk by removing ArgN's TS features from all features.

Despite the low performance by each argumentation feature set, the performance increase by combining those argumentation features with the base AES model confirms our hypothesis of the improvement benefit of argumentation features. The improvements are significant when the base model is enhanced with ArgN's features. This makes us believe that base features such as length statistics and writing mechanics grade essays at a coarse grain and argumentation features help further refine the prediction.

Feature set	κ		qwk	
	ArgS	ArgN	ArgS	ArgN
Length + Count (Base)	0.463		0.591	
ARG	0.346†	0.361†	0.481†	0.492†
ARG + Base (All)	0.470	0.475*	0.597	0.600*
Base + AC	0.471*	0.471*	0.596	0.596
Base + CL	0.467	0.473*	0.594	0.600*
Base + AF	0.465	0.469*	0.594	0.597*
Base + RL	0.464	0.466	0.592	0.593
Base + TS	0.465	0.465	0.593	0.592
All – AC	0.468	0.474*	0.595	0.600*
All – CL	0.466	0.474*	0.595	0.600*
All – AF	0.470	0.474*	0.597	**0.601***
All – RL	0.470	0.469	0.597	0.595
All – TS	0.472*	**0.476***	0.599*	**0.601***

Table 42: Cross-prompt performance of essay score prediction of base and argumentation features.

The above finding is also confirmed in Table 42 where we conduct cross-prompt validation. In each run, we hold essays of a prompt as a test data and use essays of the 7 remaining prompts for training the models. Similarly to 10-fold cross validation results, adding argumentation features improve cross-prompt AES performance, and the improvements are significant ($p < 0.05$) for features extracted by ArgN. For ArgS's features, a significant improvement is obtained when removing TS features from the complete set. Not using TS features also helps obtain the best $\kappa = 0.476$ and $qwk = 0.601$ for ArgN. Overall, AES results by adding ArgN features are better than adding ArgS features most of the times.

Cross-prompt experiment is considered a more difficult evaluation because test and training essays are of different writing topics. The AES improvements by adding argumentation features are revealed more clearly in the cross-prompt setting which demonstrates the topic-independent advantage of argumentation features. In the next chapter, we further study this aspect of argumentation features in cross-domain AES.

Our experiments did not do an exhaustive feature selection, but aimed for evaluating argumentation features by groups to get an insight of how possible outputs of argument mining help improve AES. When comparing results in 10-fold cross validation and cross-prompt validation, a finding is that the best combination of argumentation features is {AC, CL, RL, AF} and it is true for both ArgS and ArgN. Argument typology features (TS) perform the worst when used alone, and give the lowest (or second lowest) performance when adding to the base model. Although adding TS features still gains improvement for the base AES model (by a small amount), we hypothesize that the value of typology features is restricted by the low performance of argumentative relation mining. In future work, we plan to improve argumentative relation mining with joint prediction and study if relation-based features (i.e., RL and TS) can be more effective.

10.3.2 Test Performance

In this experiment, we evaluate argumentation features and the base features on a held-out test set as described in Klebanov et al. (2016). This allows us to directly compare our results with the prior study. For the best performance of the base AES model, we conduct 10-fold cross validation in the training set for model selection. The result shows that Random Forest algorithm (Breiman, 2001) works the best. Therefore, all AES models in this experiment are trained with Random Forest algorithm.[4] Test performance is shown in Table 43. Values higher than Base are highlighted in bold.

First, our base AES model performs much better than the word-count baseline used by Klebanov et al. (2016). The author reported a test performance on this data set using

[4]Logistic Regression that was used in our previous experiments was not set up for ridge regularization so that we can assure all features are considered in the training process. By this way, we had a fair evaluation of each feature set. However, we believe not all features are equivalently effective so we exploit Random Forest, a learning algorithm with built-in feature selection for the best results.

Feature set	κ		qwk	
(Klebanov et al., 2016)	0.389		0.540	
Length + Content (Base)	0.486		0.604	
	ArgS	ArgN	ArgS	ArgN
ARG	0.340	0.361	0.474	0.491
ARG + Base (All)	0.484	0.490	0.602	**0.607**
Base + AC	0.481	0.481	0.600	0.600
Base + CL	**0.488**	**0.493**	**0.606**	**0.612**
Base + AF	0.484	0.484	0.597	**0.605**
Base + RL	**0.489**	0.486	0.603	0.603
Base + TS	**0.490**	0.485	**0.608**	0.604
All – AC	0.477	**0.496**	0.597	**0.611**
All – CL	**0.489**	**0.492**	0.603	**0.608**
All – AF	0.483	**0.491**	0.601	**0.609**
All – RL	0.488	**0.508**	0.604	**0.622**
All – TS	0.483	**0.503**	0.602	**0.618**

Table 43: Test performance of essay score prediction of base and argumentation features.

word-count feature with z-transform yielding $\kappa = 0.365$ and $qwk = 0.518$. Our baseline even yielded notably higher performance than their best model which combined word count with 9 argument structure features to obtain $\kappa = 0.389$ and $qwk = 0.540$. Second, using argumentation features to augment the base AES model yielded better performance. However, adding all argumentation features does not return the best performance. In fact, with regard to ArgN's argumentation features, the best result is obtained when not using RL features: AES $\kappa = 0.508$ and $qwk = 0.622$ as shown in the table. About ArgS's features, the best result is when using CL and TS features with the base model: AES $\kappa = 0.501$ and $qwk = 0.618$ (not shown in the table). While argument structure typology (TS) has little

value in cross-validation AES (see Tables 41 and 42), it helps a lot for ArgS features to improve AES performance. Regarding ArgN features, using the best combination of argumentation features (i.e., {AC, CL, RL, AF}) which was determined in cross-validation AES above, we obtain the second best result in this experiment: $\kappa = 0.503$ and $qwk = 0.618$. Even though this does not gain the best result, the improvement is impressive given the fact that the learning algorithm was optimized for the base AES model.

Overall, the test results again confirm our prior findings of the value of argumentation features for automated essay scoring, and more accurate argument mining helps gain higher improvement.[5] The results also suggest that the best set of argumentation features for automated essay scoring is an open problem and may need extensive studies to determine for different use cases.

10.4 SUMMARY

Our current study evaluates argumentation features for essay scoring in the context of a competitive base model for automated essay scoring. The results strongly suggest that argumentation features extracted by a more accurate argument mining system improve essay score prediction more effectively. With the use of a base AES model, we showed that argumentation features extracted by an end-to-end argument mining system indeed improve essay scoring performance significantly. Thus, this study support our third main hypothesis H3 (§1.2) and brings up a stronger evidence about an application of argument mining for essay scoring tasks. To the best of our knowledge, none of the prior studies have addressed completely the matter of end-to-end argument mining for improving holistic score prediction in persuasive essays.

[5] Argument mining accuracies are mentioned based on evaluation in previous studies. We do not have human annotation to conclude whether ArgN is more accurate than ArgS and by how much in this data.

11.0 ARGUMENT MINING FOR CROSS-DOMAIN ESSAY SCORE PREDICTION

11.1 INTRODUCTION

For the best performance, AES models are usually trained and tested with data of the same or similar topic domains. However, this requires additional data whenever an AES model is deployed for new writing prompts. Because collecting and annotating new data are typically costly or even not possible for quick deployment, domain adaptation in AES has recently been studied as a remedy for the lack of new data (Phandi et al., 2015; Dong and Zhang, 2016). In this study, we further investigate the application of argument mining in AES by showing that argumentation features which are not dependent on topic domains can help improve AES in cross-domain evaluation.

11.2 DATA AND BASE MODEL

Our current study utilizes the Kaggle's Automated Student Assessment Prize (ASAP) data[1] which has been studied widely in automated essay scoring research (Phandi et al., 2015; Dong and Zhang, 2016; Taghipour and Ng, 2016). The ASAP data consists of 8 essay sets each of which include essays of the same prompt. Selected essays range from an average length of 150 to 550 words per response. All essays were written by students ranging in grade levels from Grade 7 to Grade 10. All essays were hand graded and were double-scored. Each of the eight data sets has its own unique characteristics. Phandi et al. (2015) and Dong

[1] https://www.kaggle.com/c/asap-aes/data

Set	#essays	Average length	Score range	Median
1	1783	350	2–12	8
2	1800	350	1–6	3

Table 44: Essay score data description.

and Zhang (2016) have utilized the ASAP data to develop domain adaptation algorithms for the AES problem. They experimented with different pairs of essay sets in which essays of one set were used to train the models, and essays of the other set were for testing.

Among 8 writing prompts, we use two prompts whose essays are argumentative. Data statistics of the two essay sets are shown in Table 44. Essays of both sets were double-graded but while the essays of set 2 have resolved scores, essays of set 1 have finals score as the summation of the two expert scores.

> Prompt 1: More and more people use computers, but not everyone agrees that this benefits society. Those who support advances in technology believe that computers have a positive effect on people. They teach hand-eye coordination, give people the ability to learn about faraway places and people, and even allow people to talk online with other people. Others have different ideas. Some experts are concerned that people are spending too much time on their computers and less time exercising, enjoying nature, and interacting with family and friends. Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you.
>
> Prompt 2: Write a persuasive essay to a newspaper reflecting your views on censorship in libraries. Do you believe that certain materials, such as books, music, movies, magazines, etc., should be removed from the shelves if they are found offensive? Support your position with convincing arguments from your own experience, observations, and/or reading.

Our primary goal in the current study is to evaluate argumentation features in cross-domain AES. We hypothesize that argumentation features which abstract over the arguments and argumentation structure of the writing will work effectively even in cross-domain AES. Thus we differentiate our study from prior studies which proposed different machine learning approaches for domain adaptation in AES, e.g., correlated linear regression (Phandi et al., 2015) and automatic features using neural network (Dong and Zhang, 2016). As our base AES model, we use a publicly available open-source AES system called "Enhanced AI Scoring

Engine"[2]. EASE system was ranked in the top three of the Kaggle ASAP competition despite the fact that it used simple features as described in (Phandi et al., 2015):

1. Length:
   - Number of characters
   - Number of words
   - Number of commas
   - Number of apostrophes
   - Number of sentence ending punctuations (".", "?", "!")
   - Average word length (in character).
2. Prompt:
   - Number and fraction of words in the essay that appears in the prompt divided by the total number of words in the essay.
   - Number and fraction of words in the essay which is a word or a synonym of a word that appears in the prompt.
3. Bag of words:
   - Count of useful unigrams and bigrams (unstemmed)
   - Count of stemmed and spell corrected useful unigrams and bigrams
4. Part-of-speech: number and fraction of good POS sequence over the total number of words.

EASE system uses NLTK library[3] to tag essays and WordNet[4] to extract synonyms. While bag of words and POS sequences are commonly used in AES, EASE proposed to use refined ngrams and POS features for the best performance. Useful n-grams were defined as n-grams that separate high-score essays and low-score essays, determined using the Fisher test. EASE use the top 200 n-grams for each of unstemmed and stemmed set. To collect good POS sequences, 12 novels in the collection The Adventures of Sherlock Holmes by Sir Author Conan Doyle[5] were tagged and POS sequences of size 2 to 4 were collected. For each essays, ratio of good POS, i.e., POS sequences found in the collection, is computed.

