A Machine-Learning-Based Pipeline Approach to Automated Fact-Checking

Hanselowski, Andreas

(2020)

DOI (TUprints): https://doi.org/10.25534/tuprints-00014136

Publikationstyp: Dissertation

Fachbereich: 20 Fachbereich Informatik

Quelle des Originals: https://tuprints.ulb.tu-darmstadt.de/14136

A Machine-Learning-Based Pipeline Approach to Automated Fact-Checking

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades Dr. rer. nat.

vorgelegt von
Dr.-Ing. Andreas Hanselowski
geboren in Sokuluk

Tag der Einreichung:	17. März 2020
Tag der Disputation:	14. Mai 2020
Referenten:	Prof. Dr. Iryna Gurevych, Darmstadt Prof. Chris Reed, Dundee, U.K.

Darmstadt 2020
D17

Hanselowski, Andreas:
A Machine-Learning-Based Pipeline Approach to Automated Fact-Checking
Darmstadt, Technische Universität Darmstadt,
Jahr der Veröffentlichung der Dissertation auf TUprints: 2020
URN: urn:nbn:de:tuda-tuprints-141365
Tag der mündlichen Prüfung: 14.05.2020

To my beloved wife, Lidiia

Wissenschaftlicher Werdegang des Verfassers[1]

09/06–08/09 Bachelor Studium (B.Eng.) in Maschinenbau an der Hochschule für Technik, Wirtschaft und Gestaltung (HTWG) Konstanz

09/09–11/11 Master Studium (M.Sc.) in Computational Mechanics of Materials and Structures (COMMAS) an der Universität Stuttgart

12/11–09/16 Promotion am Institut für Technische und Numerische Mechanik (ITM) an der Universität Stuttgart zum Dr.-Ing.

10/16–09/19 Promotion am Ubiquitous Knowledge Processing (UKP) Lab an der Technischen Universität Darmstadt zum Dr. rer. nat.

[1] Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

Abstract

In the past couple of years, there has been a significant increase of the amount of false information on the web. The falsehoods quickly spread through social networks reaching a wider audience than ever before. This poses new challenges to our society as we have to reevaluate which information source we should trust and how we consume and distribute content on the web. As a response to the rising amount of disinformation on the Internet, the number of fact-checking platforms has increased. On these platforms, professional fact-checkers validate the published information and make their conclusions publicly available. Nevertheless, the manual validation of information by fact-checkers is laborious and time-consuming, and as a result, not all of the published content can be validated. Since the conclusions of the validations are released with a delay, the interest in the topic has often already declined, and thus, only a small fraction of the original news consumers can be reached.

Automated fact-checking holds the promise to address these drawbacks as it would allow fact-checkers to identify and eliminate false information as it appears on the web and before it reaches a wide audience. However, despite significant progress in the field of automated fact-checking, substantial challenges remain: (i) The datasets available for training machine learning-based fact-checking systems do not provide high-quality annotation of real fact-checking instances for all the tasks in the fact-checking process. (ii) Many of today's fact-checking systems are based on knowledge bases that have low coverage. Moreover, because for these systems sentences in natural language need to be transformed into formal queries, which is a difficult task, the systems are error-prone. (iii) Current end-to-end trained machine learning systems can process raw text and thus, potentially harness the vast amount of knowledge on the Internet, but they are intransparent and do not reach the desired performance. In fact, fact-checking is a challenging task and today's machine learning approaches are not mature enough to solve the problem without human assistance. In order to tackle the identified challenges, in this thesis, we make the following contributions:

(1) We introduce a new corpus on the basis of the Snopes fact-checking website that contains real fact-checking instances and provides high-quality annotations for the different sub-tasks in the fact-checking process. In addition to the corpus, we release our corpus creation methodology that allows for efficiently creating large datasets with a high inter-annotator agreement in order to train machine learning models for automated fact-checking.

(2) In order to address the drawbacks of current automated fact-checking systems, we propose a pipeline approach that consists of the four sub-systems: *document retrieval, stance detection, evidence extraction*, and *claim validation*. Since today's machine learning models are not advanced enough to complete the task without human assistance, our pipeline approach is designed to help fact-checkers to speed up the fact-checking process rather than taking over the job entirely. Our pipeline is able to process raw text and thus, make use of the large amount of textual information available on the web, but at the same time, it is transparent, as the outputs of sub-components of the pipeline can be observed. Thus, the different parts of the fact-checking process are automated and potential errors can be identified and traced back to their origin.

(3) In order to assess the performance of the developed system, we evaluate the sub-components of the pipeline in highly competitive shared tasks. The stance detection component of the system is evaluated in the Fake News Challenge reaching the second rank out of 50 competing systems.[2] The document retrieval component together with the evidence extraction sub-system and the claim validation component are evaluated in the FEVER shared task.[3] The first two systems combined reach the first rank in the FEVER shared task Sentence Ranking sub-task outperforming 23 other competing systems. The claim validation component reaches the third rank in the FEVER Recognizing Textual Entailment sub-task.
(4) We evaluate our pipeline system, as well as other promising machine learning models for automated fact-checking, on our newly constructed Snopes fact-checking corpus. The results show that even though the systems are able to reach reasonable performance on other datasets, the systems under-perform on our newly created corpus. Our analysis reveals that the more realistic fact-checking problem setting defined by our corpus is more challenging than the problem setting posed by other fact-checking corpora. We therefore conclude that further research is required in order to increase the performance of the automated systems in real fact-checking scenarios.

[2] http://www.fakenewschallenge.org/

[3] http://fever.ai/2018/task.html

Zusammenfassung

In den letzten Jahren hat die Menge an Falschinformation im Internet stark zugenommen. Falsche Informationen verteilen sich sehr schnell in sozialen Netzwerken und erreichen durch diese größere Leserschaft als je zuvor. Das stellt unsere Gesellschaft vor neue Herausforderungen, da wir neu bewerten müssen, welchen Informationsquellen wir Glauben schenken dürfen und wie wir Webinhalte konsumieren und mit anderen teilen. Als eine Antwort auf die wachsende Menge an Falschinformation im Internet hat sich die Anzahl der Fact-Checking Organisationen erheblich erhöht. Auf diesen Plattformen validieren professionelle Fact-Checker publizierte Informationen und veröffentlichen die Ergebnisse ihrer Untersuchungen. Die manuelle Validierung der Informationen durch Fact-Checker ist jedoch sehr arbeitsintensiv und zeitaufwendig. Dadurch können nicht alle Inhalte überprüft werden und für validierte Inhalte erfolgt die Publikation der Analyse oft mit Verspätung. Zu diesem Zeitpunkt ist das Interesse an dem Thema in vielen Fällen schon gesunken, wodurch nur ein Bruchteil der ursprünglichen Leserschaft erreicht werden kann.

Automatisches Fact-Checking hat das Potenzial, diese Probleme zu lösen, weil es den Fact-Checkern ermöglichen könnte, Falschinformation zu erkennen und zu entfernen, bevor diese ein weites Publikum erreicht. Trotz der substanziellen Fortschritte auf diesem Gebiet, müssen noch mehrere Herausforderungen bewältigt werden, bevor automatisches Fact-Checking unter realen Bedingungen einsatzfähig wird: (i) Den Datensätzen, die für das Trainieren von Machine-Learning basierten Fact-Checking Systemen zur Verfügung stehen, fehlen qualitativ hochwertige Annotationen aus realen Fact-Checking Fällen für alle Teilaufgaben in dem Fact-Checking Prozess. (ii) Viele der heutigen Fact-Checking Systeme basieren auf Wissensdatenbanken, die nur eine relativ geringe Anzahl von Fakten abdecken, und weil für solche Systeme Sätze in natürlicher Sprache in formale Anfragen umgewandelt werden müssen, sind sie fehleranfällig. (iii) Moderne Machine-Learning basierte Systeme, die mittels Ende-zu-Ende Ansatz trainiert werden, können Text in natürlicher Sprache verarbeiten und dadurch potenziell die große Menge an Information im Internet nutzen. Diese Systeme sind aber intransparent und erreichen nicht die gewünschte Leistung. In der Tat ist Fact-Checking eine anspruchsvolle Aufgabe und moderne Machine-Learning basierte Systeme sind nicht ausgereift genug, um das Problem völlig ohne menschliche Unterstützung zu lösen. Um den identifizierten Herausforderungen zu begegnen, leisten wir in dieser Thesis die folgenden Beiträge:

(1) Wir erstellen ein neues Korpus, das auf der Snopes Fact-Checking Plattform basiert. Dieses Korpus beinhaltet reale Fact-Cheking Fälle, die mit qualitativ hochwertigen Annotationen für verschiedene Teilaufgaben innerhalb des Fact-Checking Prozesses angereichert wurden. Des Weiteren veröffentlichen wir unseren Ansatz für den effizienten Aufbau von großen Datensätzen, die dafür geeignet sind, Modelle für das automatisierte Fact-Checking zu trainieren.

(2) Um den Nachteilen heutiger Fact-Checking Systeme zu begegnen, stellen wir in dieser Thesis einen neuen Pipeline-Ansatz vor, der aus folgenden vier Komponenten besteht: *Document Rertrieval, Stance Detection, Evidence Extraction, und Claim Validation.* Weil heutige Machine-Learning basierte Systeme noch nicht ausgereift genug sind um das Fact-Checking Problem eigenständig zu lösen, ist unser

Pipeline-Ansatz speziell dafür entwickelt worden, Fact-Checker bei ihrer Arbeit zu unterstützen und nicht etwa den gesamten Fact-Checking Prozess eigenständig durchzuführen. Unser Pipeline-Ansatz ist dazu in der Lage, natürliche Sprache zu verarbeiten und dadurch die große Menge an Information in Textform aus dem Internet zu nutzen. Gleichzeitig ist unser System transparent, da die Ausgaben der dazwischenliegenden Systeme in der Pipeline eingesehen werden können. Dadurch ist es möglich, einzelne Aufgaben in dem Fact-Checking Prozess zu automatisieren und gleichzeitig potenzielle Fehler zu erkennen und auf ihren Ursprung zurückzuführen.
(3) Um die Leistungsfähigkeit der Subkomponenten der Pipeline zu testen, evaluieren wir sie in mehreren hart umkämpften internationalen Wettbewerben. Die Stance Detection Komponente der Pipeline erreicht den zweiten Platz unter 50 konkurrierenden Systemen in der Fake News Challenge.[4] Die Dokument-Retrieval Komponente, die Evidence-Extraction Komponente, und die Claim-Validation Komponente werden in dem FEVER Shared Task evaluiert.[5] Die ersten zwei Komponenten kombiniert erreichen den ersten Platz bei der *FEVER Shared Task Sentence Ranking* Aufgabenstellung. Die Claim-Validation Komponente erreicht den dritten Platz in der *FEVER Recognizing Textual Entailment* Aufgabenstellung.
(4) Wir evaluieren unser Pipeline System, sowie andere leistungsfähige Modelle, die für das automatisierte Fact-Checking entwickelt worden sind, mit unserem neu erstellten Snopes Fact-Checking Korpus. Die Ergebnisse zeigen, dass, obwohl die Systeme gute Ergebnisse an anderen Korpora erzielen, die Leistung der Systeme auf unserem Korpus relativ gering ausfällt. Unsere Analyse ergibt, dass die realistische Aufgabenstellung, definiert durch unser Korpus, deutlich schwieriger ist als diejenigen Fact-Checking Aufgabenstellungen, die durch die anderen Korpora definiert werden. Wir folgern daraus, dass weitere Forschung notwendig ist, um die Leistungsfähigkeit der automatisierten Systeme in realistischen Fact-Checking Szenarien zu erhöhen.

[4] `http://www.fakenewschallenge.org/`

[5] `http://fever.ai/2018/task.html`

Acknowledgments

The doctoral program and ultimately this work has received extraordinary support from many people for which I am very grateful and for which I would like to express my gratitude. I would like to thank Prof. Iryna Gurevych for giving me the opportunity to complete my doctoral thesis under her supervision and fulfill my long-term goal of becoming a researcher of machine learning and language. Her guidance in research projects, as well as her assistance to get connected with other researchers was extraordinarily helpful to make the doctoral program a success.

There were two competitions at the core of my doctoral research the Fake News Challenge and the FEVER shared task, and it was here, where my work had the most impact and where I have experienced the most pleasure of doing research. The projects would never have been so successful without the people who joined the projects, and I am very grateful to have had the opportunity to collaborate with such talented researchers. Therefore, I would like to thank Benjamin Schiller, Avinesh PVS, Felix Caspelherr, and Debanjan Chaudhuri for participating with me in the Fake News Challenge and Hao Zhang, Zile Li, Daniil Sorokin, Claudia Schulz for taking part with me in the FEVER shared task.

I also would like to thank my UKP colleagues for fruitful discussions of my presentations, for helpful feedback on my papers drafts, but also for their friendship. In particular, I would like to thank my collaborators in various projects for their support and guidance, namely Christian Mayer, Christian Stab, Christopher Tauchmann, and Thomas Arnold.

The AIPHES research group was a great source of inspiration. I have very much enjoyed our controversial reading groups, the research retreats, and the Fridays philosophical coffees. Therefore, I would like to thank all members of AIPHES and in particular Teresa Botschen, Markus Zopf, Tobias Falke, Ana Marasovic, Todor Mihaylov, Beniamin Heinzerling, Fabrizio Ventola, Aicha Diallo, Shweta Mahajan, Leonardo Filipe Rodrigues Ribeiro, Ajie Utama, Federico López, Wei Zhao, Debjit Paul, and Benjamin Hättasch.

Finally, I also would like to thank my parents for their support during demanding periods of the doctoral project, their pride has served as a valuable motivation to pursue my research goals.

Contents

Chapter 1

Introduction and Background

The ever-increasing role of the Internet as a primary communication channel is arguably the most important development in the media over the past decades. In drastic contrast to the traditional editorial-board-based publishing, the modern Internet is decentralized, and anyone with access to a network can produce and broadcast new content to a wide audience at almost zero expense. This has resulted in unprecedented growth in information coverage and distribution speed. Nevertheless, there are also considerable downsides to the increased technical possibilities of freely generating and distributing content. False information, such as false claims or entire false-news articles (Paskin, 2018), can be shared through this channel and reach a wider audience than ever before (Howell et al., 2013). Individuals or organizations hiding behind fake accounts on social media can generate and distribute false content without being held accountable for their actions (SafeGuardCyber, 2019). This phenomenon is typically called *disinformation*, as opposed to the *information* of an audience about the true state of events. In addition to malicious content generated by humans, an increased amount of harmful content is generated by machines. Recent advances in AI and machine learning have made it possible to generate fake images,[1] fake videos,[2] or fake speech.[3] Recently developed deep neural networks pretrained on massive corpora can generate coherent pieces of text on a desired issue and thus mislead information consumers[4] (Radford et al., 2019; Zellers et al., 2019).

In response to the rising amount of false information on the Internet, the number of fact-checking platforms has increased. Whereas in 2014 only 44 fact-checking platforms were active worldwide, in 2019 the number of active platforms has risen to 188 organizations.[5] On these platforms, professional fact-checkers validate the distributed information and make their *fact-checks* publicly available. The fact-

[1] https://www.theverge.com/2018/12/17/18144356/ai-image-generation-fake-faces-people-nvidia-generative-adversarial-networks-gans

[2] https://www.theguardian.com/news/shortcuts/2019/aug/13/danger-deepfakes-viral-video-bill-hader-tom-cruise

[3] https://www.bbc.com/news/av/technology-40598465/fake-obama-created-using-ai-tool-to-make-phoney-speeches

[4] https://www.technologyreview.com/s/612960/an-ai-tool-auto-generates-fake-news-bogus-tweets-and-plenty-of-gibberish/

[5] https://www.poynter.org/fact-checking/2019/number-of-fact-checking-outlets-surges-to-188-in-more-than-60-countries/

checks typically contain a *verdict* for the validated claim, such as *true* or *false*, and evidence to back up the verdict.

However, despite its popularity, traditional, manual fact-checking practiced by most of the fact-checking platforms today has substantial drawbacks. The process is laborious, and because of the large amount of the false information on the web, it is not possible to identify and validate every false claim. Moreover, even if false information is identified and validated, most of the damage is already done once a misleading message goes viral. The majority of the news consumers are unlikely to review the facts on a story once the focus of the media has shifted to a different topic. Nevertheless, even if the public is informed about the corrected version of the story, psychological studies show that once beliefs are formed, they cannot be easily reversed (Levy, 2017a,b).

Many of the issues with manual fact-checking can be addressed by automated approaches, as they would allow fact-checkers to validate a large amount of information as it appears on the web. Thus, the operators of the platforms on which the deceptive content was published can remove the falsehoods before they can spread through social networks. Owing to recent advances in machine learning and AI, a number of tasks previously performed by humans can be successfully automated, e.g., speech recognition (Graves et al., 2013), object detection (Krizhevsky et al., 2012), or product recommendation (Linden et al., 2003). Progress in Natural Language Processing (NLP) as a sub-field of AI has led to the development of advanced methods for solving problems related to natural language that could also be leveraged to automate the fact-checking process. Thus, *automated fact-checking* has received increased research interest in the last couple of years. Today, scientists as well as corporations are exploring different approaches to address this task, such as designing fact-checking systems based on knowledge bases (Ciampaglia et al., 2015) or using machine learning methods to harness textual information from the web to validate published content (Popat et al., 2017).

In this chapter, we provide background information for our contributions to automated fact-checking presented in this thesis. Because the information validation process depends on the characteristics of the false information and the proliferation of information on the Internet, the first part of this chapter is devoted to the analysis of these two topics. We investigate different kinds of false information and the proliferation of false content on the Internet. We analyze the mechanics of the proliferation and why disinformation on the web has such a large influence.

The second part of this chapter provides background information about the actual fact-checking process. We first examine traditional fact-checking that is internally practiced by news magazines. Thereafter, we discuss the recent increase in the number of external fact-checkers in response to the increased amount of false information on the web. Next, we explore the emerging field of automated fact-checking, whereby different automated fact-checking frameworks are presented. The section is concluded with the discussion of the fields related to fact-checking: fake-news detection, computational argumentation, interactive evidence detection, and the identification of automatically generated false textual content.

1.1 False information and its proliferation

The distribution of false information to manipulate others for achieving a certain goal is not a new phenomenon but has evolved over time. *Propaganda* or *disinformation* was widely used by people in power from antiquity until the modern era to manipulate public opinion (Jowett and Donnell, 2006). With the rise of the Internet, not only the way we consume information, but also the tools *bad actors* (SafeGuardCyber, 2019) use to get their message across have changed. Today, information can be shared instantly, and it can reach a wide audience in a short period of time through the distribution of the message on social networks. Below, we give an overview of different kinds of false information and discuss the proliferation mechanisms of false information on the web. Here, the term false information will be broadly used to refer to different kinds of manipulation of the original message, e.g., intentional/unintentional manipulation, presenting information out of context, and inappropriately highlighting/hiding different aspects of the original message.