[2]https://github.com/edx/ease
[3]http://www.nltk.org/
[4]https://wordnet.princeton.edu/
[5]The texts are made available online by Project Gutenberg (http://www.gutenberg.org).

	Essay set 1		Essay set 2	
Feature set	κ	qwk	κ	qwk
(Phandi et al., 2015)	–	0.781	–	0.621
EASE	0.316	0.792	0.463	0.663
ARG	0.308	0.763	0.414	0.612
EASE + ARG	**0.328***	**0.797**	**0.475***	**0.676**

Table 45: In-domain performance of essay score prediction in ASAP data. ARG denotes all argumentation features.

11.3 EXPERIMENT RESULTS

11.3.1 In-domain Cross Validation

Similarly to our previous study, we augment EASE with argumentation features which were described in Table 36. However, because essays of ASAP data do not have paragraphs, all paragraph-related features are not available for ASAP essays, thus we have only 25 argumentation features. Argumentation features are extracted from output of our argument mining pipeline ArgN. Because the test sets of Kaggle ASAP data are not publicly available, we follow prior studies to conduct 5-fold cross validation for each essay set. In this experiment, AES is formulated as a classification problem. EASE system uses Stochastic Gradient Boosting provided in Scikit-learn library[6] to train its AES model. We keep all default settings of EASE system and its training process. The only modification we make, which is the only focus of this study, is that we add argumentation features into EASE's feature set.

Table 45 reports 5-fold cross validation performance of the EASE model with and without argumentation features. Symbol * means significantly higher than EASE ($p < 0.05$). As shown in the table, argumentation features perform significantly worse than EASE features. However, similarly to results in TOEFL11 data, the current experiment shows that adding

[6]http://scikit-learn.org

argumentation features improves AES performance. Improvements in κ are significant.

Phandi et al. (2015) and Taghipour and Ng (2016) also reported 5-fold cross validations using EASE features with ASAP data. While we do not have the data splits used in those prior studies, we experimented with different runs of 5-fold cross validation for EASE, and observed that results of different runs were very close to each other and all higher than those reported in the prior studies (Phandi et al., 2015; Taghipour and Ng, 2016). A possible reason is that the EASE system used in our study utilizes a stochastic gradient boosting algorithm while the EASE models in previous studies used Bayesian Linear Regression and Support Vector Machine algorithms. The authors only utilized features extracted by EASE but did not use the learning algorithm which was implemented for the system. Our next experiment shows that the learning algorithm of EASE is also effective for cross-domain predictions.

11.3.2 Cross-domain Validation

Phandi et al. (2015) were the first to conduct cross-domain AES with ASAP data. With the essay sets 1 and 2, they experimented with set 1 as the training data and set 2 as the test data. Because essays of sets 1 and 2 have different score range, they scaled essay scores into an intermediate range of [-1, 1] and solved AES as a regression problem. Given the regression output, predicted values are re-scaled back to the score range of test essays, and κ and qwk can be computed. We follow their experiment setting for score scaling. However, we use EASE in regression mode which activates the gradient boosting regressor of the system. We conduct two cross-domain experiments in which each of essay set will be training and test data turn by turn.

Cross-domain results are reported in Table 46. Values higher than EASE are highlighted in bold. Best results in (Phandi et al., 2015; Dong and Zhang, 2016) are reported in the top rows. In Dong and Zhang's experiment, essay scores were scaled to range [0, 1] before the machine learning process. Similarly to the in-domain experiments, our use of EASE obtains higher qwk than the prior studies, which may be due to different learning algorithms. However, while the prior studies conducted experiments with other essay sets, we only work

with sets 1 and 2. Therefore, results in Table 46 is not an evidence to conclude that Gradient Boosting algorithm of EASE is generally better than prior studies for cross-domain AES. However, because the main focus of our current study is the impact of argumentation features to cross-domain AES, using a learning algorithm that is particularly good for the data of interest gives us a better context to conclude our hypotheses.

Recall that argumentation features are placed in 5 sets: AC (argumentative components), CL (argument component label), AF (argument flow), RL (argumentative relation label), and TS (argument structure typology). First, we see that cross-domain Set:2→1 (set 2 as training and set 1 as test data) has much lower performance than cross-domain Set:1→2 (set 1 as training and set 2 as test data). Also, cross-domain performances are generally lower than in-domain performances (see Table 45). We reason that scaling essay scores in range [1, 6] and [2, 12] to smaller range, e.g., [-1, 1], then re-scaling to original ranges will cause information loss, which degrades performance. The information loss is more severe when the target range is larger than the original range, e.g., the case of Set 2 → 1. Our experiments in Appendix F shows that the choice of intermediate range affects greatly to regression performance.

Second, we observe that adding argumentation features generally improve cross-domain AES. While combining all argumentation features with EASE (i.e., EASE + ARG) returned higher κ and qwk for both cross-domain settings, the results also show that better improvements are achieved when not using all argumentation features. We experimented with different combination of argumentation feature sets and record the best combination for each cross-domain setting. For Set:1→2, the best performance is obtained when using AC, CL, and TS features with EASE for $\kappa = 0.336$ and $qwk = 0.649$. For Set:2→1, we have the best $\kappa = 0.053$ and the best $qwk = 0.529$ when adding AC, RL and TS to EASE. Both argument component and argumentative relation features are present in the best set, which shows the necessity of complete argument mining from argument component identification to argumentative relation classification. However, comparing with our prior experiments in Chapter 10, we see that the best sets of argumentation features do not generalize across experiments. We hypothesize that argument mining accuracy and interactions between argumentation features and base features determine which classes of argumentation features

Feature set	Set:1→2		Set:2→1	
	κ	qwk	κ	qwk
(Phandi et al., 2015)	–	0.545	–	–
(Dong and Zhang, 2016)	–	0.569	–	–
EASE	0.234	0.585	0.048	0.491
EASE + ARG (All)	**0.298**	**0.622**	**0.049**	**0.493**
EASE + AC	**0.302**	**0.628**	**0.052**	**0.529**
EASE + CL	0.225	**0.589**	**0.049**	**0.493**
EASE + AF	**0.241**	**0.596**	0.041	0.482
EASE + RL	0.230	**0.595**	**0.050**	0.483
EASE + TS	**0.242**	**0.598**	**0.051**	**0.492**
All – AC	**0.261**	**0.610**	0.041	**0.516**
All – CL	**0.271**	**0.596**	0.033	0.456
All – AF	**0.263**	**0.611**	**0.055**	**0.498**
All – RL	**0.311**	**0.626**	0.047	0.471
All – TS	**0.303**	**0.622**	**0.050**	**0.494**
Our best	**0.336**	**0.649**	**0.053**	**0.529**

Table 46: Cross-domain performance of essay score prediction in ASAP data.

are more effective. This suggests that feature selection is a necessary task-specific practice when deploying argument mining for automated essay scoring tasks.

11.4 SUMMARY

The current study expands our research on application of argument mining in automated essay scoring with new data and cross-domain validation. Experiment results confirm again

the value of argumentation features for improving AES performance even when the training and test essays of different writing prompts. This proof is more valuable when argumentation features are evaluated using a real AES system which is one of the most competitive for the studied data. While prior studies explored different machine learning approaches for boosting simple, domain-independent features in cross-domain persuasive essay score prediction (Phandi et al., 2015; Dong and Zhang, 2016), our study addresses the problem by exploiting argument structure of the persuasive essays. Argument structure has been shown to be an effective indicator of persuasive essay quality, which abstracts over the essay content.

12.0 CONCLUSIONS AND DISCUSSIONS

12.1 CONTRIBUTION SUMMARY

In this thesis, we propose context-aware argument mining models that use global and local contextual information to improve state-of-the-art argument mining performance. Our works on argument component identification (Chapters 4 and 5) show that context features that exploit argument indicators and writing topic significantly improve the prediction performance. Our studies on argumentative relation mining (Chapters 6 and 7) investigated features extracted from context segments and achieved significant improvement. Thus, the first contribution of this thesis is the innovative contextual features which were shown to effectively improve argument mining accuracy. Results show that our context-aware argument mining models achieved comparable performance with the state-of-the-art despite the fact that we did not optimize with joint prediction (see §2.3.5). This result makes us believe that we can further increase the state-of-the-art argument mining when our models are optimized with joint prediction.

The second contribution is presented through Chapters 9, 10, and 11 where we conducted extensive studies on the application of argument mining in automated essay scoring. This thesis is the first where argument mining systems are extrinsically compared with respect to AES performance. As expected, our finding support that more accurate argument mining helps predict essay score more accurately. Moreover, our thesis explored a large set of argumentation features and demonstrated that argument mining output can be used to extract features that significantly improve competitive AES models, even when the test essays are of different writing prompts or topic domains.

Both argument mining and persuasive essay score prediction have been studied com-

prehensively in this thesis. Argument mining models were evaluated with different corpora ranging from academic writings by college students to persuasive essays by ESL learners, from high-quality practice writings to real-test essays. Models were also validated in different experimental settings from cross-fold, cross-topic to end-to-end. Regarding automated persuasive essay scoring, the thesis studies different uses of argumentation features in different contexts of base AES models ranging from score classification to score regression, in-domain to cross-domain validations. Our thesis is the first time that application of argument mining in AES is studied in three important perspectives: end-to-end argument mining, improvement to advanced AES model, holistic score of essays. Our thesis brings the strongest demonstration of the values of argument mining in practical AES.

Two other contributions of our thesis are derived from the side works of our research. First, we proposed a novel algorithm to extract argument and domain words semi-automatically from texts. Features computed from those lexicons play vital roles in the success of our context-aware argument mining approaches. Second, we developed an end-to-end argument mining system that can parse persuasive essays to extract argument components and argumentative relations. We are in the process of making the system publicly available for anyone who is interested in argument mining and its application.

12.2 LIMITATIONS AND FUTURE WORK

Despite our great effort to improve argument mining and uncover its potentials in educational application, the thesis still exposes limitations that we hope to resolve in future work.

First of all, our research has not answered how the quality of argument/domain word lexicons might affect the argument mining performance. In Chapter 5, we compared the extracted argument words with the set of argument seed words to test the lexicon coverage. However, such test did not handle the precision of the lexicon. In other words, we did not evaluate the quality of our extracted argument words, and how the argument mining accuracy will be if we adjusted the lexicon extraction algorithm. Second, our lexicon extraction algorithm is semi-supervised in that it needs a set of argument seed words to

start. This makes our algorithm not easy to adapt to other writing genres such as online debates or product reviews. In the future, we first plan to revise the argument and domain word extraction algorithm by first automating the argument keyword selection phrase. One solution is to use argumentative discourse markers to initiate the process. Second, we will compare our lexicon extraction algorithm with other approaches for argument and domain word learning proposed in prior studies (Madnani et al., 2012; Séaghdha and Teufel, 2014) in terms of how the extracted argument/domain words help improve argument mining models. It is also worth measuring the relative precision and coverage of lexicons learned by different approaches to see how their quality impacts argument mining accuracy.