1.1.1 Different kinds of false information

The manipulation of groups of people by distributing false information is an ancient phenomenon dating back at least to antiquity. However, the methods for this manipulation have changed over the years, as the newest technology is always adopted to efficiently distribute the misleading content. During the Protestant Reformation, the printing press was widely deployed by different parties to spread false information to discredit the opponent. In the modern era, radio, television, and now the Internet are used to manipulate others.

The terminology to address different kinds of falsehoods has developed ever since. The term *propaganda* was originally used by the Vatican, in order to refer to the propagation of the faith of the Roman Catholic Church, but the term lost its neutral connotation owing to its excessive use against alternative beliefs to establish the Catholic faith in the "new world" (Jowett and Donnell, 2006). Since then, different forms of the term propaganda have evolved. Jowett and Donnell (2006) differentiate between three different kinds of propaganda:

- *White propaganda* is distributed by a source that correctly reveals itself and, the distributed message is mostly accurate. It emphasizes the good intent of the information provider, and its purpose is to build a relation of trust with the information consumer. The domestic mainstream media, for example, is in general more in favor of the actions of the government than foreign reporters. A more left-leaning news network favors politicians on the left spectrum and is more likely to be silent about their misconduct than the right-learning media and vise versa. The reports of the source are mostly accurate, but through the selection of the content and by highlighting different aspects of the message, a biased view is presented.

- *Black propaganda* refers to different kinds of deception, the distribution of lies, and fabricated stories. The source of the message is thereby often concealed or credited to a false authority. One form of black propaganda is the widely used term *disinformation*. It was derived from the Russian word *dezinformatsia*, which was the name of the KGB division devoted to black propaganda (Jowett and Donnell, 2006; Shultz and Godson, 1984). Shultz and Godson (1984) proposes the following definition of the term: Disinformation means "*false, incomplete, or misleading information that is passed, fed, or confirmed to a targeted individual, group, or country*". Disinformation differs from *misinformation* in that the former is a purposeful and intentional act of spreading false information (Golbeck, 2008) and the latter is an unintended error in the propagated message.

- *Gray propaganda* refers to information that is somewhere between black and white propaganda. The source of the message may or may not be correctly revealed, and the accuracy of the content is uncertain.

More recently, *fake news* has emerged as a new term to refer to false information and particularly to fabricated news articles. The term was mainly popularized during the 2016 U.S. presidential elections, in response to the growing amount of highly partisan news on the Internet. Fake news can be defined as an intentional (Rubin et al., 2015) and knowing (Klein and Wueller, 2017) deception, with the purpose of either political or monetary gain (Allcott and Gentzkow, 2017). Facebook defines fake news as "*inaccurate or manipulated information/content that is spread intentionally*" (Weedon et al., 2017). Because the definitions stress the deceptive nature of the message and the information being intentionally distributed, fake news can also be considered as a kind of black propaganda.

The term fake news is not new, but its meaning has altered over the years. It was originally used to refer to satirical and parody news shows (Paskin, 2018). More recently, the term has been excessively used by public figures to discredit news organizations that negatively report about them or simply to attack the political opponent (Klein and Wueller, 2017). Because fake news is now often used as an insult rather than according to the definition given by Allcott and Gentzkow (2017); Rubin et al. (2015); Klein and Wueller (2017); Weedon et al. (2017), the meaning of the term has become increasingly deluded. For this reason, avoiding the term altogether in a serious discussion about disinformation has been suggested (Thorne and Vlachos, 2018; Vosoughi et al., 2018). Thus, we will refer to this kind of false information as *false news* in the rest of the thesis.

For a better understanding of the relations between the various terms referring to different kinds of false information, in Figure 1.1, we display a taxonomy of these terms.

1.1.2 Proliferation of false information on the Internet

In this section, we examine the proliferation of false information on the Internet and the topics associated with it. The proliferation of false information has many facets that need to be examined for a comprehensive overview of the topic. Below we discuss the distribution of false information on social media by bad actors, the proliferation mechanisms, the influence of the false information on the information consumers, and the containment of disinformation. The goal is to analyze the dynamics of disinformation on the web, which in turn can help to develop effective countermeasures.

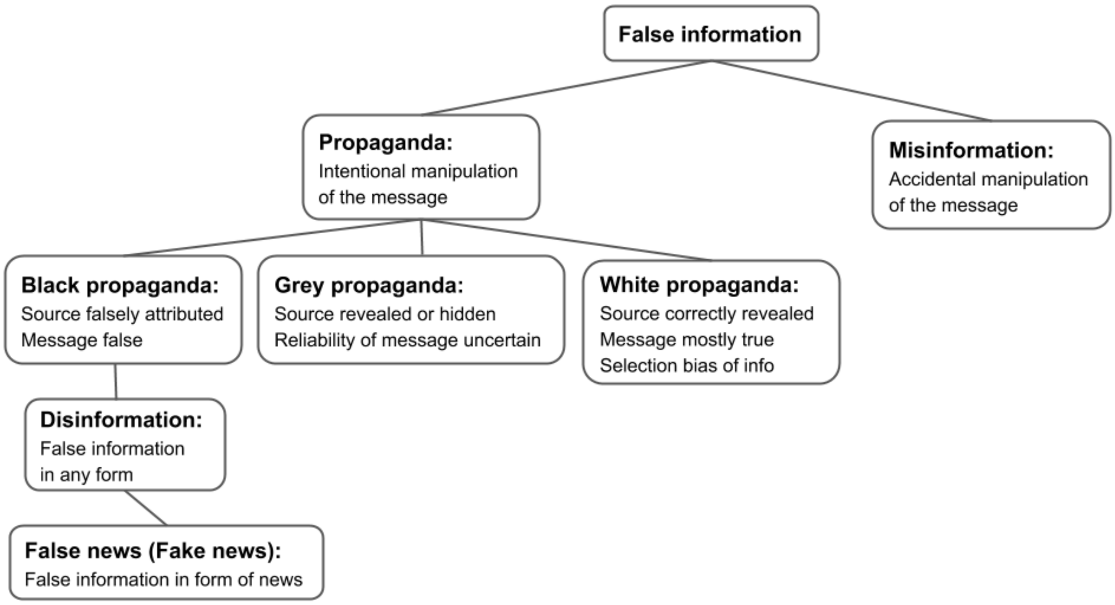

False information
Propaganda: Intentional manipulation of the message
Misinformation: Accidental manipulation of the message
Black propaganda: Source falsely attributed Message false
Grey propaganda: Source revealed or hidden Reliability of message uncertain
White propaganda: Source correctly revealed Message mostly true Selection bias of info
Disinformation: False information in any form
False news (Fake news): False information in form of news

Figure 1.1: Taxonomy of the different kinds of false information

Social media as a news provider. Social media has recently emerged as a new vehicle to drive the distribution of information on the web. Platforms such as Facebook, Twitter, the Russian Odnoklasniki, or the Chinese Sina-Weibo have hundreds of millions of active users (Webb et al., 2016), and the distribution of news articles and political messages is an important means of interaction on these platforms. Users are increasingly abandoning mainstream media and moving to social media platforms, where they receive the news reports on their news feeds (Bakshy et al., 2015). According to Gottfried and Shearer (2016), in the U.S., for instance, 62% of adults get their news on social media.

Distribution of false information by bad actors. Media depends on sensationalism, novelty over newsworthiness, and clickbait to get the attention of the information consumers, and this makes it vulnerable to manipulation (Marwick and Lewis, 2017). *Bad actors* (SafeGuardCyber, 2019) take advantage of this media ecosystem to manipulate news frames, set agendas, and propagate ideas. Among them are far-right groups, trolls, white nationalists, men's rights activists, gamergaters (Braithwaite, 2016), the "altright" (Marwick and Lewis, 2017), and individuals representing the interests of large corporations[6] or of different governments (Ramsay and Robertshaw, 2019). The incentives for spreading the disinformation are thereby very diverse, ranging from monetary gains, such as profits from ads (Chen et al., 2015), to the attempt of manipulating the public opinion in elections in one's favor (Persily, 2017; SafeGuardCyber, 2019).

Proliferation mechanisms and their influence on information consumers. A number of studies have shown that the influence of false information on consumers

[6] https://www.theguardian.com/environment/2015/mar/25/fossil-fuel-firms-are-still-bankrolling-climate-denial-lobby-groups

is substantial and should not be underestimated. Silverman (2016) has observed that the most popular false news stories were more widely shared on Facebook than the most popular mainstream news stories. Vosoughi et al. (2018) have found that false news stories reached more people than reliable news articles: "*The top 1% of false news cascades diffused to between 1000 and 100,000 people, whereas the truth rarely diffused to more than 1000 people. Falsehood also diffused faster than the truth*". False information therefore proliferates significantly farther, faster, deeper, and more broadly in social networks than the truth over all categories of information.

Numerous studies have analyzed why false information is distributed more widely than the truth and what are the properties of false news that make it more worth sharing. Vosoughi et al. (2018) have discovered that false news is more novel than true news, and because novel information is more worth sharing, they imply that people are more likely to share false information. Moreover, false stories inspired fear, disgust, and surprise, whereas true stories inspired anticipation, sadness, joy, and trust. Thus, as suggested by Vosoughi et al. (2018), false information appears to be more exciting and therefore receives much attention. Nevertheless, users share false information not only because it is novel or exciting, but also because they often believe in the content and are unaware of its deceptive message (Silverman and Singer-Vine, 2016).

An additional negative effect of the consumption of news on social media is that the diversity of the information decreases. According to Bakshy et al. (2015), users are less likely to encounter information in their news feeds from the opposite political view than the information aligned with their political perspective. Friends, for instance, share substantially fewer news articles presenting an opposing ideology. Another source of bias on social media is the algorithm that filters and ranks the content in the news feed. Bakshy et al. (2015) report that in a study, users encountered about 15% less content from their opposite political spectrum in their news feed owing to the ranking algorithm and clicked up to 70% less through this content. This consumer biased feedback reinforces the bias of the ranking algorithm, and the algorithm is therefore less likely to present news with an alternative view in the future.

Psychological perspective on disinformation. Given that false information is widely distributed through social networks and that many of the users believe the falsehoods, the question arises of why people are susceptible to false information in the first place and are not more critical about the information that they consume. Psychological theories suggest that this is an interplay of different factors. (i) One factor is our *implicit bias* or *implicit stereotype* (Greenwald and Banaji, 1995).[7] We innocuously form a set of beliefs according to which we perceive and treat other people. We consider people like ourselves as more honest and trustworthy than people belonging to a different ethical, religious, or age group. (ii) Another factor is our *confirmation bias*, which refers to the phenomenon that we are more willing to accept new information that is in agreement with our beliefs than to accept information that is at odds with our worldview (Waldman, 2017; Sunstein, 2014).

[7] `https://www.psychologytoday.com/us/blog/contemporary-psychoanalysis-in-action/201612/fake-news-why-we-fall-it`

Thus, conservative people are more inclined to accept false claims about left-leaning politicians, whereas progressive people are more susceptible to false information about conservative politicians.

Owing to our biases, we are less likely to accept information from an unfamiliar source and information that is not in agreement with our worldview. The two biases reinforce each other and we end up in a cycle, in which we are seldom are confronted with an alternative perspective. The selective consumption of information leads to *echo chambers* or *filter bubbles*, which refers to the phenomenon that a group of people is only sharing information within the group. The group members are only exposed to information from like-minded people confirming their biases. Attitude-changing content is therefore avoided, leading to polarization effects (Waldman, 2017; Bakshy et al., 2015; Pariser, 2011; Flaxman et al., 2013).

For a better overview of how different factors affect our consumption of information, in Figure 1.2, we illustrate the information consumption process and the risks associated with it, such as our biases and reinforcing feedback loops.

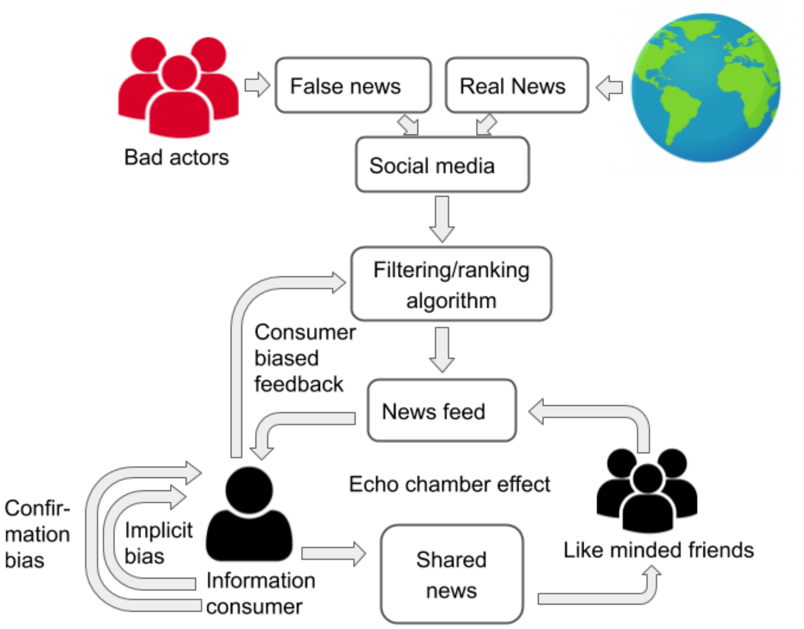


Figure 1.2: Information consumption process and its associated risks, such as our biases and reinforcing feedback loops

Containment of disinformation. If disinformation on the web is so harmful, why have we not yet found an effective solution to the problem? The proliferation of false information on social media cannot be easily controlled for several reasons. The insensitive removal of information by authorities or social media platforms can be conceived as censorship and can face great opposition. Moreover, because of the sheer amount of false information on the web, it is difficult to detect and eliminate every single false claim or article. Self-correcting mechanisms—for example, as observed on Wikipedia, where some users repair the damage done by other users—do not seem to apply to the fast proliferation of information on social media. The information often goes viral before it is detected and corrected by other users in a social network (Webb et al., 2016). To address these problems, information validation approaches are required that can quickly and reliably identify false information on the Internet. The deceptive content can then be removed before it spreads through social networks. Modern fact-checking approaches, discussed in the following section, hold the promise to fulfill these objectives and are therefore in the focus of the public debate about disinformation.

1.2 Fact-checking

Traditionally, fact-checking has been manually done by trained employees at news magazines to ensure the quality of an article before publication. With the increase in the amount of false information on the web, external fact-checking platforms emerged, where fact-checkers manually validate information published on the web and share their findings with the public. Modern approaches to fact-checking include different degrees of automation of the fact-checking process, ranging from only using a search engine to extract information from the web to completely automated fact-checking pipelines. In this section, we first present the traditional manual fact-checking approaches before discussing different variants of modern automated fact-checking methods. The section is concluded with a discussion of NLP tasks related to automated fact-checking.

1.2.1 Manual fact-checking

Internal fact-checking. Fact-checking originally emerged as an internal quality assurance process within news magazines. In this process, an article written by a reporter is validated by a professional fact-checker. The internal fact-checking process was proposed by Briton Hadden, who co-founded the Time magazine (Thomas and Weiss, 2010); however, the validation process was not originally called *fact-checking*.[8] The information validation process practiced by the Time magazine, as well as by the News Week magazine,[9] is referred to as *reporter-researchers*.[10] It can be considered as a collaboration between the reporter and the fact-checker (researcher), in which the fact-checker accompanies the reporter and helps with putting the article together. The fact-checker validates the alleged facts gathered by the reporter while the article is written. However, the fact-checker is sometimes also involved in interviewing the sources (experts, witnesses) and writing the article (Thomas and Weiss, 2010).

At the New Yorker,[11] which is an American magazine well known for its fact-checking process, a different information validation approach was established. It significantly differs from the *reporter-researchers* approach and relies on *checks and balances* (Thomas and Weiss, 2010). Here, the fact-checker works independently of the reporter and is not helping to write the article. Instead, the fact-checker gets involved after the article is written. The fact-checker obtains the background material

[8] `http://time.com/4858683/fact-checking-history/`

[9] `https://www.newsweek.com/`

[10] `http://time.com/4858683/fact-checking-history/`

[11] `https://www.newyorker.com/`

from the reporter, such as the newspaper clips, magazine stories, websites the reporter consulted, telephone numbers of the sources, etc. (Thomas and Weiss, 2010). Then, "*the fact-checker takes the article apart and puts it back together again*",[12] which basically means validating the alleged facts in the article, double-checking the statements of the sources, and validating the coherence of the arguments.

External fact-checking. With the increased amount of false information on the web, there is a growing demand for *external fact-checking*. External fact-checkers or fact-checking platforms are (mostly) independent of news providers and are only concerned with the validation of information coming from a secondary source, such as news articles, claims made by public figures, or rumors emerging on social media. According to Duke Reporters' Lab,[13] 188 of such organizations were active in 2019. More than 90% of them were established since 2010, and about 50% in 2016 alone. The American Press Institute reports that from 2004 to 2008 the number of fact-checked stories increased by more than 50% and from 2008 to 2012 by more than 300%.[14] To give a brief overview of how external fact-checkers operate, below we present four popular external fact-checking platforms. Each of these platform focuses on one particular kind of disinformation and follows its own fact-checking process.

`www.politifact.com`: Politifact is a nonprofit organization located in Florida and is arguably the most prominent external fact-checker. It received the Pulitzer Prize for national reporting in 2009 for its fact-checking efforts during the 2008 U.S. presidential campaign. Politifact focuses mainly on U.S. politics and validates claims made by elected officials, candidates, their staff, lobbyists, bloggers, and other public figures. Validated claims receive a *Truth-O-Meter* rating (or verdict) that ranges across six different categories from *true* to *pants on fire* (meaning that the claim is entirely false). Each claim is accompanied by a detailed analysis, where the fact-checker presents evidence that supports the given verdict and refutes evidence that supports a different verdict.

`www.snopes.com`: Snopes,[15] also known as *Urban Legends Reference Pages*, is an organization owned by the Snopes Media Group.[16] Snopes Media Group is almost entirely funded through digital advertising sales, which are independent of their fact-checking efforts: that is, advertisers have no influence over the published content. Donations over $10,000 are disclosed to the public.[17] Snopes focuses on debunking myths, rumors, and political claims on the web. Their "fact-checks" are frequently cited by the news media, such as CNN, MSNBC, or the New York Times. Like Politifact, Snopes provides a verdict for each claim, but the number of classes

[12] `https://archives.cjr.org/critical_eye/fact-checking_at_the_new_yorker.php`

[13] `https://www.poynter.org/fact-checking/2019/number-of-fact-checking-outlets-surges-to-188-in-more-than-60-countries/`

[14] `https://www.americanpressinstitute.org/fact-checking-project/new-research-on-political-fact-checking-growing-and-influential-but-partisanship-is-a-factor/`

[15] Because the Snopes website serves as a source for the corpus construction part of this thesis, the structure of the website is discussed in more detail in Section 3.2.1.