Secondly, there is still room for improvement in window-context features for argumentative relation mining. This thesis has proposed two approaches to create context-windows including window-size heuristics and text segmentation. Experimental results (Table 17) showed that each approach worked better for one of the two data sets. Thus, one question to investigate in future work is whether quality of text segmentation output has impact on the effectiveness of window-context features. To answer this, we plan to manually compare text segmentation output in persuasive essays and academic writings and conduct an error analysis for argumentative relation mining with respect to window-context features. Beside hard-boundary context windows as we have studied, soft-boundary window (i.e., shaped windows) is an interesting idea to explore next. The basic idea is that each context sentence will have a probability of belonging to the context window of an argument component. This membership probability can be inferred from the data and may depend on the distance, discourse relation and content relatedness between the context sentence and the argument components. We see research in probabilistic topic models, text segmentation and discourse parsing are valuable resources for us to develop an approach for this idea.

Our other follow-up plan is a joint model that labels argument components and argumentative relations simultaneously to take advantage of mutual information between the two problems. Joint prediction has shown great successes in argument mining and we believe such a mechanism will further improve our context-aware argument mining models. Also, an interesting direction is to apply deep learning in argument mining. Researchers have used word embeddings as feature vectors for argument component classification. We are thinking

of exploiting deep learning to model argumentative discourse relations in an unsupervised way. With great availability of free texts, we believe to have more than enough pairs of sentences/clauses with explicit discourse markers that likely signal argumentative relations, e.g., because, therefore, but, however. We now need a learning mechanism to create an relation embedding that can represent the deep semantics of the relevance in each pair. For this ambition, we see recent research on sentence embedding, representation learning for discourse parsing as valuable resources.

In parallel with continuously improving argument mining models, deploying argument mining in real-world tasks is also of interest. A possible direction is to use argument mining for a complete assessment of student argumentative writing in the SWoRD peer review system. However, this would need a larger corpus of academic writing annotated for argumentation structures, and also the same or other writings graded by experts. We believe that with larger data sets for argument mining and essays with expert scores, we can have a stronger evidence of the effectiveness of argumentation features for academic essay score prediction.

Finally, expanding our context-aware argument mining research to different writing genres beyond persuasive essays, and with broader concepts of context is our long-term vision. We believe our proposed topic- and window-context features are applicable to wide range of text genres, e.g., online debate and product reviews. In fact, researchers have studied argumentative relations between user comments in a debate but have not yet considered the full potential of discourse structures among comment sentences. Moreover, topic information is generally available in such texts, e.g., debate topics, product keywords, user opinions. Thus, we can adapt our algorithm to extract argument and domain words from texts. On the other hand, the scope of context can be expand to beyond the documents. To identify argument moves in online debates, or evidences in user comments, we should not limit ourself to the textual domain. External contexts such as thread structure, user activity history, and other metadata can be useful. Therefore, an ultimate model should be the one that exploits as much context as the data provides to get the most insights of the data.

APPENDIX A

LISTS OF ARGUMENT WORDS

A.1 ARGUMENT WORDS IN PERSUASIVE ESSAYS

263 argument words extracted from the persuasive development set (6794 essays). Words are stemmed, named entities are replaced by their NER labels. Words are sorted in descending order of their probabilities returned by the LDA topic model.

that the is of it peopl some be to other in are a on as this there for more believ view opinion both howev can with NUMBER than discuss not while have own think an or benefit would should argu may give no conclus advantag agre hand point who which issu could has reason do side argument differ from consid by such way certain fact those topic better say when individu instanc whether exampl abov been posit negat therefor effect much disagre societi clear sinc extent claim disadvantag result will rather moreov obvious far regard drawback nevertheless tend aspect concern still onli seem thus take well consider furthermor might number support strong controversi perspect becom bring hold outweigh case signific lead although benefici experi debat even alway import idea admit impact due base undeni second merit consequ group matter word into addit first come essenti compar henc sever especi wide convinc firm term one major particular doubt sum great evid despit approach up method deni these favor con out role begin anoth obtain each abl mention pros belief wherea influenc besid sens usual varieti phenomenon nowaday less inevit necessari former trend illustr contrari prefer viewpoint often seen rang main conclud befor critic possibl various greater numer plenti assert suitabl encourag oppon valuabl practic potenti vital mean latter opposit analyz crucial meanwhil same advoc accept relat contrast though capabl instead examin aforement enhanc put depend said harm easili turn acquir stand divers definit further accord worth general attent appropri undoubt total pivot effici regardless oppos known appar contend deal remain maintain nonetheless inde absolut

A.2 ARGUMENT WORDS IN ACADEMIC ESSAYS

315 argument words extracted from the academic development set (254 essays). Words are stemmed, named entities are replaced by their NER labels. Words are sorted in descending order of their probabilities returned by the LDA topic model.

> the to of a in and that studi DATE PERSON is this more be on are it as with or NUMBER was by ORGANIZATION they an not will for were research like have would found than when their if also differ there from 's which other at these has result becaus hypothesi observ how find could show been but support can howev anoth whether between what increas import less LOCATION previous may such those then mani predict both suggest conduct look them had hypothes while done base variabl way into all rate about did some question examin focus similar therefor test see determin so specif well compar general expect signific same oppos doe measur often due onli even believ understand order seem consid either set evid high better ani whi lead state possibl rather idea act much ask work given investig although sinc amount shown indic larg actual prior correl thus among say conclud depend come further addit exampl includ SET still play data purpos certain literatur explain involv attempt fact independ life regard overal made common assum natur part though sever design particular opposit form defin frequent main potenti creat just consist build topic answer strong psycholog across relev problem aim turn alway conflict befor tendenc littl great mention simpl evalu own off respect new appear within refer regardless avoid implic chanc exist assess reveal benefit knowledg yet down again she long conclus attribut various normal behind frequenc along necessari appli insight least whole extrem kind one e.g. ad must despit seek manner essenti wide instanc effici propos distinct equal start describ unlik goal probabl sourc combin categori remain obtain enough everyon analyz quick comparison move success confound circumst event impli real togeth limit open util taken statist absenc came reduc infer accur assumpt inclin extens contrari went slight divid ultim perhap inde difficult proven separ final contrast end half too last replic demograph

APPENDIX B

ARGUMENT CODING MANUAL FOR ACADEMIC ESSAYS

B.1 LABEL EXPLANATION

B.1.1 Finding

- Text segment that is a summary, claim or conclusion of/about one or more ideas of the cited study.
- The text must include citation expression inside, and forms a continuous segment covering related content
- If the cited study supports (opposes) the hypothesis then supporting (opposing) idea must be included in the finding.
- Example: "Students who lack academic effort as well as perceive controllability leads to unwillingness to help, anger and neglect (Weiner, 1980)."

B.1.2 Hypothesis

- Do they clearly state at least one hypothesis for their study?
- Hypothesis is expressed in form of a to-be-proven statement, but not a premise statement.
- Example: "Our study predicts that students will provide more positive responses if the email excuse is uncontrollable rather than controllable."

B.1.3 Support

- Does the author cite at least one study that lends support to at least one of their hypotheses?
- The supporting cite must be relevant to the hypothesis.

B.1.4 Opposition

- Does the author cite at least one study that opposes at least one of their hypotheses?
- The opposing cite must be relevant to the hypothesis.

B.1.5 Relevance

- Does the student compare the cited study to his/her own study or to other studies?
- And/or does he/she use the term relevance/relevant?
- The student should compare the ways that the study was conducted, not the results that were found.
- Example: "while they looked at the front half and back half of the classroom, we looked at the classroom in thirds."

B.1.6 A note about idea development

The author can start with a summary of a study and end up with a conclusion. When you identify support/opposition, only need to locate the statement that conveys the supporting/opposing idea.

B.2 CODING PROTOCOL

- Please make sure you annotator strictly follow this protocol when code the data
- Your cooperation is important to evaluate the protocol in our effort of improving annotation quality and coder agreement

- Step 1: Read the attached introduction carefully
  - At this step annotator doesn't need to care about the coding manual.
  - Instead, pay attention to understand the author's hypothesis statement
  - And author's intent of making argument to support and/or oppose his hypothesis.
- Step 2: Start coding sentences of the introduction using the coding file
  - Identify the hypothesis. If no hypotheses are identified, no needs to identify support/opposition sentences.
  - Identify the finding. It's more important to locate the core sentence (i.e., sentence with citation expression) and content-related satellite sentences than the transition sentences
  - Identify the support/opposition sentences. Most of the time, support/opposition sentences are satellite sentences. There however are cases whether orphan sentences (non-satellite, non-core) play support/opposing role.
  - If the study supports/opposes the hypothesis, choose the best sentence(s) that states the ideas: (1) Differentiate idea statement and explanation/elaboration sentences, or (2) Sentences that explain/elaborate the key idea should not be coded as support or opposition sentences.
- Highlight guidelines
  - Hypothesis sentence is not a question sentence.
  - Only mark the hypothesis content.

    Example: "Our hypothesis as a class was that **time of day and gender will not make a difference in the responses of strangers** and our alternative hypothesis Is that **time of day and gender will alter the responses of participants**"
  - Mark all possible study mentions (i.e., citation) no matter which standard they have.

    Example: "Another supporting study was conducted **Rutkowski in 1983** that also demonstrated that with larger groups comes less help for victims in non-emergency situations due to less social pressure **Rutkowski, 1983**."
  - Only mark the support/opposition idea(s), include citation text if necessary.

Example: "One strong study that opposes the bystander effect was done in 1980 by Junji Harada that showed that **increase in group size, even in a face to face proximity, did not decrease the likelihood of being helped** (Harada, 1980)."

APPENDIX C

SAMPLE OUTPUT OF SEGMENTATION ALGORITHM

Gender discrimination is prevalent in varying degrees of severity worldwide. Some countries have a reported lack of gender discrimination but it is difficult for every individual society to remove all gender bias. Some cultures are inherently gender-biased through the use of a gendered language. A study published in October of 2011, researchers found that countries with gendered language exhibit less gender equality than those with gender neutral language. (Prewitt-Freilino, Caswell, & Laasko (2012).) Gendered language can take the form of masculine and feminine verbs in romance languages but in English, a naturally gendered language (Prewitt-Freilino, Caswell, & Laasko (2012).), certain words are given a gender through their continued use in a gender discriminatory way. Gendered language affects how people perceive themselves and how they present themselves to others through the use of language, biased or neutral.

Gendered language affects self-perception beginning at a very young age and carries through to adult life in many people. Gender-biases are highly prevalent in adult society when it comes to self-perception, whether it is division of labor in the home or success and compensation in the workplace. A study published in the European Journal of Social Psychology comparing the femininity and masculinity of someone's actual and ideal selves, found that, regarding professional life, people, both male and female, described their ideal-self being more masculine than their true-self. In the same study, researchers found that in personal relationships people tended to value neutral, or feminine qualities over masculine ones. (DeMarree (2014).)

The traits people assign themselves, whether ideal or true, define who they are and how they describe themselves. Beginning at a young age, each person gathers information about his or her-self based on his or her perceived worth as a person and as a member of a specific group of people or society. Self-esteem is not static and can change on a daily basis. Even something as simple as a person's mood can change how the perceive themself. Although people with both high and low self-esteem rate themselves positively when in a good mood, it only takes a bad mood for someone with low self-esteem to look at themselves negatively. (Brown, & Mankowski (1993).) People with a higher self-esteem are more influenced by extreme or high intensity words. (Bowers (1963).) The dynamic shifts in self-esteem make understanding it and learning to manipulate it so important to allow society to grow in a more positive, self-confident direction.

In the study we conducted in our research methods in psychology class, we wanted to see if people chose words to describe themselves based on the gender identities assigned to them by their biological sex. We predicted that Participants would more strongly endorse gender-biased words to fit the gender-stereotypes society expects them to fit. The second thing we tested was if a participant had high self-esteem, would they more strongly endorse formal words to describe themselves rather than informal counterparts.

APPENDIX D

PREDICTING PEER RATING IN ACADEMIC ESSAYS

D.1 PEER RATING DATA

In Chapters 9 and 10 we showed that argument mining output helps improve persuasive essay score prediction. In this study, we explore an application of argument mining for academic essay scoring. We utilize the academic essay corpus which has been used for our argument mining research (Chapter 3).

The corpus consists of 115 introductions of observational studies written by college students. The essays were submitted to the SWoRD peer review system (Cho and Schunn, 2007) and reviewed by students in the same classes.[1] Student reviewers were asked to provide textual comments and numerical ratings to the papers that they review. The rating rubric is listed in Figure 17. Among 115 essays, we have 113 essays reviewed and graded by student reviewers. Each essay was graded by at least 3 and at most 5 students in scale 1–7. The final score of each essay is a weighted average of peer ratings in which weights indicate rating reliability computed by SWoRD. Although we do not have teacher's grade for the essays, research in peer assessment has shown that peers' grade can be as reliable as teacher's in multiple peer condition (Cho et al., 2006). Thus, our current study uses the weighted average rating of student reviewers as an estimate of essay quality. As shown in Figure 18, the majority of the essays have high score (> 4) and no essay was graded below 2.

[1] https://sword.lrdc.pitt.edu/sword

Consider the following points when giving your rating:

- Central topic introduced and background information provided?
- Brief high-level overview of study design and clear statement of hypotheses?
- Appropriate integration of conflicting research findings into a convincing argument for at least one hypothesis?

Figure 17: Peer rating rubric.

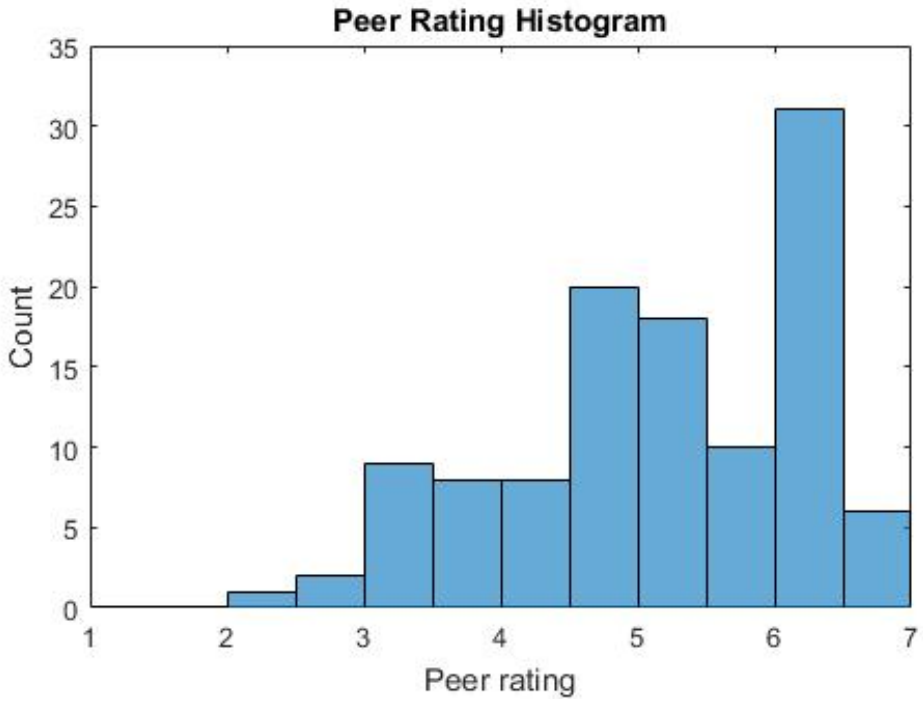

Figure 18: Peer rating histogram.

D.2 ARGUMENTATION FEATURES

We extract the following argumentation features from the essays. Because the rating rubric explicitly asks reviewers to check for presence of opposition findings in the essay, our feature

set emphasizes more on the presence and ratio of opposition findings. Argumentation features are placed in two sets: H (Hypothesis) and F (Finding):

1. H (Hypothesis):
    - Number of hypothesis sentences.
    - Number and percentage of hypotheses that are supported.
    - Number and percentage of hypotheses that are opposed.
    - Number and percentage of hypotheses that are neither support or opposed.
    - Does the essay have at least one hypothesis opposed.
2. F (Finding):
    - Number of finding sentences.
    - Number and percentage of support findings.
    - Number and percentage of opposition findings.
    - Number and percentage of findings that neither support or oppose.
    - Does the essay have at least one finding that opposes

D.3 EXPERIMENT RESULTS

Our current study examines whether argumentation features can help improve a baseline model for peer rating prediction in academic essays. Our baseline model uses solely word and part-of-speech n-grams features which achieves higher performance than the count-based base model in Chapter 10. In particular, we extract 1, 2, 3-grams of tokens and their POS tags, and use one numeric feature to indicate the frequency of corresponding ngram in the essay. We remove ngrams that have less than 5 occurrences in the corpus. We extract argumentation features in two ways: (TRUE) using true labels annotated by experts, and (ARG) predicted labels by our models in Chapters 5 and 7. For ARG extraction method, because we do not have a dedicated data to train our argument mining models, we conduct 10-fold cross validation and take prediction output to extract argumentation features. Given a set of features, peer rating prediction models are trained using LibSVM regresssion algorithm in Weka (Hall et al., 2009).

Table 47 reports 10×10-cross validation performances including Pearson's correlation (cc), mean absolute error (mae), and root-mean-square error ($rmse$). While for the cc measure one should look for higher value, the two error measures are better if lower. Values better than Base are in boldface. Symbol ** and † indicates significantly higher and lower than Base values ($p < 0.01$), respectively. As we can see in the upper half of the table, argumentation features extracted from true labels significantly improved base performance, and the best performance is achieved when adding all argumentation features by true labels to the base model. On the contrary, using all argumentation features extracted from predicted labels significantly degrade performance of the base model. However, F features by predicted labels help improve the baseline. While the results show the value of argumentation features for predicting peer rating in academic essays, it is only true for argumentation features extracted from true labels. This further shows that predicted labels of argumentation structures might not be accurate enough to gain AES improvement.

D.4 DISCUSSIONS

In this experiment, we have showed an application of argument mining in peer rating prediction for academic essays. While the results suggest that argumentation features can help improve the peer rating prediction, there are limitations that prevent us from a strong conclusion of the value of argument mining for AES in academic writings. First of all, we do not have a dedicated training data to develop an end-to-end argument mining model. Second, the annotated data is small which may limit our argument mining accuracy. Third, although peer ratings are usually considered a good estimate of teacher's grades, we cannot conclude the quality of peer rating in our data due to the lack of teacher's grades. In the future, we plan to annotated more data to improve argument mining and apply argumentation features to predict teacher's grades.

Feature sets	*cc*	*mae*	*rmse*
Base	0.408	0.848	1.024
Base + TRUE(H, F)	**0.414****	**0.845** †	**1.021** †
Base + TRUE(H)	**0.411****	**0.846** †	**1.023** †
Base + TRUE(F)	**0.413****	**0.845** †	**1.021** †
Base + ARG(H, F)	0.404 †	0.848	1.027**
Base + ARG(H)	0.403 †	0.850**	1.027**
Base + ARG(F)	**0.409**	**0.846** †	1.024

Table 47: Peer rating prediction performance in academic essays.

APPENDIX E

ESSAY SCORE EXPLANATION BY ARGUMENTATION FEATURES

E.1 INTRODUCTION

Machine learning-based approaches for automatically scoring essays (also written/spoken responses in general) are usually optimized for the best agreement between scores produced by the models and those by human raters. However this process can lend the outcome model to criticism for model validity when its most predictive features fail to represent or interpret the certain basic considerations of the assessment design (Williamson et al., 2012; Bernstein et al., 2010; Ramineni and Williamson, 2013). Different researches have been conducted to build automated essay scoring models that are balanced between performance and validity. Rahimi et al. (2014) designed features for their machine learning model using scoring rubrics. Loukina et al. (2015) evaluated different feature selection methods in terms of how selected features cover criteria identified by a scoring expert.

In previous chapters, we have evaluated impact of argument mining to automatically scoring persuasive essays in terms of scoring performance by different sets of argumentation features. In this study, we evaluate the validity of the argumentation features in terms of how the features explain the essay scores. For this purpose, we use Decision Tree algorithm to build the prediction model because decision tree models are easy to visualize, interpret and explain for how feature values separate classes.

Moreover, to set-up a reference standard for decision rules in the tree models, we consider

Score	Task description
5	• Is well organized and well developed, using clearly appropriate explanations, exemplifications and/or details
4	• Addresses the topic and task well, though some points may not be fully elaborated • Is generally well organized and well developed, using appropriate and sufficient explanations, exemplifications and/or details
3	• Addresses the topic and task using somewhat developed explanations, exemplifications and/or details
2	• Inappropriate or insufficient exemplifications, explanations or details to support or illustrate generalizations in response to the task
1	• Little or no detail, or irrelevant specifics, or questionable responsiveness to the task
0	• Merely copies words from the topic, rejects the topic, or is otherwise not connected to the topic

Table 48: TOEFL iBT Independent Writing Rubrics

the TOEFL iBT Independent Writing Rubrics[1] which were used to grade essays of the TOEFL11 corpus (Blanchard et al., 2013). Table 48 shows part of the scoring guidelines for TOEFL essays. To make this reference more relevant to the decision rules learned with the argumentation features, we keep only rubric statements that are related to topic development and response elaboration, but ignore those about organization and language usage. For the complete rubrics, one can refer the link provided. Although the scoring guidelines were designed for score range [0, 5] while essays in our data have scored categorized to levels a, b, c, the guidelines are still applicable to our study because the score levels a, b, c were derived consistently from the raw numerical scores (Blanchard et al., 2013).