[16] `https://www.snopes.com/about-snopes/`

[17] `https://www.snopes.com/disclosures/`

for the verdict is much more diverse. A fact-checker can label the claim as being not only *true* or *false* but also as *undetermined*, *mixed*, *mostly false*, *legend* etc., if the validated issue is more nuanced. The fact-checkers also provide a detailed analysis for each validated claim, in which they present explicit evidence extracted from various sources that supports the different perspectives on the topic.

`www.fullfact.org`: FullFact is an independent fact-checking charity based in London and mostly focuses on validating and correcting claims made by U.K. politicians and news providers. The fact-checking process relies not only on their own research but also on insights of external academics.[18] Like Politifact and Snopes, FullFact provides a detailed analysis for each validated claim. However, in contrast to other fact-checkers, no verdicts for the claims are given, that is, the claim is not labeled as *true, false, etc.* Instead, FullFact presents a summary that contains the claim and a short conclusion of the analysis.

`www.factcheck.org`: FactCheck is a nonprofit project of the Annenberg Public Policy Center of the University of Pennsylvania. It is funded by the Annenberg Foundation and donations from individuals. FactCheck avoids funding from corporations, unions, partisan organizations, or advocacy groups.[19]The project's mission is to reduce the level of deception and confusion in U.S. politics.[19] FactCheck has won four Webby Awards[20] in the Politics category for their fact-checking efforts. On their website, FactCheck investigates different political events that are subject to some kind of controversy. For the investigated events, they provide an analysis and often explicit evidence that either supports their conclusions or presents a different perspective on the topic. As practiced by FullFact, besides the analysis, no verdicts for the investigated events are given.

For a better overview of the four external fact-checkers, the main characteristics of the fact-checking platforms are given in Table 1.1. In order to be credible, the fact-checkers try to disclose their sources of funding and abstain from donations from political parties or lobbyists. The four fact-checkers have a different fact-checking procedure and structure their *fact-checks* differently. Whereas some provide an explicit verdict for the claims, others only present evidence.

The fact-checking process. The validation approaches of different internal and external fact-checkers significantly vary and the American Press Institute (as one important authority on the issue) has its own guidelines for fact-checking.[21] However, there are a number of steps which many of the fact-checkers follow when evaluating a piece of text and that we consider to be important to include in the fact-checking process. A summary of these steps is given below, and steps are illustrated in Fig-

[18] `https://www.poynter.org/fact-checking/2016/lessons-from-fact-checking-the-brexit-debate/`

[19] `https://www.factcheck.org/spindetectors/about/`

[20] The Webby Award is an Internet-oriented award that honors outstanding Internet content in different web categories: `https://www.webbyawards.com/`

[21] `http://cdn.gatehousemedia.com/custom-systems/ghns/files/upload/files/home/ghns-staff/Training/SampleFact-CheckingGuidelinesforNewsrooms%20(4).docx`

	main focus	explicit verdict	explicit evidence	country	funding
`Politifact`	U.S. politics	yes	no	U.S.	nonprofit
`Snopes`	Web rumors	yes	yes	U.S.	Snopes Media
`FullFact`	U.K. politics	no	yes	U.K.	charity
`FactCheck`	U.S. politics	no	yes	U.S.	nonprofit

Table 1.1: Overview of external fact-checking platforms (explicit verdict: the claim is labeled with a verdict (true, false, ...); (explicit evidence: the evidence is highlighted in text))

ure 1.3. Nevertheless, note that such a list can never be considered as an objective summary of the fact-checking process because the kind of the validated information as well as the goals of the different fact-checking organizations are too diverse to be summarized into one single best practice that fits all needs.

1. *Getting familiar with the topic*: The fact-checker first needs to get acquainted with the topic of the given text by reading related information that is published on the topic.

2. *Identification of the claims/alleged facts in the text*: The central claims/alleged facts in the text that are not commonly agreed to be true need to be identified. These claims form the basis of the text and need to be validated.

3. *Evidence aggregation*: The fact-checker consults different sources to collect evidence that supports or refutes the identified claims. This includes finding facts on the subject on the Internet and in databases or interviewing experts or witnesses.

4. *Checking the credibility of sources*: When aggregating evidence, the evaluation of the credibility and independence of the sources of the evidence is important. A source might have an interest to manipulate the public's perception on an issue for their own advantage and can therefore be biased.

5. *Claim validation*: In this step, the identified claims/alleged facts are validated, i.e., the claims are labeled with a verdict (*true*, *false*, ...) on the basis of the collected evidence.

6. *Validation of the reasoning chain*: Once the individual claims are validated, the reasoning chain in the text as a whole needs to be assessed: that is, whether the conclusions drawn by the author logically follow from the validated claims.

7. *Checking for fallacies*: The argumentation of the author must be evaluated for fallacies or red flags,[22] such as deceptive dramatization, guilt by association, deception by omission, etc.

[22] `http://cdn.gatehousemedia.com/custom-systems/ghns/files/upload/files/home/ghns-staff/Training/SampleFact-CheckingGuidelinesforNewsrooms%20(4).docx`

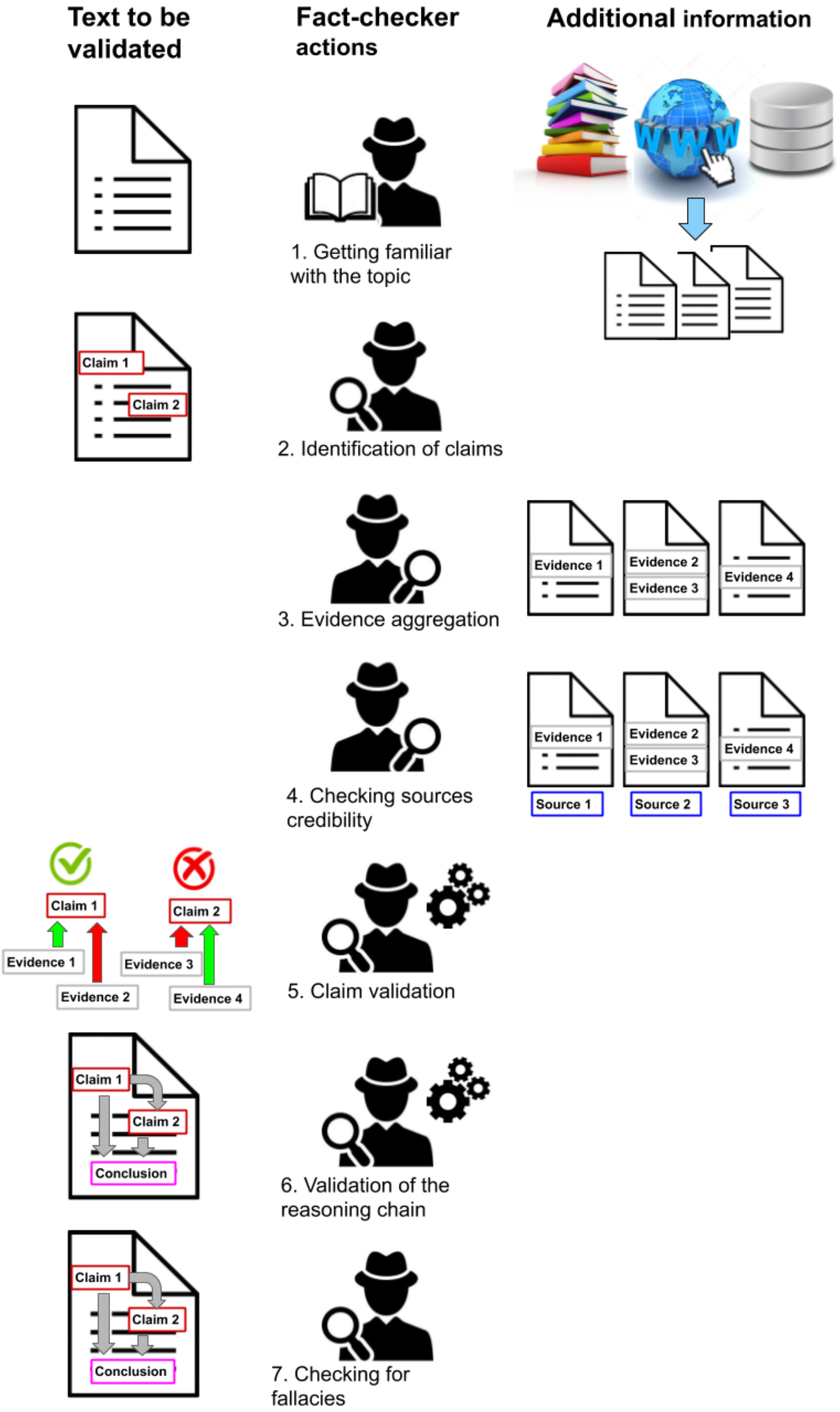


Figure 1.3: The fact-checking process

1.2.2 Automated fact-checking

Even though the number of external fact-checking platforms has substantially increased in the past couple of years, because the manual fact-checking process is laborious, it is unlikely that these platforms will be able to keep up with the increasing amount of false information on the web. Because many of the upcoming issues of manual fact-checking can be addressed by automated approaches, the interest in this research field has significantly increased. In this sub-section, we give an overview of existing automated fact-checking approaches. We first introduce the terminology that is typically used to refer to important concepts in the field of automated fact-checking and that we have adopted in this thesis. Then, we present traditional automated fact-checking approaches based on knowledge bases. Next, we discuss the automation of the sub-tasks of the fact-checking process by leveraging machine learning techniques developed for different NLP problems. Thereafter, an overview of existing machine learning-based pipeline approaches to automated fact-checking is given.

Terminology

Fact-checking. As described above, the term *fact-checking* is typically used by journalists to refer to the internal or external process for the validation of a piece of text or a claim. Shapiro et al. (2013) define fact-checking as a "*discipline of verification*", and in fact, the term *verification* is often interchangeably used with fact-checking. More recently, however, the meanings of the two terms have become more distinct (Kovach and Rosenstiel, 2014; Thorne and Vlachos, 2018). Whereas fact-checking still refers to the entire fact-checking process, the term verification in its more recent use refers to a part of it: "*verifying the source, date, and location of materials (evidence)*" (Thorne and Vlachos, 2018, page 2).

Evidence. In this thesis, we only consider the automated validation of claims based on textual data. Thus, unlike in manual fact-checking, where a piece of evidence can be a graph, a diagram, or an image, evidence will only refer to pieces of a text on different levels of granularity, such as text snippets, sentences, or phrases. A given piece of text needs to provide some kind of valuable information for the validation of a given claim to be considered as evidence. It can be a conclusion of a scientific study, an expert opinion, or a witness account. Moreover, because we grant the authors of the web document some authority on the issue, we consider statements in web documents as evidence if they paraphrase the claim or contradict it. Evidence typically expresses a stance with respect to the claim: that is, it supports or refutes the claim. However, evidences can also be *fragmentary*: that is, they might only in combination support or refute a claim. E.g. **Claim:** *Tesla is planning to build a new factory near the capital of Germany.* **Evidences:** *Tesla revealed plans to build a new factory near Berlin. Berlin is the capital of Germany.* Such evidences are closely related to *premises* in the field of logic and argumentation, where several premises in combination (linked premises (Schneider, 2014)) can provide sufficient information to solve a *syllogism.*

Claim. A claim will be defined as a factual statement that is under investigation: that is, it is not yet clear whether the claim is a fact or not (whether it is true or false). It must refer to factual matters in the world that can be objectively assessed, e.g., *a high sugar diet leads to diabetes* or *Donald Tusk is the president of the European Council.* This definition deviates from the interpretation of a claim in the field of computational argumentation and argumentation mining, where claims often refer to matters of opinion that are subject to debate and for which no objective conclusions can be obtained. Examples of such claims are *"abortion is morally wrong and should be banned"*, or *"homeschooling is superior to education at public schools"*.

Verdict. A verdict will be defined as a rating indicating whether a given claim is true. Alternative definitions are *veracity*, *truthfulness*, or *rating*. Typical verdicts for a claim are *true*, *false*, *mostly false*, *mostly false*, or *not enough info* if the provided information is not enough to classify the claim as true or false with high confidence.

Automated fact-checking based on knowledge bases

Many of the first approaches to automated fact-checking rely on information from knowledge bases (Shi and Weninger, 2016a,b; Ciampaglia et al., 2015; Conroy et al., 2015; Gerber et al., 2015). Such approaches use general knowledge bases that contain large collections of facts on a variety of topics, see for instance: Wikidata (Vrandečić and Krötzsch, 2014), DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), or NELL (Carlson et al., 2010). For the validation of a claim on the basis of knowledge bases, a number of different methods have been devised. For a better understanding of how such approaches work, we briefly discuss two approaches below.

Ciampaglia et al. (2015) and Conroy et al. (2015) determine the veracity of a claim by measuring the proximity of two entities of the claim in a knowledge graph. For instance, in order to validate the claim *Barack Obama is a Muslim*, the shortest path between the entities *Barack Obama* and *Islam* is determined. The path length and the entities along the path are considered as features for the validation of the claim.

Shi and Weninger (2016a,b) define claim validation as a *link prediction problem*. Given a relational triple in the form (subject, predicate, object), for example, *(Chicago, capitalOf, Illinois)*, the approach derives a generalized statement *(U.S. city, capitalOf, U.S. state)*. These general statements, in turn, are used to validated candidate triples: e.g., the statement *(Barack Obama, capitalOf, Illinois)* is false because according to the general statement *(U.S. city, capitalOf, U.S. state) Barack Obama*, is not a *U.S. city*.

Both approaches have their strengths and weaknesses. The method developed by Ciampaglia et al. (2015) is relatively easy to implement, and it can be used to validate an unlimited number of claims, as any claim containing two entities from the knowledge base can be validated. The method proposed by Shi and Weninger (2016a,b) is more complicated, because before the claim can be validated, appropriate general statements need to be derived. Nevertheless, in contrast to the method by (Ciampaglia et al., 2015), it does take the relation between the two entities in the claim into account. In fact, the method proposed by Shi and Weninger (2016a,b) is particularly designed to evaluate whether a particular relation between two entities in the knowledge base holds.

The fundamental drawback of using knowledge bases for fact-checking is that they have a relatively low coverage, as they only contain a small fraction of the knowledge available on the web. Moreover, because the process of updating a knowledge base is laborious, important new facts are often missing. Furthermore, to compare the information contained in the claim with the information in the knowledge base, a claim needs to be transformed to a relational triple that introduces an additional source of error.

The described challenges can be addressed by text-based fact-checking approaches that can directly process raw text and thus, potentially harness the vast amount of information available on the web. Such approaches are presented in the following sub-sections.

Text-based approaches to automated fact-checking

To speed up the fact-checking process, for some steps of the process, machine learning approaches have already been introduced: *claim detection*, *document retrieval*, *evidence extraction*, *stance detection*, and *claim validation*. These approaches process raw text and thus use the information contained in large document collections, such as Wikipedia, Clue Web[23], or the entire web. A brief overview of these approaches is given below.

Claim detection is the problem of identifying claims in a given text (Levy et al., 2014; Vlachos and Riedel, 2015; Stab and Gurevych, 2017; Konstantinovskiy et al., 2018; Aharoni et al., 2014; Lippi and Torroni, 2016a; Atanasova et al., 2018). For this purpose, lexical, structural, or contextual features can be used (Levy et al., 2014; Stab and Gurevych, 2017). In the field of *argumentation mining*, where argument components including claims are identified in a given text, the problem is often defined as sequence labeling. In this problem setting, a sequence of tokens is classified using the Beginning-Intermediate-Outside (BIO) scheme (Habernal and Gurevych, 2017). The problem is often tackled using LSTM based network architectures (Eger et al., 2017) that reach high performance on this task.

Claim detection is relevant for the second step of the presented fact-checking process (Figiure 1.3), where the fact-checker needs to identify factual statements in the text before they are validated.

Document retrieval is a sub-field of *information retrieval* (Manning et al., 2010; Baeza-Yates et al., 2011) and refers to the problem of identifying documents relevant to a given query. The problem is usually solved in two steps: In the first step, a document collection is retrieved based on the token overlap with the query (inverted index). In the second step, the identified documents are ranked according to their relevance to the query. Traditional approaches to ranking are based on *Okapi BM25* (Robertson et al., 1995), which is a method based on measuring the lexical overlap using Term Frequency-Inverse Document Frequency (TF-IDF), or on the *Page Rank* algorithm (Page et al., 1999), which utilizes the information about the cross-references between documents. Modern approaches, such as *Semantic Search* (Bast et al., 2016) or Google's *Rank Brain*[24], make use of machine learning approaches or/and explicit semantic representations from knowledge bases to compare the query and the documents in the ranking process.

[23] `https://lemurproject.org/clueweb12/`

[24] `https://moz.com/learn/seo/google-rankbrain`

Document retrieval is one of the central tools in fact-checking because it allows the fact-checker to query a large collection of documents or the entire Internet to find information. Thus, it is important in the first step of the fact-checking process (Figure 1.3), where the fact-checker needs to find information about the topic of the text to be validated, and in the third step, where evidence for the claims needs to be found. In Chapter 6 of this thesis, the topic of information retrieval is discussed in more detail.

Stance detection is the problem of identifying the stance of a piece of text with respect to another piece of text. Much work in stance detection is focused on target-specific stance prediction in which the stance of a piece of text with respect to a topic or a named entity is determined; for instance, see stance detection for tweets (Mohammad et al., 2016; Augenstein et al., 2016; Zarrella and Marsh, 2016) or online debates (Walker et al., 2012; Somasundaran and Wiebe, 2010; Sridhar et al., 2015). In the context of fact-checking, it is of interest to determine the relation between the retrieved documents or the evidence and the claim. This problem setting is similar to the task defined by Pomerleau and Rao (2017), where the stance of a document with respect to a headline needs to be classified, or the task formulated by Ferreira and Vlachos (2016), where the stance of the headline of an article towards a claim needs to be determined. In argumentation mining, the relation between the claim and a premise is typically classified as *support* or *attack* (Stab and Gurevych, 2017), which is also related to the stance detection problem in fact-checking. Annotation schemes used in stance detection often consider three classes: a favorable stance *(support, agree)*, an unfavourable stance *(attack, refute, disagree)*, and a neutral stance *(no stance, discuss, neutral, unrelated, etc.)*

The stance detection task is not explicitly contained in the manual fact-checking process described above. However, when a fact-checker collects evidence or documents for a claim, they are certainly aware of the stance of the evidence, as this information is essential for the subsequent validation of the claim in step five. Thus, in this thesis, stance detection is considered as an important step in the automated fact-checking process. This task will be discussed in Chapter 5 in more detail.