Data used in this study is the set of 107 TOEFL essays which were annotated for argumentation structure (see Chapter 9). Decision tree models are trained with argumentation features as extracted from (1) true labels of argumentation structures (referred to as TrueLabel tree models), and (2) predicted labels of our argument mining pipeline (referred to as ArgN tree models), respectively. Data statistics and list of argumentation features are reported in Tables 31 and 36.

[1] http://www.ets.org/s/toefl/pdf/toefl_writing_rubrics.pdf

	AC	CL	AF	RL	TS	All
κ	0.446	0.569	0.260	0.216	0.360	0.550
qwk	0.708	0.768	0.485	0.465	0.540	0.742

Table 49: 10-fold cross validation performance with Decision Tree algorithm in TE107 data. Argumentation features are extracted from true labels.

E.2 ARGUMENTATION FEATURES FROM TRUE LABELS

For each set of argumentation features, i.e., argument component (AC), component label (CL), argument flow (AF), relation label (RL), and typology structure (TS), Table 49 shows 10-fold cross validation performance with the Decision Tree classifier implementation in Scikit-learn[2]. Comparing with Logistic Regression algorithm (see Table 37), while Decision Tree yielded lower score prediction performance for each argumentation feature set, the algorithm obtained better κ and qwk than Logistic Regression when trained with all argumentation features. One reason is that Decision Tree algorithm is capable of pruning ineffective features which do not help further classify data. Unfortunately, the pruning capability of Decision Tree algorithm does not always yield to optimal feature set. An evidence is that 10-fold cross validation performance with all argumentation feature is lower then the performance with component label features.

To illustrate the feature importance, Figure 19 visualizes the Decision Tree trained with 107 essays using all argumentation features. Five features that show up in the tree include:

1. WordInArgument: number of words in argument components (AC)
2. SentencewArgument: number of sentences that have AC
3. danglingClaim: number of claims that have no support premises
4. SentencewArgumentPct: percentage of argumentative sentences
5. PremisePct: percentage of premises

[2]scikit-learn.org

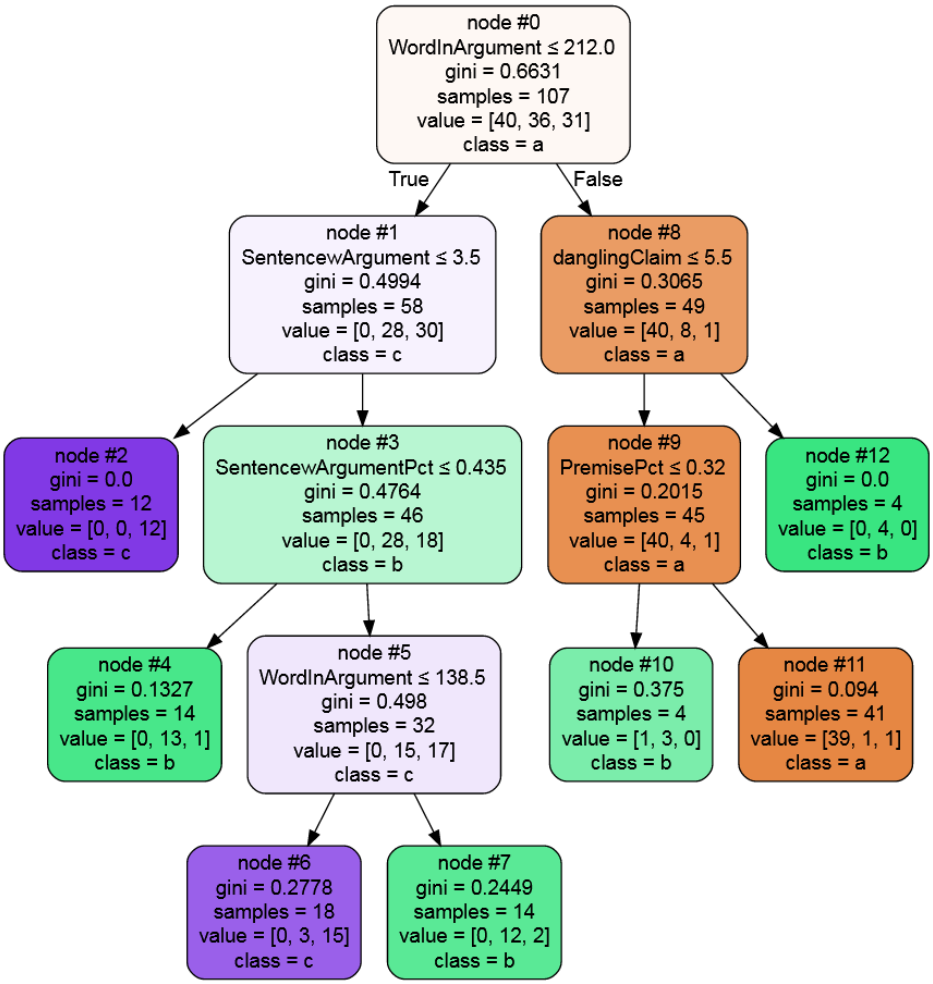

Figure 19: Decision tree learned using argumentation features with true labels

Each node of the tree has darker color if the distribution of its data is more skewed to the major class, and includes the following content in top-down order:

- The condition that split its data. For example, the right branch from node #0 says that essays with more than 212 words in AC can have scores either b or a. However, essays with less than 212 words in AC have scores c or b.
- Gini impurity score that measures the error probability of a random labeling given the distribution of labels in the subset (Breiman et al., 1984). The splitting conditions that yield small gini scores are desired.

Set	Feature	Short name
AC	Number of words in AC	WordInArgument
	Number of argumentative sentences	SentencewArgument
	Percentage of argumentative sentences	SentencewArgumentPct
CL	Number of premises	Premise
	Premise percentage	PremisePct
	Number of claims	Claim
AF	Number of paragraphs with claim and premise	ParagraphwClaim-Premise
	Percentage of typed bigram MajorClaim-Claim	MajorClaim-Claim
	Percentage of typed bigram Claim-Claim	Claim-Claim
	Percentage of type bigram Premise-Claim	Premise-Claim
RL	Number of supporting premises	supportPremise
	Number of dangling claims	danglingClaim
TS	Number of paragraphs that have chain arguments	pwChainTopo
	Number of tree arguments	treeTopo

Table 50: Most important features of each feature set

- Total samples of the subset, e.g., node #1 contains 58 essays.
- Class distribution of the subset. For instance, node #2 has all 12 essays of score c and no essays of scores a or b.
- The major class, e.g., score b in node #3.

As shown in the figure, branches from nodes #0 and #5 generalize a rule that essays with more words in argument components (AC) tend to have higher scores than those with less words. Among essays with more words in AC, the number of dangling claims (node #8) and percentage of premises (node #9) further refines essay scores. To get scores of a, essays should not have many dangling claim (e.g., more than 5) and small percentage of premise (e.g., less than 0.3). While the conditions that formulate this decision tree may be specific to the training data, e.g., WordInArgument ≤ 212, the generalized rules instantiate well rubric statements in Table 48, especially the rules of dangling claims and percentage of premises.

To examine feature importance in each argumentation feature set, Figures 20, 21, 22, 23, and 24 visualize the decision trees learned with argumentation features of each set, respectively. Table 50 shows important features which show up in the learned decision trees. Over the five trees, we consider the leaf nodes that have the smallest gini scores and obtain

the following rules for each of the score levels. By only considering the leaf nodes with the smallest gini scores, we aim for the most reliable decision rules to validate the argumentation features that show up.

- Score c:
  - WordInArgument $\leq$ 212 AND SentencewArgument $\leq$ 3.5 (Figure 20, gini = 0)
  - ParagraphwClaim-Premise $\leq$ 1.5 AND MajorClaim-Claim $>$ 0.415 (Figure 22, gini = 0)
- Score b:
  - WordInArgument $\leq$ 212 AND SentencewArgument $>$ 3.5 AND SentencewArgumentPct $\leq$ 0.435 (Figure 20, gini = 0.1327)
- Score a:
  - Premise $>$ 4.5 AND PremisePct $>$ 0.345 AND Claim $\leq$ 9.5 (Figure 21, gini = 0.0986)

First, we can see that decision rules for score c and a have very low gini score and are consistent with the writing rubrics. One of the rules for score c states that essays that have one or no paragraph with claim and premise, but a high ratio of major claim – claim chain will have score c. On the other hand, essays that have many premises (more than 4) but not too many claims (less than 9) will have score a. However, rules for score b have higher gini and their clauses are contradictory. As stated in the rule above, essays that have less words in argument components (less than 212), but not too few or too many argumentative sentences will have scores of b. The conflicting clauses of that rule and also many other rules of score b (e.g., node #7 in Figure 21) may reveal the challenges of classifying this score level which is considered more ambiguous than the levels a and c.

Overall, the results show that the extracted rules from decision tree models trained with argumentation features align well with the reference writing rubrics. This is expected because the decision tree models were trained with argumentation features derived from true labels of argumentation features. In the next experiment, we study argumentation features in the case they are computed from predicted labels of argumentation features.