Evidence extraction is an information retrieval problem. In this problem setting, one is given a query in the form of a claim or a hypothesis and needs to find propositions, phrases, sentences, or paragraphs in a document that support or refute the claim or hypothesis. The problem can be approached either as a ranking or as a classification task. In the ranking problem setting, one needs to rank potential evidence according to its relevance to the claim (see, for instance, the sentence ranking problem in the FEVER shared task (Thorne et al., 2018a)). This task is similar to *community-based question answering*, where candidate answers need to be ranked according to their usefulness in answering the question (Feng et al., 2015; Rücklé and Gurevych, 2017). In the classification problem setting, one needs to classify whether a piece of text contains valuable information for the validation of the claim. This task is related to the classification of relations between *claims* and *premises* in the field of argumentation mining (Stab and Gurevych, 2017).

Evidence extraction is relevant for the third step of the presented fact-checking process (Figure 1.3), where the fact-checker needs to find evidence for a given claim.

A more detailed discussion of the evidence extraction problem settings is given in Chapter 6.

Claim validation is the problem of determining the veracity of a claim, or alternatively, the prediction of the verdict for a claim. The predicted verdicts typically range on a scale from *true* to *false*, with intermediate labels such as *mostly true* or *mostly false*. To account for cases, in which the claim cannot be distinctly classified as true or false, a neutral verdict, such as *not enough info*, is given.

A number of tasks defined in the literature correspond to claim validation, even though they are often named differently: claim verification (Vlachos and Riedel, 2015), fact-checking (bar), rumor veracity prediction (Derczynski et al., 2017), credibility assessment (Popat et al., 2017), fact verification/Recognizing Textual Entailment (RTE) (Thorne et al., 2018a), or Natural Language Inference (NLI) (Thorne and Vlachos, 2019). As some of the works already suggest, the claim validation problem is related to RTE (Dagan et al., 2005) or NLI (Bowman et al., 2015). In RTE or NLI, one needs to only determine whether the *hypothesis* (claim) can be logically deduced from the *premise* (evidence). However, claim validation is more challenging because the verdict for a claim needs to be determined on the basis of a number of evidence pieces, some of which can originate from unreliable sources. In such cases, the evidence can be contradictory and only evaluating the entailment relation may not be sufficient.

Claim validation corresponds to the fifth step of the proposed fact-checking process (Figure 1.3), where the fact-checker validates the claims identified in a text. The automated claim validation problem setting is discussed in depth in Chapter 7 of this thesis.

Automating the remaining tasks in the fact-checking process. In step four, the credibility of the sources of the evidence needs to be determined. The problem is difficult because even generally trustworthy sources, such as the New Your Times or the Washington Post, have been proven wrong by fact-checkers in a number of cases. Moreover, articles often include quotes from other sources that have a different credibility compared to the credibility of the publisher of the article. Thus, the information within the document needs to be examined, and individual statements need to be assigned to the correct source, which is a task difficult to accomplish automatically. Nevertheless, there are a number of works that try to tackle the problem. Often, simple heuristics, such as the Page Rank and the Alexa Rank [25] of web sources (Popat et al., 2016), are used to estimate the credibility of the source of a document. A more elaborated approach for the problem was developed by Pasternack and Roth (2013). They proposed a probabilistic graphical model that predicts the credibility of the source and the veracity of the claims made by the source.

A number of steps in the described fact-checking process are difficult to automate, and at least to our knowledge, there are very few methods in the literature to tackle these tasks.

[25] Alexa Rank or Alexa Traffic Rank is a metric to measure the popularity of websites `https://www.alexa.com/`

For step six in the fact-checking process, the entire reasoning chain of the article needs to be assessed: that is, whether the conclusions made by the author logically follow from the validated claims in the article. For this purpose, multi-hop reasoning over the claims of the article needs to be performed. For example, after the claims *"The German car manufacturers have announced that they are planning to lay off thousands of workers."* and *"The German Ifo business climate index has decreased three months in a row."* have been validated as true, it needs to be assessed, whether the conclusion made by the author *"The German economy is slowing down and is at risk of recession."* logically follows from the two claims. This is notoriously difficult to accomplish for raw text using machine learning approaches (Marasovic, 2018; Marcus, 2018). Step seven would require some form of automated *fallacy detection*, which is also a difficult task. To our knowledge, only a few studies have explored this problem (see for instance (Habernal et al., 2018a,c)).

Pipeline approaches to automated fact-checking

Even though all steps in the fact-checking process cannot be automated, by combining systems that can solve some of the tasks with reasonable accuracy, it is possible to construct a pipeline to automate at least a part of the fact-checking process. In fact, a number of such fact-checking pipelines have been proposed in the literature. To reduce the complexity of the addressed problem setting, most of the pipelines are designed to validate statements, such as claims, rumors, or alleged facts, instead of validating entire articles. Below, we present some of the most popular pipeline approaches and analyze their strengths and weaknesses.

RumourEval. RumourEval was a shared task in SemEval-2017, which was concerned with "*determining rumour veracity and support for rumours*" (Derczynski et al., 2017). The shared task focused on validating information on Twitter in two steps. Given a rumorous thread, first, the stance of the individual tweets with respect to a rumor had to be identified. Next, the veracity of the rumor had to be determined on the basis of the tweets in the thread, which corresponds to *claim validation*. The defined problem setting represents a two-step pipeline.

The drawbacks of this approach are that (1) it is only suitable for validating claims on Twitter and (2) the systems trained on the provided dataset are unlikely to generalize to other domains. In fact, domain transfer is often only feasible for relatively similar datasets, i.e., a model trained on one dataset and applied to the same (or similar) task on a different dataset achieves a substantially lower performance (see for instance Daxenberger et al. (2017)).

Moreover, the pipeline only contains two steps, and it is therefore expected that the performance can be further increased if the system is extended by incorporating a document retrieval module and an evidence extraction module. This would allow the system to additionally extract information from the web to increase the performance for claim validation (determining the veracity of the rumor). Tweets provide relatively little information, and it is expected that the additional information would help to validate the more difficult cases.

CLEF-2018. Nakov et al. (2018) organized the CLEF-2018 fact-checking task, which also corresponds to a two-step pipeline approach. In the first sub-task, participants had to identify “check-worthy” claims in a political debate, which corresponds to *claim detection*. In the second sub-task, the “factuality” of the identified “check-worthy” claims had to be determined, which corresponds to the *claim validation* step. However, the participants had to develop their own systems for evidence retrieval if they intended to use external knowledge to validate the claims. Similar to the pipeline approach proposed in RumourEval, the CLEF-2018 two-step approach is restricted. For both systems, additional supervision for the development of the evidence retrieval model would be helpful but is not provided.

Where the Truth Lies 2017. Popat et al. (2017) developed a pipeline system for validating emerging claims on the web and social media. The developed pipeline consists of four steps, *document retrieval*, stance determination (*stance detection*), credibility assessment (*claim validation*), and evidence presentation (*evidence extraction*). The system is superior to RumourEval and CLEF-2018 because the proposed pipeline can access information from the web in order to validate the claims.

However, even though the system also provides evidence to the user at the end of the pipeline, it is not guaranteed that the predicted verdict is based on this evidence, as the evidence extraction step and the claim validation step are decoupled. Furthermore, Popat et al. (2017) use the Google search engine for the document retrieval step. It is not optimized for the identification of evidence and can therefore lead to low performance. As we show in Section 4.4, traditional document retrieval approaches can be inferior to methods tailored for automated fact-checking.

ClaimBuster. ClaimBuster (Hassan et al., 2017) is an automated fact-checking pipeline designed to validate web documents in four steps: (1) Claim Monitor: Using *document retrieval*, web documents are downloaded from the web. (2) Claim Spotter: In the retrieved documents, claims are identified, which corresponds to *claim detection*. (3) Claim Matcher: The identified claims are matched to those for which a verdict has already been determined to avoid validating the same claim again. (4) Claim Checker (*evidence extraction* and *claim validation*): For the identified claims, supporting and refuting evidences are aggregated from knowledge bases as well as from the web using *document retrieval*. Then, the actual claim validation is performed: “If any clear discrepancies between the returned answers (evidence) and the claim exist, then a verdict may be derived and presented to the user.”(Hassan et al., 2017).

The ClaimBuster system is comprehensive and includes a claim matching module that could save the user some time when already validated claims are encountered. However, the claim validation part of the system is not very elaborated, as only the “discrepancy” between a given claim and the evidence is determined. Machine-learning-based classifiers, on the contrary, achieve better performance in such problem settings. Moreover, the stance information about the evidence is missing, which does not allow the user to observe the relation between the evidence and the claim.

FEVER 2018. The FEVER shared task (Thorne et al., 2018b) was designed to foster the development of comprehensive fact-checking pipelines on the basis of a large dataset with 185,445 validated claims. Participants were asked to develop pipeline systems consisting of three steps: *document retrieval*, sentence selection (*evidence extraction*), and recognizing textual entailment (*claim validation*). The shared task organizers provided a dataset with supervision for the three tasks, which is based on Wikipedia. The shared task attracted a relatively large number of participants, with 23 competing teams. Even though large performance gains have been achieved in the shared task, the developed systems do not generalize well to other domains (see Section 7.4.3). Because the entire dataset is only based on Wikipedia, the articles do not substantially vary in style and are identically structured. Moreover, because the claims are also extracted from Wikipedia, the systems did not have to deal with unreliable sources, contradicting evidence, or different writing styles from heterogeneous web sources, as in real life fact-checking. This topic is discussed in more depth in Section 7.4.3.

There are a number of other systems for fact-checking, which are similar to the presented pipeline approaches; therefore, they are not discussed in further detail (Nadeem et al., 2019; Hassan et al., 2015; Shao et al., 2016).

Discussion. The presented pipeline approaches are important contributions to automated fact-checking, as they helped to formalize the problem by dividing it into a number of sub-tasks, and report first results for these sub-tasks. Nevertheless, as discussed above, the approaches have a number of drawbacks: (1) Important steps in the pipeline are missing, and no annotation for these parts is provided. In particular, many pipelines lack modules for document retrieval and evidence aggregation. (2) Systems are mostly tailored to a single domain because they are trained on single-domain corpora. They are therefore unlikely to generalize to heterogeneous web sources, on which false information emerges. Because of these drawbacks, the proposed pipelines are only partly applicable to real fact-checking instances. More discussion on this topic is given in Section 7.4.

To address these problems, in this thesis, an alternative pipeline approach for fact-checking is proposed (Section 2.2). Our pipeline includes the most crucial steps of the fact-checking process: *document retrieval*, *evidence extraction*, and *claim validation*. To increase transparency, we include a *stance detection model* that allows us to classify the stance of the retrieved documents and the evidence with respect to a claim. To enable our pipeline system to generalize across different types of text, our system is developed on the basis of a heterogeneous corpus from the web presented in Chapter 3. The corpus provides not only the annotation of the verdicts for the claims but also the annotation of evidence and documents containing the evidence.

1.2.3 Tasks related to automated fact-checking

A of tasks are often associated with automated fact-checking: *fake-news detection*, *neural fake-news detection*, *computational argumentation/argumentation mining*, and *interactive evidence detection*. To highlight the differences as well as similarities of these tasks to automated fact-checking, a brief overview of these tasks is given below.

Fake-news detection

Motivated by the recent increase in the amount of false-news articles on the web, a considerable number of studies have been devoted to the problem of determining whether a news article is reliable or not. We summarize this kind of work under the name *fake-news detection*.

In these studies, datasets with different types of articles are provided: mainstream, left-wing, and right-wing news articles (Potthast et al., 2018); legitimate news and fake news (Pérez-Rosas et al., 2018); real stories, fake stories, and satire stories (Rubin et al., 2016; Horne and Adali, 2017); and real news, satire, hoaxes, and propaganda (Rashkin et al., 2017).

For the classification of the articles, many different features have been proposed: n-grams, characters, stop words, part-of-speech tags, readability scores, term frequency, syntactic features, features from the linguistic inquiry and word count (LIWC) dictionary (Tausczik and Pennebaker, 2010), named entities, grammatical features, punctuation, and others (Potthast et al., 2018; Pérez-Rosas et al., 2018; Rubin et al., 2016; Horne and Adali, 2017; Rashkin et al., 2017).

The described features represent only the internal characteristics of the articles. More sophisticated approaches analyze the proliferation characteristics of the news articles on social networks and use this information as an additional feature for the classification (Tacchini et al., 2017; Volkova et al., 2017; Farajtabar et al., 2017; Monti et al., 2019; Zhou and Zafarani, 2018; Ruchansky et al., 2017).

Even though increasingly more complex methods are used for fake-news detection, there is a fundamental drawback to these methods: because fake-news detection methods are based on shallow stylistic and lexical features or/and proliferation characteristics of the articles on social media, they are more likely to learn to differentiate between different genres of text rather than to keep reliable and unreliable articles apart. Tabloid press articles are, for instance, more likely to be of the same writing style as false-news articles. It would be therefore be more difficult to differentiate between them only on the basis of stylistic features or distribution patterns on social media. Moreover, it is likely that the trained systems can be misled by news articles that mimic the style of an established news magazine but contain only false information.

Thus, from our perspective, these methods do not address the problem in depth. We therefore clearly differentiate between *fake-news detection* and *automated fact-checking*, which we consider to be a more promising approach. Automated fact-checking allows for the identification of claims in the articles and the validation of the claims on the basis of evidence stemming from external sources. As a result, a verdict for the entire article can be derived on the basis of the truthfulness of all the claims in the article. This is expected to be more fruitful than the classification of the entire document mostly on the basis of shallow linguistic features, without considering the propositional content of the document and without consulting external sources.

Neural fake-news detection

Deep generative models trained in a self-supervised manner on massive corpora (Radford et al., 2019; Dai et al., 2019; Zellers et al., 2019) have recently achieved the capability of generating long coherent text. Because entire text documents can now be automatically generated, there is increasing concern over the misuse of such models for the automatic generation of fake-news articles.[26] Because such models have only recently been developed, research on how to detect automatically generated text is limited. However, first results in this research area suggest that the same deep generative models can be leveraged to identify automatically generated fake content (Zellers et al., 2019). In fact, as shown in the same study, methods used for fake-news detection appear to not be reliable enough to solve the problem and are inferior to the deep generative model classifier.

Computational argumentation and argumentation mining

Argumentation is a multidisciplinary field concerned with the analysis of debating and reasoning processes (Lippi and Torroni, 2016b). Owing to the recent advances in machine learning and natural language processing, computational models for argumentation and automated reasoning are becoming more feasible, which has given rise to *computational argumentation* (Slonim et al., 2016; Atkinson et al., 2017). The data required for computational argumentation is available on the web in the form of online newspapers, debate sections of news articles, product reviews, or blogs. For the identification of argument components in raw text, a number of strategies have been developed, and the corresponding field of research has become known as *argumentation mining* (Palau and Moens, 2009; Mochales and Moens, 2011; Lawrence and Reed, 2020). Lippi and Torroni (2016b) define argumentation mining as follows: *"The main goal of argumentation mining is to automatically extract arguments from generic textual corpora, in order to provide structured data for computational models of argument and reasoning engines"* (Lippi and Torroni, 2016b, p. 2).

For modeling argumentation structures in text, many different annotation schemes have been proposed (see for instance Lippi and Torroni (2016b); Stab and Gurevych (2014); Stab et al. (2018a)). Most of them include the annotation of *claims* and *premises* and the definition of *attack* or *support* relations between the premises and the claims.

The annotation schemes are to some extent similar to the terminology used in automated fact-checking (see Section 1.2.2). In both frameworks, claims are considered and *premises* to some extent resemble *evidence*. However, important differences exist. Argumentation mining is in general concerned with the identification of the argumentation structures in the text and not with the validation of the identified arguments. Typical text sources, on which argumentation mining is applied, are debate forums, social media discussions, or political debates (Gurevych et al., 2016; Stab and Habernal, 2016; Habernal et al., 2018b). The discussions are therefore often centered around controversial topics, such as nuclear energy, homeschooling, or abortion, for which valid supporting or attacking arguments can be brought forward, but for which, in general, no conclusive verdicts can be determined. Nuclear energy, for instance, has its upsides and downsides, and whether it causes more harm than good to our society cannot be easily validated. In fact-checking, however, as we define it in Section 1.2.2, the subject of the discussion must be factual, and one should be able to objectively assess the issue.

[26] `https://openai.com/blog/better-language-models/`

Interactive evidence detection

There is a line of research devoted to *interactive evidence detection for hypotheses validation in humanities* (Stahlhut et al., 2018; Stahlhut, 2019) that is related to our definition of automated fact-checking. This work is meant to assist researchers in humanities in their investigations. In particular, when humanities researchers analyze a text on a particular topic, they formulate hypotheses about the topic and then collect evidence that strengthens their hypotheses or refutes them. This kind of work is very laborious because in order to find evidence, the researcher needs to go through large collections of text.

Research in *interactive evidence detection* aims to automate or at least to speed up the process of evidence detection and hypotheses validation. The problem setting is similar to automated fact-checking, as for both tasks, evidence needs to be identified in a large collection of text. Moreover, *hypotheses* to some extent correspond to *claims*, as they represent an assertion that is not yet proven to be true. Consequently, *hypotheses validation* and *claim validation* are similar problems. Nevertheless, important differences between automated fact-checking and interactive evidence detection exist. As shown in (Stahlhut et al., 2018), there is no consensus among humanities researchers regarding what constitutes a hypothesis and what constitutes a piece of evidence. Thus, it is not possible to design annotation guidelines that clearly describe both concepts. As a result, there is no *inter-annotator agreement* between researchers in evidence detection and hypotheses validation studies (Stahlhut et al., 2018). This is in strong contrast to our annotation framework (see Section 3.2.1), in which we provide a number of clearly defined rules describing what properties a piece of text needs to have to be considered as evidence.

Because there is no inter-annotator agreement among humanities researchers, it is not possible to design an evidence detection model for hypotheses validation that fits all needs. Instead, Stahlhut (2019) have designed an interactively trained evidence detection model that adapts to the user's needs and learns to detect such kind of evidences as they are defined by the individual users. This approach is in contrast to our framework presented in Chapter 6, where we propose a generally applicable evidence extraction model for automated fact-checking that is independent of the user.

1.3 Chapter summary

In this chapter, we gave a comprehensive overview of the fact-checking problem on the basis of the literature on this topic. Because the knowledge about the characteristics of the false information is important for fact-checking, we first discussed the different kinds of false information and the proliferation of false information through social media. We categorized false information into white, gray, and black propaganda and examined the properties of each type of false information. We have argued that whereas different kinds of false information distributed on the web can be assigned to different categories of propaganda, false-news articles are typically a kind of black propaganda that is the most severe form of deception. The proliferation of false information was investigated in a number of sub-sections, each of which discussed a specific aspect of the proliferation process. We have described bad actors who distribute false information, analyzed the dynamics of information distribution on the web in general, and investigated why people are susceptible to false information. We have pointed out that disinformation on the Internet is successful because of the characteristics of the media landscape, such as its dependence on novelty and sensationalism, and psychological factors, such as our prejudices towards people with different political views.

After analyzing the nature of false information, we have discussed the manual fact-checking process and introduced the internal and external fact-checking processes. The internal fact-checking process was presented as the internal quality assurance process within news magazines. The external fact-checking process was defined as the validation process practiced by the growing number of fact-checking platforms on the web. We then described the manual validation process followed by many internal and external fact-checkers in more detail. We remarked that whereas manual fact-checking is still very popular, it is unlikely that this kind of information validation alone would enable us to control the spread of false information, and we argued that automated approaches are required to address the problem.

Next, we gave an overview of the field of automated fact-checking. We first introduced the terminology that is typically used in automated fact-checking and that we have adopted for this thesis. Thereafter, we analyzed traditional fact-checking systems based on knowledge bases. We identified the low coverage of the knowledge bases as a major drawback of these approaches and introduced machine-learning-based systems for different tasks in the fact-checking process as an alternative because they can harness the large amount of textual information on the web. We then described pipeline approaches that combine several machine-learning-based systems to automate parts of the fact-checking process. We concluded the chapter with a discussion of NLP tasks related to automated fact-checking: fake-news detection, neural fake-news detection, argumentation mining, and interactive evidence detection. We pointed out that whereas argumentation mining targets a different problem compared to automated fact-checking, namely the analysis of the discussion of a controversial topic instead of the validation of factual claims, automated fact-checking approaches can be leveraged for fake-news detection. In fact, automated fact-checking can be used to validate an article by assessing the veracity of the claims in the article. Interactive evidence detection is related to automated fact-checking, as for both tasks, evidence needs to be aggregated. Nevertheless, whereas a developed automated fact-checking pipeline should be generally applicable by most fact-checkers, interactive evidence detection systems are designed to adapt to the needs of an individual user in the hypotheses validation process.

Chapter 2

Contributions of the thesis

Despite significant progress in the area of automated fact-checking, the current fact-checking systems still have a number of major shortcomings that need to be addressed. After analyzing the state-of-the-art in this field, we have identified the following main challenges: (1) The datasets available for training machine learning models for fact-checking do not provide high-quality annotations of real fact-checking instances for all the tasks in the fact-checking process. (2) Automated fact-checking approaches based on knowledge bases are restricted by the low coverage of the knowledge bases and are error-prone because statements in natural language need to be converted into formal knowledge-base queries. (3) Current end-to-end trained machine learning systems can process raw text and thus can potentially make use of the vast amount of knowledge on the Internet. However, such systems are intransparent, and the reason for a prediction cannot be easily determined. Moreover, end-to-end systems do not reach the desired performance because today's machine learning techniques are not mature enough to solve the fact-checking problem without human assistance.

The validation process is very challenging, and a fact-checking system needs to possess a number of important abilities to solve this task: *common-sense reasoning*, which is the ability to foresee outcomes of actions, such as what is going to happen if we suddenly remove the table under a vase; *world knowledge*, which is knowledge about the relations of objects in the world, e.g., a pencil is smaller than the Eiffel Tower and whereas the former fits into a handbag, the latter does not. A system should be able to infer such facts without being explicitly programmed to know all possible relations between objects. Because these abilities are still unresolved problems in the AI research community, the development of a fully automated fact-checking system is a very difficult task. To address the identified challenges, in this thesis, we make the following two main contributions:

I Corpus work. To address the lack of an appropriate dataset for training machine learning systems for automated fact-checking, we introduce a new richly annotated corpus on the basis of the Snopes fact-checking website. The corpus provides high-quality annotations for the different sub-tasks in the fact-checking process (Section 1.2.1). The corpus is based on information from real fact-checking instances aggregated from heterogeneous web sources, many of which are the origin of the false information. In addition to the corpus, we publish our corpus creation methodology that allows for efficiently creating large datasets with high inter-annotator agree-

ment.
II Methodological contribution. To address the need for automated fact-checking methods that do not suffer the drawbacks of previous approaches, we propose a novel *fact-checking pipeline* consisting of several sub-systems for validating textual claims. In contrast to many other fact-checking systems, our pipeline approach is able to process raw text and thus make use of the vast amount of textual information available on the web, but at the same time, the pipeline is transparent, as the outputs of the sub-systems can be observed. Correctly completed sub-tasks do not need to be performed manually: potential errors can be traced back to their origin, and the predictions can be revised by the fact-checker. In fact, because we believe that fact-checking is an AI-complete problem, meaning that human-level intelligence is required to perform the task, the objective of this thesis is to develop a system that assists the fact-checker in the validation process, to speed up the procedure rather than taking over the task entirely.

In this chapter, we first give a detailed overview of all the contributions of the thesis and present a number of research questions that this thesis sets out to answer. Thereafter, we present our fact-checking pipeline for validating textual claims. The chapter is concluded with the publication record and the thesis outline.

2.1 Contributions and research questions

For a better overview of the individual contributions of this thesis, we illustrate the relations of the contributions in Figure 2.1. The contributions are closely related to the research questions presented in this section, as both contributions address open problems in designing machine learning systems for automated fact-checking and constructing corpora for training such systems.

2.1.1 Contributions

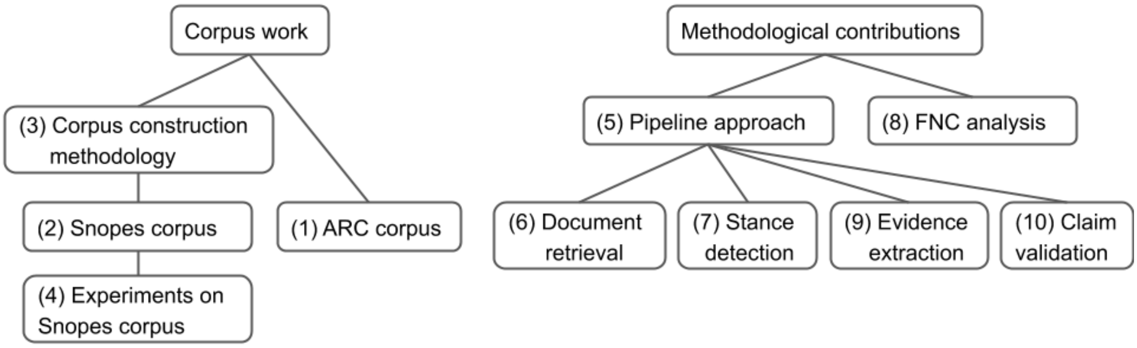


Figure 2.1: Overview of the contributions

I Corpus work.
(1) We introduce a new stance detection corpus (ARC) by modifying the argument reasoning comprehension dataset introduced by Habernal et al. (2018b). On the

basis of the new dataset and the Fake News Challenge[1] corpus, we conduct cross-domain experiments with models from the Fake News Challenge to evaluate their generalization capabilities.

(2) We construct a new, richly annotated corpus that is based on real fact-checking instances from the Snopes fact-checking website and provides annotations for training machine learning models for different sub-tasks in the fact-checking process.

(3) In addition to the corpus, we present our corpus creation framework for the efficient construction of fact-checking corpora with high inter-annotator agreement.

(4) We perform a large number of experiments on our newly created Snopes corpus with models that we have introduced and other successful approaches suitable for the different fact-checking sub-tasks. Based on these experiments, we conduct a detailed analysis of the fact-checking problem setting defined by our Snopes corpus and compare it with the fact-checking problem defined by the FEVER shared task corpus.

II Methodological contributions.

(5) We propose a novel automated fact-checking pipeline consisting of four sub-systems: *document retrieval*, *stance detection*, *evidence extraction*, and *claim validation.*

(6) We introduce a novel document retrieval system for retrieving documents for fact-checking from the Wikipedia FEVER shared task corpus. Our system is based on entity linking and hand-crafted rules, and it substantially outperforms traditional document retrieval systems based on Term Frequency - Inverse Document Frequency (TF-IDF) on the FEVER Wikipedia corpus.

(7) We propose a stance detection system based on a deep Multi-Layer Perceptron (MLP) and hand-crafted features. The system can determine the stance of a document with respect to a given headline and reaches high performance in the Fake News Challenge problem setting.

(8) We critically assess the top three systems of the Fake News Challenge and the Fake News Challenge problem setting itself. We systematically evaluate the performance of the features used by the top three systems as well as a set of novel features. Based on the insights of the analyses, we propose a new metric for the Fake News Challenge datasets and two new models for the task.

(9) We propose an evidence extraction model for extracting sentence-level evidence from documents for a given claim. The model is based on the ESIM (Chen et al., 2017b) and a ranking loss objective function. The evidence extraction model combined with our document retrieval system outperforms other competitive systems on the evidence identification sub-task of the FEVER shared task.

(10) We present our claim validation model that reaches the third rank for *recognizing textual entailment* in the FEVER shared task and analyze the performance of the model in an error analysis.

[1] `http://www.fakenewschallenge.org/`

2.1.2 Research questions

Besides constructing a new corpus and developing a pipeline approach for automated fact-checking, in this thesis, we answer a number of research questions that are important for solving the fact-checking problem:

- Is it possible to design an annotation framework for the annotation of evidence in documents for a given claim and for the annotation of the stance of the evidence with respect to the claim that leads to high inter-annotator agreement?
- If we have stance annotated evidence for a claim (in the form of text snippets that support or refute the claim), is it possible to validate this claim only on the basis of the number of *evidence text snippets* and their stances without considering the textual content of the claim and the evidence?
- How well can current machine learning models for fact-checking, that perform well on existing datasets covering a single domain, generalize to multi-domain datasets, that is, can we achieve a high performance for automated fact-checking on a multi-domain dataset if the system is only trained on a single-domain corpus?
- In most of the fact-checking problem settings defined so far, evidence for the validation of a claim is only provided in the form of one or several sentences. Is this information sufficient, or do we need additional contextual information to reach high performance on the claim validation sub-task?

2.2 A pipeline approach to automated fact-checking

Our fact-checking pipeline is designed for the validation of textual claims, as these are at the core of most of the false information distributed on the web. Moreover, such a system can be used to validate larger pieces of text, such as an elaborated news story. Using a claim detection approach, the claims in the article can be identified. The *truthfulness* (or veracity) of the article as a whole can then be determined on the basis of the truthfulness of the claims in the article.

For the validation of a given claim, we propose a pipeline consisting of four steps, which is illustrated in Figure 2.2. The four sub-steps of the pipeline are briefly discussed below.

Step 1: Document retrieval. Given a claim, the system retrieves documents that contain relevant information for the validation of the claim.
Step 2: Stance detection. In this step, the stance of the retrieved documents with respect to the given claim is determined. Documents can be classified as either *supporting* or *refuting* the claim or having *no stance* with respect to the claim.
Step 3: Evidence extraction. On the basis of the given claim, sentence-level evidence that contains important information for the validation of the claim is extracted from the retrieved documents.

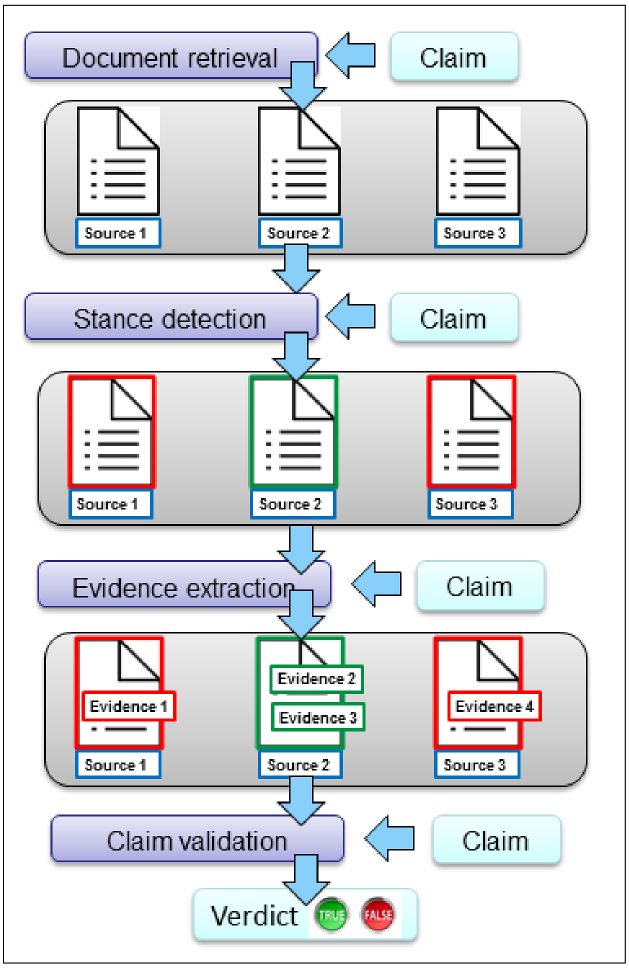


Figure 2.2: Our pipeline approach to automated fact-checking

Step 4: Claim validation. In this step, the verdict (label) for the claim is determined on the basis of the collected evidence. The claim can be classified not only as *true* or *false* but also as *not enough info*, if the evidence is not sufficient to label the claim as true or false with high confidence (see classification framework proposed in (Thorne et al., 2018a)).

In the subsequent chapters of this thesis, the individual steps of the pipeline are discussed in detail. In these chapters, a deeper analysis of each sub-task is given, and novel machine learning techniques to tackle the sub-tasks are presented.

2.3 Publication record and thesis outline

This thesis is based on the contributions of the following four publications:

Andreas Hanselowski and Iryna Gurevych: *A Framework for Automated Fact-Checking for Real-Time Validation of Emerging Claims on the Web*. In Proceedings of the 2017 NIPS Workshop on Prioritising Online Content, Long Beach, Los Angeles, CL, USA, 2017. (**WPOC2017**) https://www.k4all.org/wp-content/uploads/2017/09/WPOC2017_paper_6.pdf

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych: *A Retrospective Analysis of the Fake News Challenge Stance Detection Task.* In Proceedings of the 27th International Committee on Computational Linguistics, pages 1859–1874, Santa Fe, NM, USA, 2018. (**COLING2018**)
https://www.aclweb.org/anthology/C18-1158.pdf

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych: *UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification.* In Proceedings of the EMNLP 2018 First Workshop on Fact Extraction and Verification, pages 103-108, Brussels, Belgium, 2018. (**FEVER2018**)
https://www.aclweb.org/anthology/W18-5516.pdf

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych: *A richly annotated corpus for different fact-checking sub-tasks.* In Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 493-503, Hong Kong, 2019. (**CoNLL2019**)
https://www.aclweb.org/anthology/K19-1046.pdf

The following two publications have been completed in the course of the doctoral program, but their content is not included in the thesis.

Christopher Tauchmann, Thomas Arnold, Andreas Hanselowski, Christian M. Meyer, and Margot Mieskes: *Beyond generic summarization: A multifaceted hierarchical summarization corpus of large heterogeneous data.* In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, pages 3184-3191, Miyazaki, Japan, 2018. (**LREC2018**)
https://www.aclweb.org/anthology/L18-1503.pdf

Norbert Fuhr, Anastasia Giachanou, Gregory Grefenstette, Iryna Gurevych, Andreas Hanselowski, Kalervo Järvelin, Rosie Jones, YiquN Liu, Josiane Mothe, Wolfgang Nejdl, Isabella Peters, and Benno Stein: *An Information Nutritional Label for Online Documents.* In Special Interest Group on Information Retrieval Forum, volume 51, number 3, pages 46-66, 2017. (**SIGIR2017**)
https://www.aclweb.org/anthology/W18-5505.pdf

The thesis outline is given below. The contributions of the four publications are presented in five chapters:

Chapter 3 is concerned with corpora for training automated fact-checking systems. We give a comprehensive overview of existing fact-checking corpora and highlight their strengths and weaknesses. We also introduce a new stance detection dataset by modifying the argument reasoning comprehension corpus introduced by Habernal et al. (2018b) (Section 3.1.2) (contribution 1). We construct a new substantially-sized corpus, with high-quality annotations for the different fact-checking sub-tasks (contribution 2). In addition to the corpus, we present our methodology for constructing and annotating corpora, which allows efficiently creating and annotating corpora (contribution 3). The corpus and the corpus creation methodology are published in **CoNLL2019**.

In Chapter 4, we present a document retrieval system for retrieving documents for fact-checking from the Wikipedia FEVER shared task corpus (contribution 6). Our system is based on entity linking and hand-crafted rules and substantially outperforms traditional document retrieval systems based on Term Frequency - Inverse Document Frequency (TF-IDF) on the FEVER Wikipedia corpus. Our document retrieval system is published in **FEVER2018**.

Chapter 5 is concerned with stance detection and the Fake News Challenge. We introduce our stance detection system for determining the stance of a document with respect to a given headline, which was deployed in the Fake News Challenge (contribution 7). We analyze the top three systems of the Fake News Challenge, as well as the Fake News Challenge problem setting itself (contribution 8). We also evaluate the performance of the features of the top three systems and the performance of a new feature set. On the basis of the analyses, we propose two new models. We test the generalizability of the top three models from the challenge and our two newly proposed models, by evaluating their performance on a second corpus that we created by modifying the argument reasoning comprehension dataset. The proposed stance detection system and the conducted experiments and analyses are published in **COLING2018**. In this chapter, we also perform stance detection experiments on our Snopes corpus with systems that reach high performance in similar problem settings and conduct an error analysis for the best performing system (contribution 4). These experiments and the error analysis are published in **CoNLL219**.

In Chapter 6, we present an evidence extraction model for extracting sentence-level evidence from documents for a given claim and perform experiments with this model on the FEVER shared task corpus (contribution 9). The system and the experiments are published in **FEVER2018**. We perform evidence extraction experiments with a number of systems on our Snopes corpus and conduct an error analysis (contribution 4). The experiments on the Snopes corpus and the error analysis are published in **CoNLL2019**.

In Chapter 7, we introduce our model for claim validation that is able to validate a claim on the basis of an arbitrary number of evidence sentences and present experiments with this system on the FEVER corpus (contribution 9). The system and the results of the experiments are published in **FEVER2018**. We perform a large number of experiments for the claim validation problem of our Snopes corpus with models from the FEVER shared task and other successful models suitable for the task. Based on the conducted experiments, we analyze the fact-checking problem setting defined by our Snopes corpus and compare it to the fact-checking problem defined by the FEVER shared task corpus (contribution 4). The experiments on the Snopes corpus and the subsequent analyses are published in **CoNLL2019**.

In Chapter 8, we conclude the thesis and present a summary of our contributions and findings. Based on the findings, we answer the research questions posed in the introduction of the thesis. In the final part of the chapter, we discuss promising future research directions in automated fact-checking.