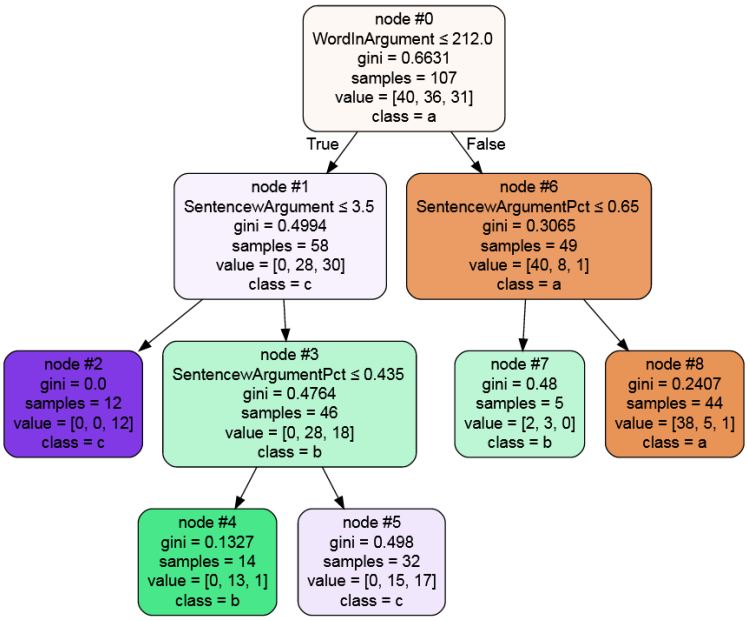

Figure 20: Decision tree learned with TrueLabel AC features

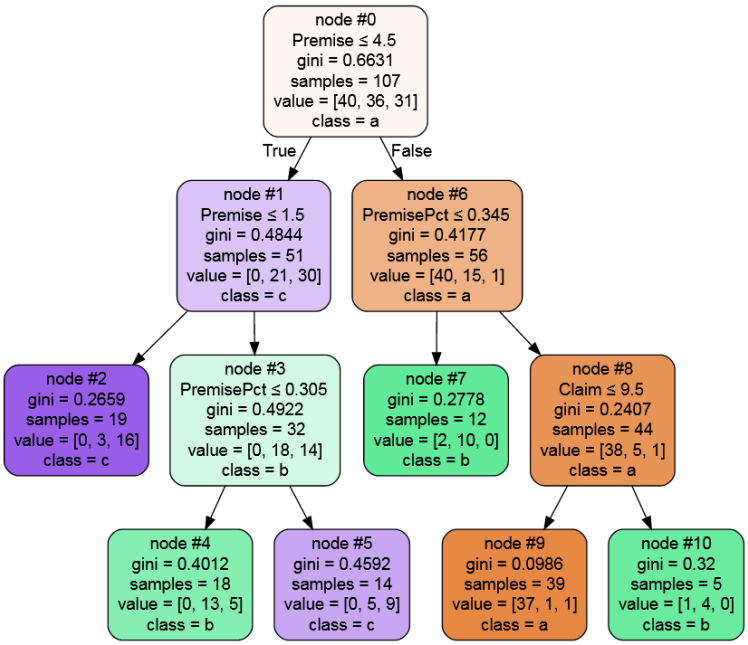

Figure 21: Decision tree learned with TrueLabel CL features

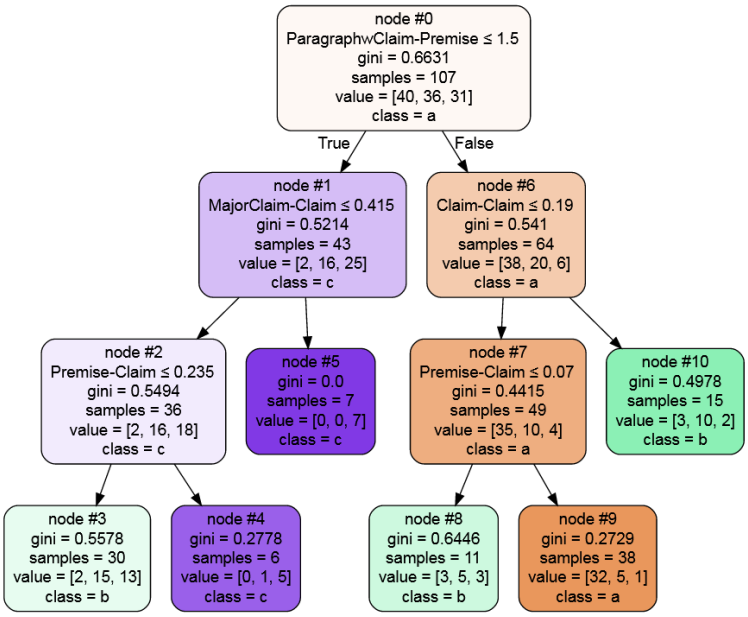

Figure 22: Decision tree learned with TrueLabel AF features

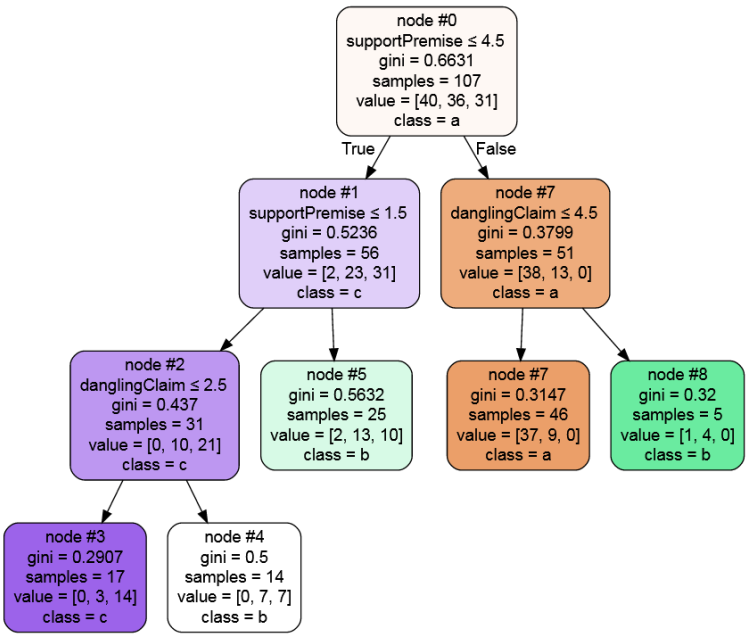

Figure 23: Decision tree learned with TrueLabel RL features

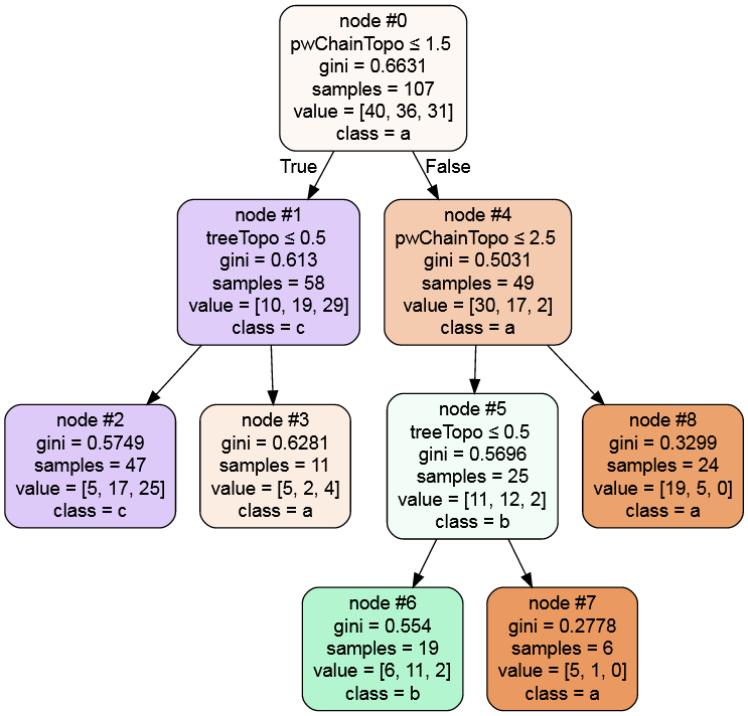

Figure 24: Decision tree learned with TrueLabel TS features

E.3 ARGUMENTATION FEATURES FROM PREDICTED LABELS

We replicate the experiment in §9.5.2 but use Decision Tree algorithm to learn the prediction model. In particular, our end-to-end argument mining pipeline (see Chapter 8) was used to first identify argument component, then classify components by their argumentative roles and determine if each pair of components holds a support relation. 10-fold cross validation performance are reported in Table 51.

Comparing to the score prediction results when argumentation features are computed from true argument labels (Table 49), the automated scoring performance are significantly worse with argumentation features from predicted argument labels. We, however, still observe that using all argumentation features yielded better performance than each of feature sets.

Regarding feature importance, we visualize ArgN decision trees which are learned with

	AC	CL	AF	RL	TS	All
κ	0.388	0.285	0.185	0.082	0.0	0.456
qwk	0.600	0.556	0.348	0.186	0.0	0.672

Table 51: 10-fold cross validation performance with Decision Tree algorithm in TE107 data. Argumentation features are extracted from predicted labels.

AC and All feature sets, and compare the decision rules with those of the corresponding TRUELABEL decision trees. We observe that at high-level ARGN AC and CL decision tree models perform similarly as the corresponding TRUELABEL models. In particular, WordInArgument is the most important feature and essays with more words in argument components usually have high scores, e.g., nodes #0, #2 and #6 in Figure 25. Number of argumentative sentences helps further refine essay scores, e.g., node #7. Number of premises is the most important feature of CL set and essays with more premises have higher scores, e.g., nodes #0, #6 in Figure 26. However, decision rules learned with ARGN features generally have higher gini scores than the rules learned with TRUELABEL features. These reflect the lower essay score prediction performance of ARGN features than TRUELABEL features.

Examining the ARGN tree models learned with AF, RL and TS feature sets, we see that decision rules are more conflicting with the scoring guidelines. This could be due to the very low score prediction performance of those tree models. For example, node #2 in Figure 27 says that essays with no dangling claim have score c but with one or more dangling claims have score a. Node #0 and #1 make a rule that essays with less then 5 supporting premises but more than one supported claim have score c.

In conclusion, while ARGN AC and CL features could yield decision rules that align with the writing rubrics, the remaining ARGN features perform much worse and their learned decision tree models are more conflicting.