Chapter 3

Corpus construction

For the development of a full-fledged automated fact-checking system based on machine learning, a corpus is required that needs to satisfy certain criteria. It must contain a large number of examples with *high-quality annotations* for the different tasks in the fact-checking process. Since false information can come from different sources, such as discussion forums, news articles or messages on Twitter, the training data should not be limited to a particular domain but cover different text sources varying in genre and writing style. In this chapter, we analyze how far existing corpora satisfy these criteria, and we introduce a new richly annotated corpus to address the drawbacks of the existing datasets.

We divide this chapter into two parts. First, we present existing corpora for training machine learning models for fact-checking. We give an overview of all fact-checking corpora we are aware of and discuss their strengths and weaknesses. Next, we highlight a number of corpora in more detail, which are used for the experiments in the following chapters in this thesis. Thereafter, we present the corpus **ARC2017** that we have created by modifying the corpus introduced by Habernal et al. (2018b).

Second, we introduce a new richly annotated corpus based on the Snopes[1] fact-checking platform, which is more in agreement with the described criteria. In addition to the corpus, we describe our corpus creation methodology for constructing fact-checking corpora with high inter-annotator agreement.

The contributions of this chapter are the following.[2]

(1) We introduce a new stance detection corpus **ARC2017** by modifying the corpus introduced by Habernal et al. (2018b).

(2) We construct a new richly annotated corpus **Snopes19** based on real fact-checking instances and providing annotations for training machine learning models for different sub-tasks in the fact-checking process.

(3) In addition to the introduced corpora, we present our corpus creation methodology for the efficient construction of fact-checking corpora with high inter-annotator agreement.

[1] `http://www.snopes.com/`

[2] The complete list of contributions ranging from 1 to 10 is given in Section 2.1.1.

3.1 Existing corpora

3.1.1 Overview and discussion of existing corpora

PolitiFact14. Vlachos and Riedel (2014) analyzed the fact-checking problem and constructed a corpus on the basis of the fact-checking blog of Channel 4[3] and the Truth-O-Meter from PolitiFact.[4] In addition to validated claims, the corpus contains evidence, which has been used by fact-checkers to validate the claims, as well as metadata, such as the speaker ID and the date when the claim was made. Since this early work in automated fact-checking, Vlachos and Riedel (2014) mainly focused on the analysis of the task. Thus, the corpus is small in size and only contains 106 validated claims.

Emergent16. A more comprehensive corpus for automated fact-checking was introduced by Ferreira and Vlachos (2016). The dataset is based on the project Emergent,[5] which is a journalist initiative for rumor debunking. The corpus consists of 300 claims, which have been validated by journalists. For each claim, the corpus provides a number of news articles that are related to the claim, giving rise to a collection of 2,595 associated documents. The journalists summarized each article in a headline and annotated the stance of the article with respect to the claim.

FNC2017. Pomerleau and Rao (2017) have altered the corpus **Emergent16** in order to create a dataset for the *Fake News Challenge* (FNC). The FNC was an international competition where the stance of a news article with respect to a headline had to be determined. Since **Emergent16** only contains the annotation of the stance of the headlines of the news articles with respect to claims, the original corpus was modified in order to derive stance labels for the news articles (the modification is described in detail in the following section).

ARC2017. We created this corpus by modifying the argument reasoning comprehension dataset introduced by Habernal et al. (2018b). The original corpus is based on user-posts from the debate section of the New York Times and was designed for the argument reasoning comprehension shared task. The shared task participants had to identify and reconstruct *implicit warrants* for the user-posts in a number of steps, one of which was stance detection. We modified this corpus to adjust it to the FNC stance detection problem setting (the modification is described in detail in the following section).

PolitiFact17. Wang (2017) extracted 12,800 validated claims made by public figures in various contexts from Politifact.[6] For each statement, the corpus provides a verdict and meta information, such as the name of the speaker, subject of the debate, or the party affiliation of the speaker. This corpus has substantially more claims compared to **PolitiFact14**, but it does not contain evidence for the valida-

[3] http://blogs.channel4.com/factcheck/

[4] http://www.politifact.com/truth-o-meter/statements/

[5] http://www.emergent.info/

[6] http://www.politifact.com/

tion of the claims.

RumEval17. Derczynski et al. (2017) have organized the RumourEval shared task, for which they provided a corpus of 297 rumourous threads from Twitter containing 4,519 tweets in total. The shared task was divided into two parts, *stance detection* and *veracity prediction* for the rumors (similar to claim validation). For stance detection, Derczynski et al. (2017) labeled the tweets as *support*, *deny*, *query* or *comment* with respect to the rumour. The 297 rumours (claims) were annotated as *true*, *false*, or *unverified.* The large number of stance annotated tweets allows for training stance detection systems that can reach a relatively high accuracy score of about 0.78. However, since the number of rumors (claims) is relatively small, and the corpus is only based on tweets, this dataset alone is not suitable to train generally applicable fact-checking systems.

Snopes17. A corpus featuring a substantially larger number of validated claims was introduced by Popat et al. (2017). It contains 4,956 claims with their verdicts, which have been extracted from the Snopes website, Wikipedia collection of proven hoaxes[7] and fictitious people.[8] For each claim they retrieved about 30 associated documents from the web using the Google search engine, resulting in a collection of 136,085 documents.

CLEF-2018. Nakov et al. (2018) introduced a corpus concerned with political debates in the course of the CLEF-2018 shared task. The corpus consists of transcripts of political debates in English and Arabic with annotations for two tasks that have been tackled in the competition. In the first task, the participants had to identify check-worthy statements (claims) in the debate, and in the second, 150 statements (claims) from the debates had to be validated.

FEVER18. The FEVER corpus introduced by Thorne et al. (2018a) is the largest available fact-checking corpus, consisting of 185,445 validated claims. The corpus is based on about 50k popular Wikipedia articles, where annotators modified their sentences to create the claims and labeled other sentences in the articles, which support or refute the claim, as evidence. This allows training machine learning models to carry out the three tasks: document retrieval, evidence extraction, and claim validation.

In Table 3.1, we give an overview of all presented fact-checking corpora including our corpus **Snopes19** that we introduce in Section 3.2 of this chapter. We focus on the key parameters: the fact-checking sub-task coverage, availability of annotation, the corpus size, and domain coverage. It must be remarked that a fair comparison between the datasets is difficult to accomplish since the length of evidence and documents, as well as the annotation quality, significantly vary across different corpora.

As discussed in the introduction of this chapter, the training data for the development of a full-fledged fact-checking system needs to satisfy several criteria. Below, we discuss in how far the presented corpora are in agreement with these criteria.

[7] `https://en.wikipedia.org/wiki/List_of_hoaxes#Proven_hoaxe`

[8] `https://en.wikipedia.org/wiki/List_of_fictitious_people`

	claims	docs.	evid.	stance	sourc.	agr.	domain
PolitiFact14 (Vlachos and Riedel, 2014)	106	no	yes	no	no	no	political debates
Emergent16 (Ferreira and Vlachos, 2016)	300	2,595	no	yes	yes	no	news
PolitiFact17 (Wang, 2017)	12,800	no	no	no	no	no	political debates
RumEval17 (Derczynski et al., 2017)	297	4,519	no	yes	no	yes	Twitter
Snopes17 (Popat et al., 2017)	4,956	136,085	no	no	yes	no	Google search
ARC2017 (Habernal et al., 2018b)	376	2,884	no	yes	no	yes	NYT debates
CLEF-2018 (Nakov et al., 2018)	150	no	no	no	no	no	political debates
FEVER18 (Thorne et al., 2018a)	185,445	14,533	yes	yes	yes	yes	Wikipedia
Snopes19 (Hanselowski et al., 2019)	6,422	14,296	yes	yes	yes	yes	multi domain

Table 3.1: Overview of corpora for automated fact-checking. docs: documents related to the claims; evid.: evidence in form of sentence or text snippets; stance: stance of the evidence; sourc.: sources of the evidence; (opinion holder or the URL of the web document from which the evidence was extracted); agr.: whether or not the inter-annotator agreement is reported; domain: the genre of the corpus

The **ARC2017** dataset is based on the corpus (Habernal et al., 2018b) that was designed for the identification of implicit warrants within the area of the *argumentation mining* (see Section 1.2.3). Thus, even though one sub-task for this corpus is stance detection, which is a task useful for automated fact-checking, the corpus as a whole is not suitable for training automated fact-checking systems.

The corpora **PolitiFact14, CLEF-2018** contain claims annotated with verdicts but do not provide annotated evidence for the validation of the claims. This means that for training the evidence extraction component of a fact-checking system, an additional source of supervision is required. Thus, these corpora can only be complementary for the development of a full-fledged fact-checking system.

The corpora **PolitiFact17, Emergent16 (FNC2017), RumourEval17** are valuable for the analysis of the fact-checking problem and provide annotations for stance detection. However, they contain only several hundreds of validated claims and it is therefore unlikely that modern deep learning models can be trained on these corpora.

The corpus **Snopes17** contains significantly more validated claims, but, for each claim, it only provides 30 documents that are retrieved from the web using the Google search engine. Since these documents are not collected by human fact-checkers, they cannot be considered as a gold standard. Thus, it is expected that many of the documents are unrelated to the claim and important information for the validation can be missing.

As discussed above, **FEVER18** is the largest corpus available for the development of automated fact-checking systems. It features a large number of validated claims and annotations of documents and evidence. Nevertheless, since the corpus only covers Wikipedia, the trained systems are unlikely to be able to extract evidence from heterogeneous web-sources. Moreover, the corpus is based on *synthetic* claims derived by modifying sentences from Wikipedia and not *natural* claims that originate from diverse sources on the Internet. Thus, a system trained on this dataset is unlikely to generalize to a real-world fact-checking problem setting, and be able to validate naturally emerging claims (see discussion in Section 7.4.3).

As our analysis shows, while multiple fact-checking corpora are already available, no single existing resource provides full fact-checking sub-task coverage and at the same time is of substantial size and covers multiple domains of text. To eliminate this gap, we created a new corpus which is described in detail in Section 3.2 of this chapter.

3.1.2 Highlighted corpora

In this section, we present the three corpora **FEVER18**, **FNC2017**, **ARC2017** in more detail, as we are going to use them in our experiments in the following chapters of this thesis.

FEVER shared task corpus (FEVER18)

The FEVER corpus (**FEVER18**) was introduced by Thorne et al. (2018a) for the Fact Extraction and Verification (FEVER) shared task.[9] It is the largest corpus available for training automated fact-checking and consists of 185,445 validated claims. The entire corpus is based on about 50k popular Wikipedia documents, and annotators have created the claims by modifying sentences in these documents. The annotators have also marked sentences that either support or refute the claim in the Wikipedia documents as evidence. The evidence sentences for one claim can thereby originate from the same document or from a number of different documents. The claims are labeled as *supported*, *refuted*, and *not enough info*. Even though the stance of the evidence is not annotated, it can be deduced from the verdict. If the claim is *true* the annotated evidences are necessarily *supporting* the claim, and if the claim is *false* the evidences are necessarily *refuting* the claim. In case of *not enough info*, no evidence could be found to refute or support the claim. One of the more elaborated instances from the corpus is given in the Example 3.1 below. Here, two sentences need to be combined in order to validate the claim. Such instances represent 31.75% percent of the dataset. For the remaining instances, only one sentence is given that directly supports or contradicts the claim. Thorne et al. (2018a) report

[9] `http://fever.ai/2018/task.html`

Claim: The Rodney King riots took place in the most populous county in the USA.

Evidence sentences:
[wiki/Los Angeles Riots] The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.
[wiki/Los Angeles County] Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.

Verdict: Supported

Example 3.1: An example for a validated claim from the **FEVER18** corpus

an inter-annotator agreement of 0.6841 Fleiss κ Fleiss (1971) for claim classification and 95.42% precision and 72.36% recall for the annotation of evidence sentences.

The corpus provides annotations for three tasks which had to be tackled during the shared task: document retrieval, sentence selection (evidence extraction), and recognizing textual entailment (which is similar to our definition of claim validation). The main statistics of the corpus is given in Table 3.2. Each instance is represented by a claim with its verdict and Wikipedia documents with annotated evidence sentences. As can be noticed, the training-set of the corpus is biased towards *supported* claims as these instances represent more than half of the instances. Nevertheless, the development set and the held out test-set are balanced. Even though there are 185,445 validated claims in the original corpus, evidence has only been annotated in 14,533 documents. This means that many of the claims are about the same topic. Since other documents have not been considered, there are more evidence sentences for a claim than have been labeled by annotators. In fact, Wikipedia articles often overlap in their content (see for instance the articles about Richard Nixon and the Watergate scandal). The shared task organizers were aware of this issue, and therefore, the evidence sentences predicted by the systems during the shared task were in part evaluated by humans. If the annotators agreed that the new evidence sentences predicted for a claim are valid, these sentences were added to the set of ground truth evidence sentences of this claim. In this way, also the number of evidence sentences for the corpus could be increased.

instances	evidence docs.	supported	refuted	not enough info
185,445	14,533	55 %	20 %	25 %

Table 3.2: Corpus statistics and label distribution of the **FEVER18** corpus

Fake News Challenge corpus (FNC2017)

The Fake News Challenge[10] (FNC) was organized by Pomerleau and Rao (2017) in order to foster the development of AI technology to automatically detect fake news. The challenge received much attention in the NLP community, as 50 teams from both academia and industry participated. The goal in the FNC was to determine the stance (or the perspective) of a news article (a document) relative to a given article headline. An article's stance can either *agree* or *disagree* with the headline, *discuss* the same topic, or can be completely *unrelated*.

For training and validation of competing systems, the organizers provided a dataset[11] (**FNC2017**) which is based on the **Emergent16** corpus. The dataset construction framework, which was used by the organizers to create the corpus, is described below. Moreover, in order to assess the performance of the machine learning model with respect to humans on the task, we have determined the human upper bound in this thesis.

Dataset construction. Ferreira and Vlachos (2016) report that the original **Emergent16** corpus contains claims, for each of which news articles are provided that are concerned with the claim. Nevertheless, after analyzing the corpus, we have found that instead of entire news articles, the corpus often contains only a number of sentences from a news article. This is also reflected in the low average token count for the news articles reported in Table 3.3. Thus, even though we are going to refer to these "news articles" as documents, the reader should keep in mind that these can also contain just a hand full of sentences.

Each news article is summarized into a headline and the stance of this headline with respect to the claim is annotated. Nevertheless, the problem setting defined for the FNC is the detection of the stance of an article with respect to a headline, which was not explicitly given in **Emergent16**. Thus, the FNC organizers modified the original corpus to fit this problem setting. Below, we give a detailed description of the modification process and present an example from the resulting corpus (Example 3.2).

The Emergent database, from which the **Emergent16** dataset was constructed, contains 300 clusters of stories (documents) each of which is centered around a single claim. Each document in a cluster talks about the claim, and either agrees with the claim, disagrees with it, or simply discusses the topic without taking a side. As mentioned above, in the original corpus, each document was summarized into a headline. The organizers matched every headline within a cluster with every document within the same cluster. Depending on the stance of the headline and the stance of the matched document, the pair was annotated as *agree*, *disagree*, or *discuss*. In order to generate the *unrelated* class, headlines and bodies from different clusters were matched. The labeling rules are described below in more detail:

- **agree**: If both the document and the headline were annotated as agreeing with the claim, or both were annotated as disagreeing with the claim, the pair was labeled as *agree*.

[10] `http://www.fakenewschallenge.org/`

[11] `https://github.com/FakeNewsChallenge/fnc-1-baseline`

- **disagree**: If the headline was annotated as disagreeing with the central claim and the document agreed with the claim, the pair was labeled as *disagree*. The same label was given for the reverse case, that is, if the headline agreed and the document disagreed with the claim.
- **discuss**: If either the headline or the document or both were annotated as *discuss* relative to the central claim, the pair was labeled as *discuss*.
- **unrelated**: Headlines and documents from different clusters were randomly matched, and since both are concerned with different topics, the generated instances were annotated as *unrelated*.

Moreover, to prevent teams from using any unfair means by deriving the labels for the test set from the publicly available **Emergent16** dataset, the organizers created 266 additional instances. For this propose, they have taken claims from the web and manually collected documents that are related to the claims and annotated their stance.

Table 3.3 shows the size of the resulting corpus and the label distribution. As can be noticed, the dataset is heavily biased towards *unrelated* claims which make up almost three-quarters of the dataset. The *disagree* class has the smallest number of instances which represents only 2.0% of the dataset.

headlines	docs.	tokens	instances	agree	disagree	discuss	unrelated
2,587	2,587	372	75,385	7.4%	2.0%	17.7%	72.8%

Table 3.3: Corpus statistics and label distribution for the FNC dataset (“docs.” refers to article bodies in the FNC corpus; “tokens” refers to the average number tokens in all documents in the corpus; “instances” are labeled document-headline pairs)

Human upper bound. In order to estimate the room for machine learning systems with regard to human performance, we determined the human upper bound on the resulting stance detection task. We asked five human raters to manually label 200 randomly selected document-claim pairs. The raters reached an overall inter-annotator agreement of Fleiss' $\kappa = 0.686$ (Fleiss, 1971), which is substantial and allows drawing tentative conclusions (Artstein and Poesio, 2008). However, when ignoring the *unrelated* class, the inter-annotator agreement dramatically drops to $\kappa = 0.218$. This indicates that differentiating between the three related classes *agree*, *disagree*, and *discuss* is difficult even for humans. We assume this is because in the original dataset **Emergent16**, the stance of the headlines with respect to the claims is annotated, and in the modified version of the dataset for the FNC (**FNC2017**) the relative stance of the documents with respect to the headlines is automatically derived according to the rules described above. We have found that this often leads to ambiguous instances. In particular, in cases in which the headline *discusses* the central claim and the document *agrees* with the claim or vice versa, it was difficult to determine the stance. In these cases, the stance was annotated

Headline: Robert Plant Ripped up $800M Led Zeppelin Reunion Contract

Text snippets with their stance labels extracted from example documents:

"... Led Zeppelin's Robert Plant turned down £500 MILLION to reform supergroup. ..." **Stance label: agree**

"... No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together. ..." **Stance label: disagree**

"... Robert Plant reportedly tore up an $800 million Led Zeppelin reunion deal. ..." **Stance label: discuss**

"... Richard Branson's Virgin Galactic is set to launch SpaceShipTwo today. ..." **Stance label: unrelated**

Example 3.2: An example from the **FNC2017** corpus

as *discuss*, however, the annotators had difficulties to keep these instances apart from the *agree* instances. This is also reflected in the relatively low inter-annotator agreement for the annotation of *related* documents. This implies that the quality of the corpus is not very high and the human performance is relatively low.

Argument Reasoning Comprehension dataset (ARC2017)

The Argument Reasoning Comprehension (ARC) dataset (**ARC2017**) was constructed for the ARC shared task (Habernal et al., 2018b). The shared task covered a number of sub-tasks for which the corpus provides annotations. Since in this thesis, we are interested in the stance detection sub-task of the ARC shared task, we only discuss the part of the dataset which was created for this purpose. To evaluate the performance of the different machine learning models, which were developed for the FNC, we have modified the ARC in such a way that it fits the FNC problem setting (the experiments on both corpora are described in Section 5.3).

Below, we describe how the original ARC dataset was constructed and how we modified the corpus. Moreover, similar to the **FNC2017**, we have also determined the human upper bound in order to estimate the performance of the machine learning methods with respect to humans on this task.

Dataset construction. For the creation of the dataset, Habernal et al. (2018b) manually selected 188 debate topics with popular questions from the user debate section of the New York Times.[12] For each topic, they collected user-posts, which were highly ranked by other users, and created two claims representing two opposing views on the topic. They then asked crowd workers to decide whether a user post supports either of the two opposing claims or does not express a stance at all. The

[12] https://www.nytimes.com/

Example from the original ARC dataset	
Topic	Do same-sex colleges play an important role in education or are they outdated?
User post	Only 40 women's colleges are left in the U.S. And, while there are a variety of opinions on their value, to the women who have attended ... them, they have been ... tremendously valuable. ...
Claims	**1.** Same-sex colleges are outdated **2.** Same-sex colleges are still relevant
Label	Same-sex colleges are still relevant

Generated instance in alignment with the FNC problem setting		
Stance	**Headline**	**Document**
agree	Same-sex colleges are still relevant	Only 40 women's colleges are left in the U.S. ...

Table 3.4: An example from the original ARC dataset and a generated instance which corresponds to instances in the FNC dataset (**FNC2017**)

resulting dataset covers typical controversial topics from the news domain, such as *immigration*, *schooling issues*, or *international affairs*. While the topics are similar to the FNC dataset, there are significant differences between the corpora. A user post is typically a multi-sentence statement representing one viewpoint on the topic. The news articles of FNC, on the other hand, are longer and usually provide a more balanced and detailed perspective on an issue.

To use the ARC data for the FNC stance detection setup, we considered each user post as a document and randomly selected one of the two claims as the headline. In fact, the user-posts typically express an opinion in several sentences and can be therefore considered as short documents. Since the claims express two different views on the discussed topic, and because the user-posts always referred to the topic, we assumed that the user-posts from the same topic are always related to the two opposing claims for this topic. We labeled the claim-user-post pair as *agree* if the workers have selected this claim for the user post and as *disagree* if the workers chose the opposite claim. We annotated the pair as *discuss* if the workers selected neither of the two claims. In order to generate the unrelated instances, we randomly matched the user-posts with the claims, but avoided that a user post is assigned to a claim from the same topic. Table 3.4 shows an example of our revised ARC corpus structure.

Table 3.5 provides the statistics of the resulting corpus. We have modified the resulting dataset in such a way that the class distribution roughly corresponds to the FNC dataset. The corpus is also biased towards unrelated instances. However, the other three classes are more balanced compared to **FNC2017**.

Human upper bound. We have determined the human upper bound for the resulting **ARC17** dataset using the same procedure as applied to the FNC corpus.

headlines	docs.	tokens	instances	agree	disagree	discuss	unrelated
4,448	4,448	99	17,792	8.9%	10.0%	6.1%	75.0%

Table 3.5: Corpus statistics and label distribution for the ARC dataset ("documents" refers to article bodies in the FNC corpus; "tokens" refers to the average number of tokens in all documents in a corpus; "instances" are labeled document-headline pairs)

Five expert annotators annotated 200 samples according to the four classes scheme. The inter-annotator agreement overall is $\kappa = 0.614$ (Fleiss'-κ) which is slightly lower compared to the FNC corpus. However, the agreement for the three related classes *agree*, *disagree*, and *discuss* is higher as the annotators reach $\kappa = 0.383$. On the basis of the annotation of the five expert, we have computed the *most probable* annotation using MACE (Hovy et al., 2013). We considered this annotation as the human prediction for the task reaching an F1 macro score of 0.773. This score is slightly higher compared to the FNC corpus. We also determined the class-wise F1 scores: *unrelated* = 0.954 *agree* = 0.710, *disagree* = 0.857, and *discuss* = 0.571. As can be noted, the most difficult class, in this case, is *discuss.* For this class, we have the smallest number of instances, and the human performance is the lowest. In fact, after analyzing the *discuss* instances, we have found that in many cases it is not obvious that the user-post is related to the claim. These user-posts have therefore often been labeled as being *unrelated.*

3.2 Snopes fact-checking corpus (Snopes19)

In order to address the drawbacks of existing datasets, we introduce a new comprehensive corpus based on the Snopes[13] fact-checking website. We have chosen the Snopes website as the source for our corpus as it provides many important annotations for each validated claim, such as a verdict, evidence in form of text snippets, and links to documents that provide additional information about the claim.

The resulting corpus consists of 6,422 validated claims with rich annotations based on the data collected by Snopes fact-checkers and our crowd-workers. The corpus covers several domains, including discussion blogs, news, and social media that are often found responsible for the creation and distribution of unreliable information. In addition to validated claims, the corpus comprises over 14k documents annotated with evidence on two granularity levels and provides the annotation of the stance of the evidence with respect to the claims. Our data allows training machine learning models for the four steps of the automated fact-checking process: document retrieval, evidence extraction, stance detection, and claim validation. Below, we provide a detailed description of the created corpus.

[13] `http://www.snopes.com/`

3.2.1 Corpus construction

This subsection describes the source data from the Snopes website, followed by a detailed report on our corpus annotation framework.

Source data

Snopes is a large-scale fact-checking platform which employs human fact-checkers to validate claims (rumors) that often emerge on the web and are then distributed through social media (see also Section 1.2.1). A simple *fact-checking instance* from the Snopes website is shown in Figure 3.1. As displayed in the figure, each instance is represented by a set of fields that store information relevant for fact-checking. At the top of the page, the *claim* and the rating (*verdict*) are given. The Snopes fact-checkers additionally provide a document (*resolution*) which backs up the verdict. The important passages in the document, which we call *Evidence Text Snippets* (ETSs), are marked with a yellow bar. As additional validation support, Snopes provides URLs (underlined words in the resolution are hyperlinks) for *original documents* (ODCs) from which the ETSs have been extracted or which provide additional information. Our crawler extracts this information from the Snopes fact-checking websites, that is, from each website we extract: **1. claim**, **2. verdict**, **3. resolution**, **4. ETSs**, and **5. ODCs** with their URLs.[14].

Corpus annotation

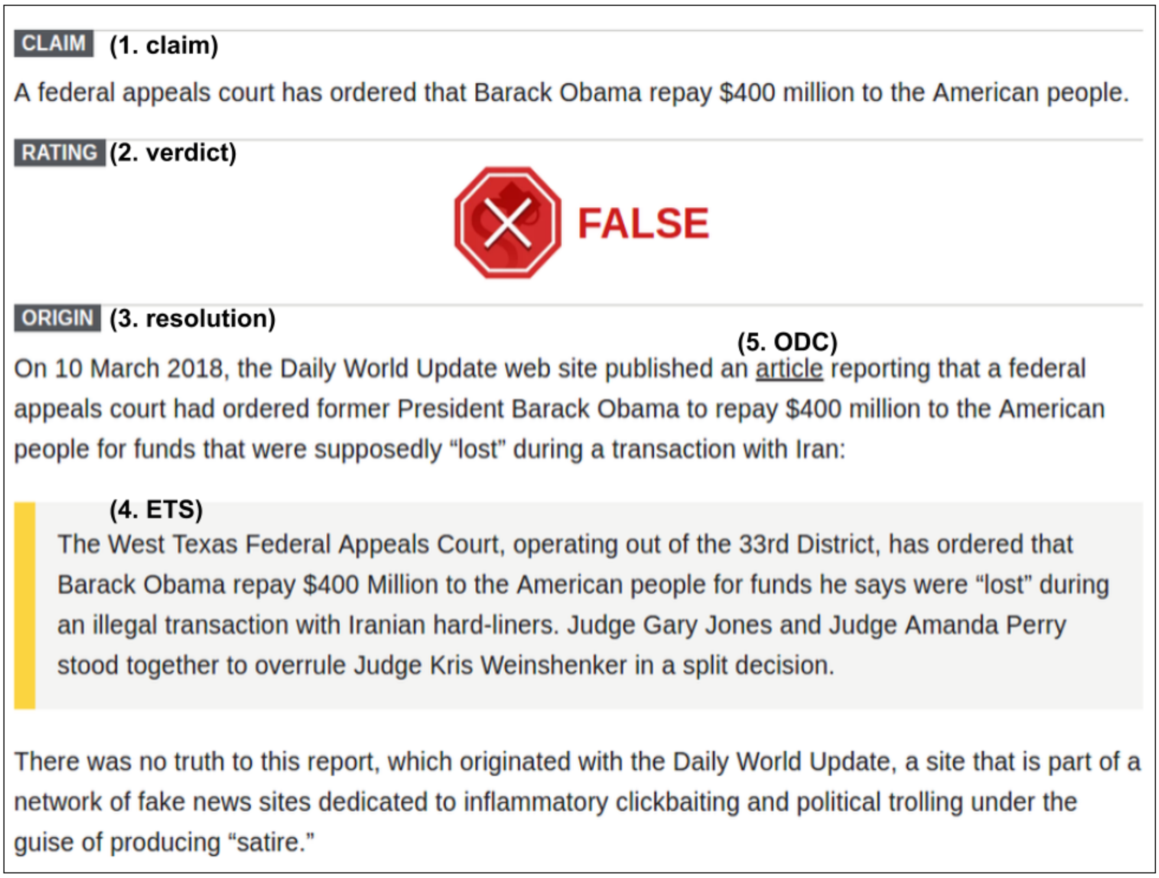
The relationship between the evidence and the claim is an important piece of information in the automated fact-checking process, as it allows the user of the system to gain a better understanding of the fact-checking instance. Moreover, the claim validation component of the fact-checking pipeline would potentially benefit from this information for the prediction of the verdict. However, while ETSs provided by Snopes express a stance towards the claim, this relationship is not explicitly stated on the web-page (see for instance example in Figure 3.1).

Another issue with the original annotation on the Snopes website is the ETS granularity. The ETSs extracted by fact-checkers are relatively coarse and often contain detailed background information. This information is not directly related to the claim, and consequently not useful for its validation. In order to create an informative high-quality collection of evidence that is important for validating the claims, we asked crowd-workers first to label the stance of the ETSs with respect to the claims, and then extract sentence-level evidence from the ETSs that are directly relevant for the validation of the claim. We further refer to these sentences as *Fine Grained Evidence* (FGE). Further details of the annotation process are given below.

Stance annotation. To identify the stance of ETSs towards the claim, we performed an annotation study on Amazon Mechanical Turk[15]. For this purpose, we have developed an annotation interface and defined annotation guidelines (see Appendix A.2.1 for more information).

[14] The crawling of the Snopes website was done in the course of two master theses: Nagaraja (2017) and Zhang (2018)

[15] `https://www.mturk.com/`



CLAIM (1. claim)

A federal appeals court has ordered that Barack Obama repay $400 million to the American people.

RATING (2. verdict)

FALSE

ORIGIN (3. resolution)

(5. ODC)

On 10 March 2018, the Daily World Update web site published an article reporting that a federal appeals court had ordered former President Barack Obama to repay $400 million to the American people for funds that were supposedly "lost" during a transaction with Iran:

(4. ETS)

> The West Texas Federal Appeals Court, operating out of the 33rd District, has ordered that Barack Obama repay $400 Million to the American people for funds he says were "lost" during an illegal transaction with Iranian hard-liners. Judge Gary Jones and Judge Amanda Perry stood together to overrule Judge Kris Weinshenker in a split decision.

There was no truth to this report, which originated with the Daily World Update, a site that is part of a network of fake news sites dedicated to inflammatory clickbaiting and political trolling under the guise of producing "satire."

Figure 3.1: Source fact-checking instance from the Snopes website

The stance annotation is a *three-way classification*. The stance of the ETS (article) towards the claim needs to be labeled as *agree*, *disagree*, and in case the ETS is off-topic or does not explicitly express a stance towards the claim, a third label is given (*No explicit reference to the claim*). For the sake of simplicity, we refer to this label as *no stance* henceforth. Two ETSs annotated with stance are given in the Example 3.1.

FGE annotation. Since the stance is already annotated in the previous step, the complexity of the FGE annotation task can be significantly reduced. We have filtered out ETSs with *no stance*, as they do not contain any supporting or refuting FGE.

The annotation of FGE is a *binary classification* on the sentence level, that is, a sentence is either evidence or not. The crowd workers were asked to annotate sentences in the ETSs if they explicitly referred to the claim and either supported it or refuted it. The annotation guidelines for the annotation of the FGE in the ETSs and the annotation interface is described in the Appendix A.2.2.

Example 3.1 shows two ETSs with marked FGE (displayed in italics). As can be observed, not all information given in the original ETSs is directly relevant for validating the claim. For instance, sentence (1c) in the first ETS only provides additional background information and is therefore not considered as FGE. The second example furthermore demonstrates that FGE is annotated in accordance with the

Claim: The Fox News will be shutting down for routine maintenance on 21 January 2013

Stance: support
Evidence text snippet:

(1a) *Fox News Channel announced today that it would shut down for what it called "routine maintenance".*
(1b) *The shut down is on 21 January 2013.*
(1c) Fox News president Roger Ailes explained the timing of the shutdown: "We wanted to pick a time when nothing would be happening that our viewers want to see."

Claim: Donald Trump supported Emmanuel Macron during the French Elections

Stance: refute
Evidence Text Snippet:

(2a) In their first meeting, the U.S. President told Emmanuel Macron that he had been his favorite in the French presidential election saying "You were my guy".
(2b) *In an interview with the Associated Press, however, Trump said he thinks Le Pen is stronger than Macron on what's been going on in France.*

Example 3.1: Evidence selection for automated fact-checking (FGE selected are marked by italic font)

stance of the ETS: while sentence (2a) does support the claim on its own, it is not annotated as FGE, since the parent ETS stance is refuting.

3.2.2 Corpus analysis

In this subsection, we calculate the inter-annotator agreement on both the stance and FGE annotations, present the corpus statistics, and discuss the characteristics of the corpus. Moreover, we compare the created dataset to the **FEVER18** corpus.

Corpus statistics

Table 3.6 displays the main statistics of the corpus. For a better understanding of the content of the corpus, below, we give an overview over the entire corpus construction procedure.

The corpus contains 6,422 claims, which corresponds to the number of Snopes fact-checking web pages we have crawled. On these web pages, we have found 16,509 ETSs. We further followed the links on the Snopes web pages and found 14,296 web-documents (ODCs) which we also crawled. Workers annotated the stance of the ETSs and FGE in ETSs. In 8,291 ETSs the workers were able to find FGE, which we called *FGE-sets*, i.e., a FGE-set is a set of annotated sentences (FGE) from the same ETS. Many of the ETSs have been decided to have *no stance* (see Table 3.8), and following our annotation study setup, are not used for FGE annotation. Thus, the number of FGE-sets is substantially lower than the number of ETSs. We have found that, on average, an ETS consists of 6.5 sentences. For those ETSs that have support/refute stance, on average, 2.3 sentences are selected as FGE.

	claims	ETSs	FGE-sets	ODCs
count:	6,422	16,509	8,291	14,296

Table 3.6: Overall statistics of the Snopes corpus (FGE-set: a set of annotated sentences (FGE) in the same ETS; ODCs: web-documents that have been linked from the Snopes web pages)

The distribution of the verdicts in our corpus is shown in Table 3.7. As can be observed, the dataset is unbalanced in favor of *false* claims. The label *mostly false* means that some aspects of the claim are false and some are true, but the false aspects outweigh the true aspects. E.g. the claim: *Citrus fruits are an effective medicine against the scurvy disease because of their acidity* is *mostly false* because even though citrus fruits are effective against scurvy, the cause of the cure is vitamin C and not acidity. The label *mostly true* is given if the true aspects outweigh the false aspects. The verdict *other* refers to a collocation of verdicts that do not express a tendency towards declaring the claim as being false or true, such as *mixture, unproven, outdated, legend,* etc.

	false	true	most. false	most. true	other
count:	2,943	659	334	93	2,393
%	45.8	10.3	5.2	1.4	37.3

Table 3.7: Distribution of verdicts for claims on the Snopes website

The stance distribution for ETSs is displayed in Table 3.8. In this case, supporting ETSs and ETSs which do not express any stance are the majority. For supporting and refuting ETSs, annotators identified FGE-sets for 8,291 out of 8,998 ETSs. ETSs with a stance but without FGE sets often miss a clear connection to the claim, so the annotators did not annotate any sentences in these cases.

The class distribution of the FGE-sets is also given in Table 3.8. As for the stance annotation, the supporting FGE-sets are more frequent.

	support	refute	no stance
ETS:			
count	6,734	2,266	7,508
%	40.8	13.7	45.5
FGE sets:			
count	6,178	2,113	–
%	74.5	25.5	–

Table 3.8: Stance distribution for ETSs and the FGE-sets (FGE from the same ETS)

Inter-annotator agreement

Stance annotation. Every ETS has been annotated by at least six crowd workers. The workers were paid 9 cents for the annotation of the stance of three ETSs. We evaluated the inter-annotator agreement between two groups of workers following the approach presented in Habernal et al. (2018b). As reported by Habernal et al. (2018b), the more workers are deployed, the better the inter-annotator agreement between the two groups. In a number of preliminary experiments, we have found that six workers are enough to reach a high inter-annotator agreement for this task. For the evaluation of the inter-annotator agreement, we randomly divided the workers into two groups (of at least 3 workers) and determined the best annotation for each group using MACE (Hovy et al., 2013). The final inter-annotator agreement is determined by comparing the best annotation of the two groups. Using this procedure, we obtain a Cohen's Kappa of $\kappa = 0.7$ (Cohen, 1968), indicating a substantial agreement between the crowd workers (Artstein and Poesio, 2008). The gold annotations for the ETS stances were computed with MACE using the annotations of all crowd workers, that is, we have put the workers from the two groups into one group (of at least six workers) and then computed the best annotation using MACE. We have further assessed the quality of the annotations performed by crowd workers by comparing them to expert annotations. Two experts labeled 200 ETSs, reaching an inter-annotator agreement of $\kappa = 0.7$. We then computed the best expert annotation on the basis of the annotations of the two experts using MACE. The agreement between the best experts' annotations and the computed gold annotations from the crowd workers is $\kappa = 0.683$. This confirms that the gold annotations of ETS stance obtained from crowd workers are of high quality.