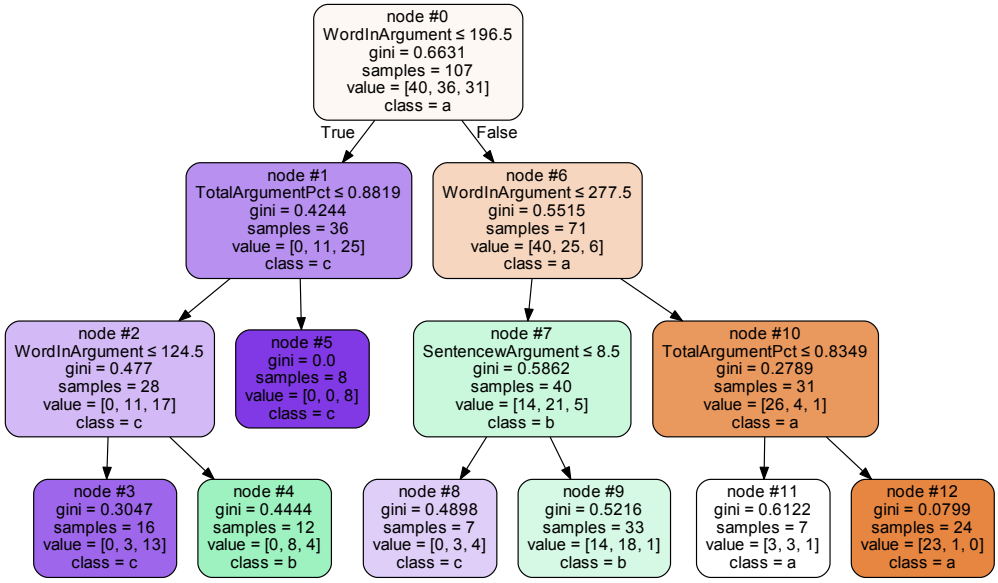

Figure 25: Decision tree learned with ArgN AC features

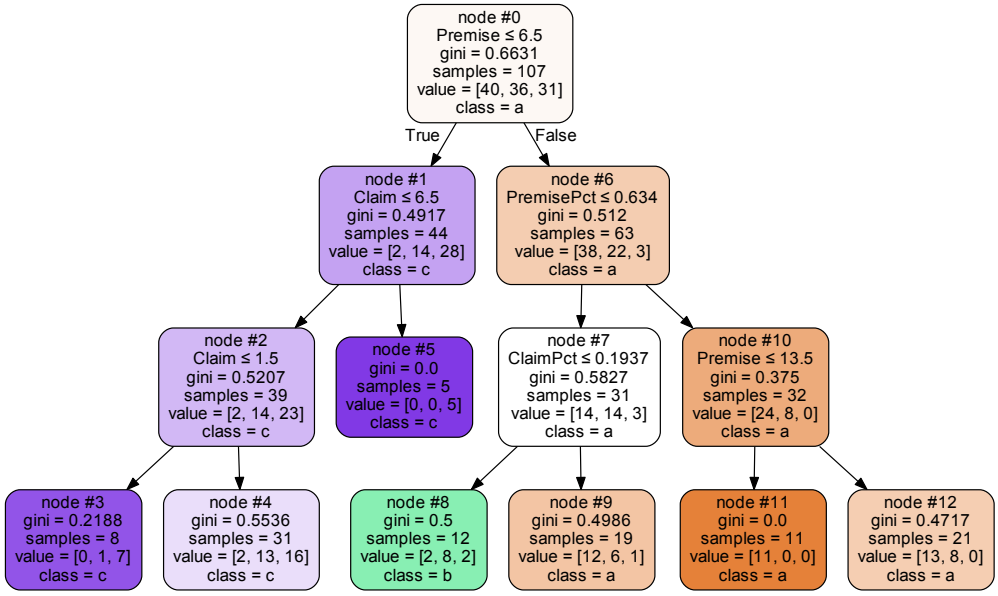

Figure 26: Decision tree learned with ArgN CL features

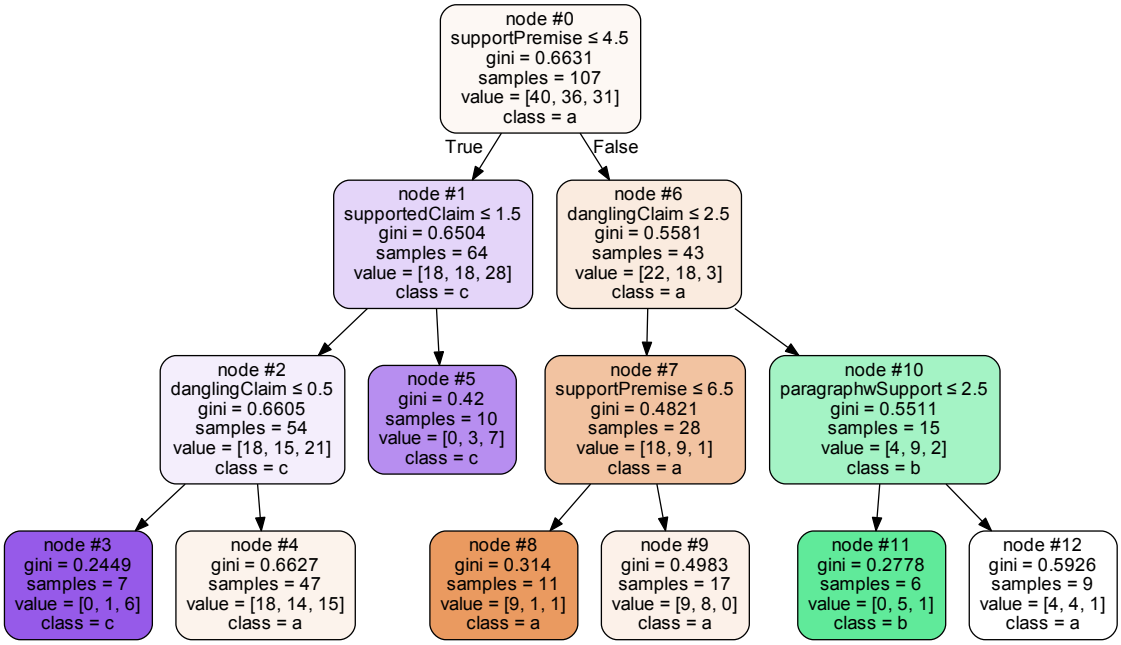

Figure 27: Decision tree learned with ArgN RL features

APPENDIX F

IMPACT OF INTERMEDIATE SCORE RANGE IN CROSS-DOMAIN ESSAY SCORE PREDICTION

In cross-domain essay score prediction, when the source and the target domains have different score ranges (chapter 11), re-scaling is required to convert the regression output into the target score range. This brings up the questions whether to use an intermediate score range, and how to determine such range for the best performance. We re-investigate the cross-domain AES in chapter 11 but with different intermediate score ranges. For each choice of intermediate score range, we compare EASE model with EASE augmented with argumentation features. Results are shown in Table 52.

The intermediate range [0, 1] was used in (Dong and Zhang, 2016). However, this intermediate range does not work for the EASE system in our study, which may be due to the choice of learning algorithm in EASE.

With no intermediate score ranges (i.e., direct scaling), AES models are trained with the original score of training essays, and regression output is scaled directly to score range of target essays. While Set:2→1 has the best performance using direct scaling, Set:1→2 achieves the highest κ and qwk with range [-1, 1]. Moving from small to large ranges, Set:1→2 has performance decrease but Set:2→1 has performance increase.

These results show that choosing the intermediate score range for cross-domain AES is not trivial and dependent factors may include characteristics of the learning algorithm, original scores, and target scores.

Inter. range	Feature set	Set:1→2		Set:2→1	
		κ	qwk	κ	qwk
[0, 1]	EASE	0.000	0.091	0.000	0.003
	EASE + ARG	0.000	0.106	0.000	0.004
[-1, 1]	EASE	0.234	0.585	0.048	0.491
	EASE + ARG	0.298	0.622	0.049	0.493
[-3, 3]	EASE	0.156	0.547	0.249	0.790
	EASE + ARG	0.185	0.565	0.274	0.792
No	EASE	0.016	0.436	0.291	0.809
	EASE + ARG	0.025	0.431	0.289	0.810

Table 52: Cross-domain performance of essay score prediction in ASAP data with different intermediate score ranges.

BIBLIOGRAPHY

Bansal, A., Bu, Z., Mishra, B., Wang, S., Ashley, K., and Grabmair, M. (2016). Document Ranking with Citation Information and Oversampling Sentence Classification in the LUIMA Framework. In *Legal Knowledge and Information Systems: JURIX 2016: The Twenty-Ninth Annual Conference*, volume 294, pages 33–42. IOS Press.

Barstow, B., Schunn, C., Fazio, L., Falakmasir, M., and Ashley, K. (2015). Improving Science Writing in Research Methods Classes Through Computerized Argument Diagramming. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, Pasadena, California.

Belz, A. and Gatt, A. (2008). Intrinsic vs. Extrinsic Evaluation Measures for Referring Expression Generation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 197–200, Columbus, Ohio. Association for Computational Linguistics.

Bench-Capon, T. J. and Dunne, P. E. (2007). Argumentation in artificial intelligence. *Artificial intelligence*, 171(10-15):619–641.

Bentahar, J., Moulin, B., and Bélanger, M. (2010). A Taxonomy of Argumentation Models Used for Knowledge Representation. *Artif. Intell. Rev.*, 33(3):211–259.

Bernstein, J., Moere, A. V., and Cheng, J. (2010). Validating automated speaking tests. *Language Testing*, 27(3):355–377.

Besnard, P., Garcia, A., Hunter, A., Modgil, S., Prakken, H., Simari, G., and Toni, F. (2014). Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4.

Besnard, P. and Hunter, A. (2008). *Elements of Argumentation*. MIT Press.

Biran, O. and Rambow, O. (2011). Identifying Justifications in Written Dialogs by Classifying Text as Argumentative. *International Journal of Semantic Computing*, 5(4):363–381.

Blanchard, D., Tetreault, J., Higgins, D., Cahill, A., and Chodorow, M. (2013). TOEFL11: A Corpus of Non-native English. *ETS Research Report Series*, 2013(2):i–15.

Blei, D. M. (2012). Probabilistic Topic Models. *Commun. ACM*, 55(4):77–84.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

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.

Boyd-Graber, J., Blei, D., and Zhu, X. (2007). A Topic Model for Word Sense Disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1024–1033, Prague, Czech Republic. Association for Computational Linguistics.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees*. The Wadsworth statistics/probability series. Wadsworth International Group.

Brody, S. and Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812. Association for Computational Linguistics.

Burstein, J., Chodorow, M., and Leacock, C. (2004). Automated essay evaluation: The Criterion online writing service. *AI Magazine*, 25:27–36.

Burstein, J., Marcu, D., and Knight, K. (2003). Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, 18(1):32–39.

Cabrio, E., Tonelli, S., and Villata, S. (2013). From Discourse Analysis to Argumentation Schemes and Back: Relations and Differences. In *Computational Logic in Multi-Agent Systems*, volume 8143 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg.

Cabrio, E. and Villata, S. (2012). Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interactions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 208–212, Stroudsburg, PA, USA. Association for Computational Linguistics.

Carlson, L., Marcu, D., and Okurowski, M. E. (2001). Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, SIGDIAL '01, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27:1–27:27.

Chiu, B., Korhonen, A., and Pyysalo, S. (2016). Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

Cho, K. and Schunn, C. D. (2007). Scaffolded Writing and Rewriting in the Discipline: A Web-based Reciprocal Peer Review System. *Computers & Education*, 48(3):409–426.

Cho, K., Schunn, C. D., and Wilson, R. W. (2006). Validity and reliability of scaffolded peer assessment of writing from instructor and student perspectives. *Journal of Educational Psychology*, 98(4):891.

Dikli, S. (2006). An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*, 5(1).

Dong, F. and Zhang, Y. (2016). Automatic Features for Essay Scoring – An Empirical Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, Austin, Texas. Association for Computational Linguistics.

Du, J., Jiang, J., Yang, L., Song, D., and Liao, L. (2014). Shell Miner: Mining Organizational Phrases in Argumentative Texts in Social Media. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, pages 797–802, Washington, DC, USA. IEEE Computer Society.

Dung, P. M. (1995). On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-person Games. *Artif. Intell.*, 77(2):321–357.

Eisenstein, J. and Barzilay, R. (2008). Bayesian Unsupervised Topic Segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 334–343, Stroudsburg, PA, USA. Association for Computational Linguistics.

Falakmasir, M. H., Ashley, K., Schunn, C., and Litman, D. (2014). Identifying Thesis and Conclusion Statements in Student Essays to Scaffold Peer Review. In *Intelligent Tutoring Systems*, volume 8474 of *Lecture Notes in Computer Science*, pages 254–259. Springer International Publishing.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A Library for Large Linear Classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Feng, V. W. and Hirst, G. (2011). Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 987–996. Association for Computational Linguistics.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Freeley, A. and Steinberg, D. (2008). *Argumentation and Debate*. Cengage Learning.

Freeman, J. B. (1991). *Dialectics and the Macrostructure of Arguments: A Theory of Argument Structure*. Foris Publications.

Funatsu, T., Tomiura, Y., Ishita, E., and Furusawa, K. (2014). Extracting Representative Words of a Topic Determined by Latent Dirichlet Allocation. In *eKNOW 2014, The Sixth International Conference on Information, Process, and Knowledge Management*, pages 112–117.

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.

Goudas, T., Louizos, C., Petasis, G., and Karkaletsis, V. (2014). Argument Extraction from News, Blogs, and Social Media. In *Artificial Intelligence: Methods and Applications*, volume 8445 of *Lecture Notes in Computer Science*, pages 287–299. Springer International Publishing.

Govier, T. (2013). *A practical study of argument*. Cengage Learning, 7th edition.

Grabmair, M., Ashley, K. D., Chen, R., Sureshkumar, P., Wang, C., Nyberg, E., and Walker, V. R. (2015). Introducing LUIMA: An Experiment in Legal Conceptual Retrieval of Vaccine Injury Decisions Using a UIMA Type System and Tools. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, ICAIL '15, pages 69–78, New York, NY, USA. ACM.

Granger, S., Dagneaux, E., Meunier, F., and Paquot, M. (2009). *International Corpus of Learner English v2*. Presses universitaires de Louvain, Louvain-la-Neuve.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Griffiths, T. L., Steyvers, M., Blei, D. M., and Tenenbaum, J. B. (2005). Integrating Topics and Syntax. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press.

Guo, Y., Korhonen, A., Liakata, M., Silins, I., Sun, L., and Stenius, U. (2010). Identifying the Information Structure of Scientific Abstracts: An Investigation of Three Different Schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 99–107, Uppsala, Sweden. Association for Computational Linguistics.

Haghighi, A. and Vanderwende, L. (2009). Exploring Content Models for Multi-document Summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*,

NAACL '09, pages 362–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

Hirohata, K., Okazaki, N., Ananiadou, S., and Ishizuka, M. (2008). Identifying Sections in Scientific Abstracts using Conditional Random Fields. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pages 381–388.

Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 289–296, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42(1):177–196.

Ji, Y. and Eisenstein, J. (2014). Representation Learning for Text-level Discourse Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Jo, Y. and Oh, A. H. (2011). Aspect and Sentiment Unification Model for Online Review Analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA. ACM.

Klebanov, B. B., Stab, C., Burstein, J., Song, Y., Gyawali, B., and Gurevych, I. (2016). Argumentation: Content, Structure, and Relationship with Essay Quality. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 70–75, Berlin, Germany. Association for Computational Linguistics.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Knott, A. and Dale, R. (1994). Using linguistic phenomena to motivate a set of coherence relations. *Discourse Processes*, 18(1):35–62.

Kohavi, R. (1995). A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Krippendorff, K. (1980). *Content analysis: An introduction to its methodology*. Sage.

Krippendorff, K. (2004). Measuring the Reliability of Qualitative Text Analysis Data. *Quality and Quantity*, 38(6):787–800.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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.

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.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM.

Lin, J., Karakos, D., Demner-Fushman, D., and Khudanpur, S. (2006). Generative Content Models for Structural Analysis of Medical Abstracts. In *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, BioNLP '06, pages 65–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2014). A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Lippi, M. and Torroni, P. (2015). Argument mining: a machine learning perspective. Buenos Aires, Argentina.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool.

Louis, A. and Nenkova, A. (2013). What Makes Writing Great? First Experiments on Article Quality Prediction in the Science Journalism Domain. *Transactions of the Association of Computational Linguistics*, 1:341–352.

Loukina, A., Zechner, K., Chen, L., and Heilman, M. (2015). Feature selection for automated speech scoring. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–19, Denver, Colorado. Association for Computational Linguistics.

Madnani, N., Heilman, M., Tetreault, J., and Chodorow, M. (2012). Identifying High-Level Organizational Elements in Argumentative Discourse. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 20–28, Montrèal, Canada. Association for Computational Linguistics.

Magnini, B., Zanoli, R., Dagan, I., Eichler, K., Neumann, G., Noh, T.-G., Padó, S., Stern, A., and Levy, O. (2014). The Excitement Open Platform for Textual Inferences. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 43–48, Baltimore, Maryland. Association for Computational Linguistics.

Mcauliffe, J. D. and Blei, D. M. (2008). Supervised topic models. In *Advances in neural information processing systems*, pages 121–128.

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 171–180, New York, NY, USA. ACM.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Mochales, R. and Moens, M.-F. (2008). Study on the Structure of Argumentation in Case Law. In *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference*, pages 11–20, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Mochales, R. and Moens, M.-F. (2011). Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Moens, M.-F., Boiy, E., Palau, R. M., and Reed, C. (2007). Automatic Detection of Arguments in Legal Texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ICAIL '07, pages 225–230, New York, NY, USA. ACM.

Navigli, R. (2009). Word Sense Disambiguation: A Survey. *ACM Computing Surveys (CSUR)*, 41(2):10:1–10:69.

Newell, G. E., Beach, R., Smith, J., and VanDerHeide, J. (2011). Teaching and Learning Argumentative Reading and Writing: A Review of Research. *Reading Research Quarterly*, 46(3):273–304.

Newman, D., Chemudugunta, C., Smyth, P., and Steyvers, M. (2006). Analyzing Entities and Topics in News Articles Using Statistical Topic Models. In *Proceedings of the 4th IEEE International Conference on Intelligence and Security Informatics*, ISI'06, pages 93–104, Berlin, Heidelberg. Springer-Verlag.

Nguyen, H. and Litman, D. (2015). Extracting Argument and Domain Words for Identifying Argument Components in Texts. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 22–28, Denver, CO. Association for Computational Linguistics.

Nguyen, H. and Litman, D. (2016a). Context-aware Argumentative Relation Mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1137, Berlin, Germany. Association for Computational Linguistics.

Nguyen, H. and Litman, D. (2016b). Improving argument mining in student essays by learning and exploiting argument indicators versus essay topics. In *Proceedings 29th International FLAIRS Conference*, Key Largo, FL.

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.

Palau, R. M. and Moens, M.-F. (2009). Argumentation Mining: The Detection, Classification and Structure of Arguments in Text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 98–107, New York, NY, USA. ACM.

Park, J. and Cardie, C. (2014). Identifying Appropriate Support for Propositions in Online User Comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

Peldszus, A. (2014). Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the First Workshop on Argumentation Mining*, pages 88–97, Baltimore, Maryland. Association for Computational Linguistics.

Peldszus, A. and Stede, M. (2013). From Argument Diagrams to Argumentation Mining in Texts: A Survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.

Peldszus, A. and Stede, M. (2015). Joint prediction in MST-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 938–948, Lisbon, Portugal. Association for Computational Linguistics.

Persing, I. and Ng, V. (2013). Modeling Thesis Clarity in Student Essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, Sofia, Bulgaria. 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*, pages 543–552, Beijing, China. Association for Computational Linguistics.

Persing, I. and Ng, V. (2016). End-to-End Argumentation Mining in Student Essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.

Phan, X.-H. and Nguyen, C.-T. (2007). GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA). Technical report, Technical report.

Phandi, P., Chai, K. M. A., and Ng, H. T. (2015). Flexible Domain Adaptation for Automated Essay Scoring Using Correlated Linear Regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Lisbon, Portugal. Association for Computational Linguistics.

Pitler, E., Louis, A., and Nenkova, A. (2009). Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 683–691. 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). ACL Anthology Identifier: L08-1093.

Qazvinian, V. and Radev, D. R. (2010). Identifying Non-explicit Citing Sentences for Citation-based Summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 555–564, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rahimi, Z., Litman, D., Correnti, R., Matsumura, L., Wang, E., and Kisa, Z. (2014). Automatic Scoring of an Analytical Response-To-Text Assessment. In *Intelligent Tutoring Systems*, volume 8474 of *Lecture Notes in Computer Science*, pages 601–610. Springer International Publishing.

Ramineni, C. and Williamson, D. M. (2013). Automated essay scoring: Psychometric guidelines and practices. *Assessing Writing*, 18(1):25 – 39.

Séaghdha, D. . and Teufel, S. (2014). Unsupervised learning of rhetorical structure with un-topic models. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING-14)*, Dublin, Ireland.

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.

Sarić, F., Glavaš, G., Karan, M., Šnajder, J., and Dalbelo Bašić, B. (2012). TakeLab: Systems for Measuring Semantic Text Similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 441–448, Montréal, Canada. Association for Computational Linguistics.

Sergeant, A. (2013). Automatic argumentation extraction. In *The 10th European Semantic Web Conference*, pages 656–660, Montpellier, France. Springer.

Shermis, M. D. and Burstein, J. (2013). *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Somasundaran, S. and Wiebe, J. (2009). Recognizing Stances in Online Debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 226–234, Suntec, Singapore. Association for Computational Linguistics.

Song, Y., Heilman, M., Beigman Klebanov, B., and Deane, P. (2014). Applying Argumentation Schemes for Essay Scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 69–78, Baltimore, Maryland. Association for Computational Linguistics.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics.

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. (2017). Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, 43(3):619–659.

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 Cabrio, E., Villata, S., and Wyner, A., editors, *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*, pages 40–49, Bertinoro, Italy. CEUR-WS.

Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440.

Taghipour, K. and Ng, H. T. (2016). A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). Sharing Clusters among Related Groups: Hierarchical Dirichlet Processes. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 1385–1392. MIT Press.

Teufel, S. and Moens, M. (2002). Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28(4).

Teufel, S., Siddharthan, A., and Batchelor, C. (2009). Towards Discipline-independent Argumentative Zoning: Evidence from Chemistry and Computational Linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3*, EMNLP '09, pages 1493–1502, Stroudsburg, PA, USA. Association for Computational Linguistics.

Toulmin, S. E. (1958). *The uses of argument*. Cambridge University Press Cambridge.

Van Eemeren, F. H. and Grootendorst, R. (1982). The speech acts of arguing and convincing in externalized discussions. *Journal of Pragmatics*, 6(1):1 – 24.

Varga, A., Preotiuc-Pietro, D., and Ciravegna, F. (2012). Unsupervised document zone identification using probabilistic graphical models. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

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.

Wallach, H. M., Mimno, D. M., and McCallum, A. (2009). Rethinking LDA: Why Priors Matter. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. Curran Associates, Inc.

Walton, D., Reed, C., and Macagno, F. (2008). *Argumentation Schemes*. Cambridge University Press.

Wang, J. and Lan, M. (2015). A Refined End-to-End Discourse Parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24, Beijing, China. Association for Computational Linguistics.

Williamson, D. M., Xi, X., and Breyer, F. J. (2012). A Framework for Evaluation and Use of Automated Scoring. *Educational Measurement: Issues and Practice*, 31(1):2–13.

Zhao, W. X., Jiang, J., Yan, H., and Li, X. (2010). Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 56–65, Stroudsburg, PA, USA. Association for Computational Linguistics.