FGE Annotation. In this case, workers were asked to annotate the ETSs on the sentence level. Because this task is more difficult than stance annotation, we paid 10 cents for the annotation of the sentences in a single ETS. Similar to the evaluation of the stance annotation, we compared the annotations of FGE in 200 ETSs by experts to the annotations of the same ETSs by crowd workers. The inter-annotator agreement between the crowd workers is 0.55 Cohen's Kappa. The agreement between expert and gold annotations obtained from the crowd workers using MACE is $\kappa = 0.56$. In fact, the task is significantly more difficult than stance annotation. Since sentences may provide only partial evidence for or against the

claim, it is difficult to determine how big the information overlap between a sentence and a claim should be for a sentence to be considered as FGE. The sentence (1a) in the Example 3.1, for instance, only refers to one part of the claim without mentioning the time of the shutdown. We can further modify the example in order to make the problem more obvious:
(a) The Fox News Channel announced today that it is planing a shutdown on 21 January 2013.
(b) The maintenance of the Fox News Channel is on 21 January 2013.
(c) Fox News made an announcement today about a shutdown.

As the examples illustrate, there is a gradual transition between sentences that can be considered as essential for the validation of the claim and those which just provide minor negligible details or unrelated information.

In summary, the agreement scores show that our method for crowd-sourcing the annotation of the stance and FGE produces labels of good quality. For stance detection, we reach a *substantial* inter-annotator agreement with a Cohens' Kappa score of 0.7 (Artstein and Poesio, 2008). For the more challenging task of annotating FGE we reach 0.55 Cohens' Kappa which is considered *moderate* inter-annotator agreement (Artstein and Poesio, 2008).

Data statement

In this subsection, we present the *data statement* (Bender and Friedman, 2018) for our Snopes corpus. According to Bender and Friedman (2018), the data statement should describe a number of specific characteristics of the introduced corpus. Below, we present a subset of these characteristics that are applicable to our Snopes corpus.

A. CURATION RATIONALE: Our corpus is based on the Snopes fact-checking website and we have chosen this website because it provides many annotations required for training machine learning models for fact-checking. The included texts are very diverse and are discussed in the paragraph *F. TEXT CHARACTERISTICS.* in more detail.

B. LANGUAGE VARIETY: Snopes is almost entirely focused on claims made on English speaking websites in the U.S. The corpus therefore only features English fact-checking instances. Since the sources of the included texts are very diverse, no further specifications can be made with respect to the language variety, e.g. whether different English dialects are present in the corpus.

C. SPEAKER DEMOGRAPHICS: As outlined in B., the sources of the texts are very diverse, and we therefore cannot pin down the speaker demographics such as age, gender, race, etc.

D. ANNOTATOR DEMOGRAPHICS: We have not made any specifications on Amazon Mechanical Turk with respect to the demographics of the annotators. Thus, the annotation study was performed by annotators from the pool of all eligible

AMT workers (most workers are female and white, 60% come from the U.S, 30% from India, 10% rest of the world). [16]

F. TEXT CHARACTERISTICS: We have investigated what kind of topics are prevalent in our corpus in order to identify potential biases. We have grouped the fact-checking instances (claims with their ETSs) according to the categories defined by Snopes. We have found that the four categories *Fake News*, *Political News*, *Politics* and *Fauxtography* are most frequent in the corpus ranging from 700 to about 900 claims. A substantial number of claims (200 - 300 claims) are present in the categories *Inboxer Rebellion (Email hoax)*, *Business*, *Medical*, *Entertainment*, and *Crime*.

We have further investigated the sources of the collected documents (Original Documents (ODCs)) and grouped them into a number of classes according to the genre of the document: *real news*, *false and satire news*, *social media*, and *diverse sources* (which encompasses documents from different sources, such as governmental domains, online retail, or entertainment websites). *Real news* is the largest group representing 38% of all the documents. The news articles come from different websites ranging from mainstream news like CNN to tabloid press or partisan news. *False and satire news* is the second-largest group of documents with a share of 30%. The most articles in this group are from the two websites *thelastlineofdefense.org* and *worldnewsdailyreport.com. Social media* documents with a share of 11% come from sources like Facebook and Twitter and represent the third largest class of documents. The rest of the documents representing 21% of the document collection come from *diverse sources.*

Comparison of the Snopes19 and the FEVER18 corpora

We view the **FEVER18** corpus as the most comprehensive dataset introduced so far, and therefore, we compare the properties of this corpus to our Snopes corpus. Although the annotations provided by our dataset are similar to the annotations of the FEVER corpus, important differences exist with respect to the following characteristics: (1) *the domain diversity*, (2) *the nature of claims*, and (3) *the relation between the stance of evidence and the verdict of the claim.* Below, we discuss these characteristics in detail.

(1) *Domain diversity:* One obvious difference is that the Snopes corpus is based on different sources of data and therefore, covers multiple domains (see Section 3.2.2). The **FEVER18** corpus, on the other hand, is only based on Wikipedia.

(2) *Nature of claims:* The claims in the **FEVER18** corpus have been created by modifying Wikipedia sentences and they are therefore *artificial.* Thus, it is not guaranteed that such claims emerge in the real world. Moreover, these claims represent factual statements about knowledge that is available in encyclopedias and knowledge bases and which was often known for years. For instance: *"Tim Roth is an English actor".*, *"The Bermuda Triangle is in the western part of the North Atlantic Ocean."*. Many of the claims contain well-known entities and have a simple structure of *subject* is\was\were *object.*

[16] https://en.wikipedia.org/wiki/Amazon_Mechanical_Turk#Research_validity

The claims in the **Snopes19** corpus, on the other hand, are *naturally* occurring claims that originated on false-news websites, social media, etc. They mostly refer to recent events that often have not been well documented, or to extraordinary events which lack the reference to known named entities e.g.: *"A tornado carried a mobile home for 130 miles and left its occupants unharmed."* The structure of the claims is more diverse often including more than two entities, and the alleged relations between the entities are more complicated. See for instance: *"Louise Rosealma was photographed holding an explosive device made from a glass bottle before she was punched by Nathan Damigo."* The lack of available information in encyclopedias about the claims in the Snopes corpus and their complex structure makes the claim validation problem more difficult compared to the **FEVER18** corpus.

(3) *The relation between the stance of evidence and the verdict of the claim:* Another difference between the two datasets results from the approach based on which the corpora have been constructed. The verdict of a claim for **FEVER18** depends on the stance of the evidence, that is, if the stance of the evidence is *agree*, the claim is necessarily true, and if the stance is *disagree* the claim is necessarily false. This is because the annotators of the FEVER corpus first created the claims and then looked for evidence for the claim in Wikipedia articles. As a result, the FEVER claim validation problem can be reduced to stance detection. Such a transformation is not possible for the **Snopes19** corpus. The evidence in the Snopes corpus might originate from unreliable sources and a claim can therefore have both supporting and refuting ETSs. The stance of ETSs is therefore not necessarily indicative of the veracity of the claim.

In order to investigate the relationship between the stance of the evidence and the verdict for our dataset, we computed their correlation. In the correlation analysis, we investigated how a claim's verdict, represented by the classes *false*, *mostly false*, *other*, *mostly true*, *true*, correlates with the number of supporting ETSs minus the number of refuting ETSs. The formal definition of the problem is as follows: The verdicts of the claims are considered as one variable $V \in \{-2, -1, 0, 1, 2\}$, where the values correspond to the following verdicts $\{-2 : \textit{false}\}$, $\{-1 : \textit{mostly false}\}$, $\{0 : \textit{other}\}$, $\{1 : \textit{mostly true}\}$, $\{2 : \textit{true}\}$, and the stance of the ETSs is considered as the other variable $S = P_{ETS} - N_{ETS}$ where P_{ETS} is the number of *supporting or positive* ETSs for the considered claim and N_{ETS} the number of *refuting or negative* ETSs. In our correlation analysis, we found that the verdict is only *weakly correlated*[17] with the stance, as the computed Pearson correlation coefficient between V and S is only 0.16.

Summary. The presented comparison of the two corpora shows that even though the corpora are similarly structured, the resulting fact-checking problem for the two corpora significantly differs. In contrast to **FEVER18**, the Snopes corpus is a multi-domain corpus, which means that machine learning systems trained on this corpus need to generalize across different kinds of text. The claims in the Snopes corpus are naturally occurring claims, they have a more complicated structure, and refer to recent events which are often not well documented. Thus, the claim validation problem is more complicated for this corpus. Another difficulty for the validation of

[17] `http://www.dmstat1.com/res/TheCorrelationCoefficientDefined.html`

the claims in the Snopes corpus is that the evidences often emerge from unreliable sources. As a result, the verdict of the claims is only weakly correlated with the stance of the evidence. Based on our comparison, we conclude that the fact-checking problem defined by our Snopes corpus is more realistic, as information from real fact-checking instances is used, but also more difficult, as the dataset is more diverse and information from unreliable sources is included. A further analysis of the differences between **FEVER18** and **Snopes19** is given in Section 7.4.2, where experiments on the two corpora are discussed.

3.3 Discussion of the annotation models

The choice of the annotation framework is always subjective and researchers often disagree on how a particular problem should be modeled. We therefore would like to discuss potential weaknesses of the chosen annotation frameworks and justify our particular choice for each sub-task.

Document retrieval. Of the four considered tasks document retrieval is probably the least controversial problem setting. As described in Section 4.1, in our definition of the document retrieval problem, a document is considered to be relevant if it includes important information for the validation of the claim. More concretely, the document is labeled as relevant if it contains one or several evidence sentences supporting or attacking claim. The annotation of the documents is therefore highly dependent on the *evidence extraction* task, where it needs to be defined what constitutes an evidence sentence and whether one should consider sentence-level evidence in the first place. For our corpus **Snopes19**, we have collected the documents that have been annotated by the Snopes fact-checkers with Evidence Text Snippets (ETSs). Thus, no annotation from our side was required.

Stance detection. We consider stance detection on the document level (**FNC2017**), user comment level (**ARC2017**) and ETS level (**Snopes19**). An alternative approach would be to perform stance detection on the sentence/sub-sentence level, as it is typically done in argumentation mining. Here, sentence/sub-sentence arguments are labeled as *supporting* or *attacking* the claim (Stab and Gurevych, 2014). The advantage of the latter approach is that the information is reduced to sentence/sub-sentence level and the stance is potentially less ambiguous. A document, on the other hand, can contain sentences with an opposing stance as the author might highlight both sides of the argument.

However, when considering sentence/sub-sentence level information, contextual information might be missing as the document often only as a whole conveys a consistent stance on a topic, which the author tries to communicate. Moreover, performing document level stance detection also simplifies the subsequent task of labeling evidence sentences, as the annotator only needs to identify evidence sentences having to the same stance as the document. This is cognitively less demanding than identifying evidence without knowing the stance in advance. Thus, the two steps can be efficiently arranged after each other, where in the first step, the stance of the document as a whole is identified, and then in a second step, sentence-level evidence sentences are determined that are most indicative of the stance of the entire document.

Due to the outlined benefits of the second approach, we have chosen the annotation of the stance on the document level/ETS level as it was also suggested for the Fake News Challenge (Pomerleau and Rao, 2017).

Evidence extraction. As described in Section 3.2.1, we define evidence sentences as sentences that explicitly refer to the claim and either support or refute it. An alternative annotation framework is often chosen in argumentation mining where sub-sentence level evidences/arguments are considered (Stab and Gurevych, 2014). In fact, arguments (which in their definition are similar to evidence) in many cases do not follow sentence boundaries. Thus, in argumentation mining, the identification of arguments is often defined as a sequence labeling task where a model annotates spans of tokens (Eger et al., 2017). On the other hand, one could also argue that sentence pieces or even entire sentences are not sufficient to capture comprehensive evidences/arguments and text snippets containing several sentences should be considered as evidence instead. This approach is followed by Snopes fact-checkers, who extract Evidence Text Snippets (ETSs) from documents as evidence. An advantage of this approach is that we have more contextual information and anaphoric pronouns are mostly avoided.

We have not adopted the sub-sentence level approach since annotators often do not agree on the span boundaries. The annotation quality for span annotation is therefore often significantly lower compared to sentence-level annotation (see for instance agreement scores reported in (Zechner, 2002; Tauchmann et al., 2018)).

Text snippet level evidence annotation, as it is practiced by Snopes fact-checkers, is problematic, as the wider context often contains irrelevant and redundant information. In fact, our experiments on the Snopes corpus have shown that learning to identify ETSs in text is almost impossible, as we were not able to train a classifier that significantly outperforms the random baseline. From our perspective, the choice of whether to include a sentence in an ETS was often made arbitrary by the fact-checkers. Thus, we consider the task of annotating ETS as not reproducible. Sentence-level annotation, on the other hand, leads to the best results in our annotation and classification studies. Moreover, we observed that if several sentences in a row are important for the validation of the claim, annotators selected these sentences. Thus, arguments that are composed of several sentences are captured. Due to these advantages, we have adopted the annotation of sentence-level evidence. There have been a number of other studies that also found that annotation of sentence-level evidence works well in practice (see for instance (Thorne et al., 2018a; Stab et al., 2018a)).

Claim validation. In our experiments described in Chapter 7, we consider claim validation as ternary classification. We reduce the different verdicts present on the Snopes website to the three ratings *true, false, and not enough info (i.e. unverifiable given the collected evidence)* following the annotation framework proposed by Thorne et al. (2018a).

Even though widely practiced, the labeling of a given claim with a verdict is disputed among fact-checkers. As described in Section 1.2.1, whereas Snopes and

Politifact label the claim according to a predefined rating scheme, FullFact and FactCheck abstain from giving a verdict and only provide an analysis. The mere classification of a claim as being true, false or not enough info might be too simplistic, miss nuances and conceal the often complex nature of the considered issue. The true state of affairs can be distorted in different ways: (1) a message could have been manipulated unintentionally, (2) different aspects of the message could have been highlighted inappropriately, (3) the message has been presented out of context, or (4) some aspects of the message are wrong whereas others are true. Thus, at least some of the fact-checkers do not reduce the problem of validating a claim to mere classification and provide an analysis instead, where they discuss the issue in detail.

However, such a comprehensive analysis of a claim is beyond the capabilities of current machine learning methods. A system would have to be able to analyze in which way a message has been distorted and discuss these findings in a summary. Since these tasks cannot yet be accomplished automatically, we make use of the more simplistic approach of only classifying the claim. This can act as a proxy for the analysis of the true state of affairs as long as we do not have more sophisticated machine learning methods that would allow us to resolve an issue more comprehensively. As illustrated in Table 3.1, most of the researchers in automated fact-checking today adopt this more simplistic approach since in most corpora, claims are annotated with a verdict.

3.4 Chapter summary

This chapter was concerned with the analysis and the construction of corpora for training automated fact-checking systems. In the first part, we presented related work in the field, that is, we discussed the characteristics of the currently available fact-checking datasets, and investigated their strengths and weaknesses. We have found that while a large number of fact-checking corpora exist, there is not a single multi-domain dataset that is of substantial size and provides annotations for the different tasks in the fact-checking process.

In the next part of the chapter, we highlighted a number of corpora in more detail: the FEVER shared task corpus **FEVER18**, the Fake News Challenge corpus **FNC2017**, and the Argument Reasoning Comprehension Corpus **ARC2017**. The **ARC2017** was created by us through modifying an existing corpus introduced by Habernal and Gurevych (2017). For the two corpora **FNC2017** and **ARC2017**, we have presented the human upper bound, which we have determined in an annotation study. The properties of these corpora are of particular importance, as they are used for the experiments in the following chapters of this thesis.

In order to address the lack of an appropriate corpus for training machine learning models for fact-checking, in the last part of the chapter, we have introduced a new richly annotated corpus based on the Snopes fact-checking website. This corpus allows training systems for document retrieval, stance detection, evidence extraction, and claim validation. We have presented the original data from the Snopes website, which we have crawled, and our annotation methodology for labeling the stance of evidence text snippets and annotating sentence-level evidence. As shown by the evaluation of the resulting corpus, our annotation approach leads to high inter-annotator agreement. Moreover, we provided the statistics of the corpus, analyzed the biases of the dataset, and compared the corpus to the **FEVER18** corpus as the most comprehensive dataset for fact-checking introduced to date. The analysis of our corpus has shown that it is biased towards false claims and many of the provided evidence text snippets originate from unreliable sources, such as false news websites. Since many of the evidences are unreliable, the veracity of a claim for our corpus cannot be easily deduced from the stance of the evidence as for the FEVER corpus. We therefore concluded that the more realistic fact-checking problem setting posed by our Snopes corpus is more challenging than the fact-checking problem defined by the FEVER corpus.

Chapter 4

Document retrieval

The previous three chapters give an introduction to automated fact-checking and present datasets for training automated fact-checking systems. This chapter and the following three chapters are concerned with specific sub-tasks of the automated fact-checking pipeline (Chapter 2). In this chapter, we are addressing *document retrieval* that is the first sub-task of the pipeline. Since in this thesis, we consider document retrieval as a sub-task of a fact-checking system, we define the problem as *the retrieval of documents that contain important information for the validation of a given claim.*

The document retrieval problem is discussed in this chapter in five sections. In the first section, we introduce the document retrieval problem setting and provide an example. In the second section, we present related work, where we in particular focus on document retrieval systems that retrieve documents for evidence extraction. The third section is concerned with different document retrieval approaches that are designed for the FEVER shared task document retrieval problem setting. In the third section, we also introduce a novel entity-linking approach for retrieving Wikipedia articles on the basis of given a claim. In the fourth section, we present experiments with the approaches introduced in the third section. Section five presents an error analysis, where we highlight weaknesses of our entity-linking approach that should be addressed in future work. In section six, we sum up the most important points of this chapter.

The contribution of this chapter is the following.[1]
(4) We propose a novel document retrieval system based on entity linking that outperforms other approaches on the **FEVER18** dataset.

4.1 Problem setting

To evaluate the document retrieval systems presented in this chapter, we use the FEVER shared task corpus **FEVER18** discussed in Section 3.1.2.

In this thesis, we define document retrieval in the context of automated fact-checking as the retrieval of documents from a document collection using a claim as a query. The retrieved documents shall contain relevant information for the vali-

[1] The complete list of contributions ranging from 1 to 10 is given in Section 2.1.1.

dation of the claim. We define this problem setting formally as follows: Find the k most relevant documents $D = \{d_1, ..., d_k\}$ to the query (claim) c in the document collection DC according to the *relevance function* $f(DC, c)$ and the *ranking function* $g(DC, c)$. Whereas the relevance function $f(DC, c)$ retrieves n relevant documents d, the ranking function $g(d, c)$ computes a ranking score for these documents. Thus, the documents can be ranked and the k highest ranked documents selected as the retrieved documents. In practice, the relevance function is often based on an *inverted-index*, meaning that if there is lexical overlap between the query and the document, the document is retrieved. The ranking function is based on a similarity measure between the query and the document, such as Term-Frequency Inverse-Document-Frequency (TF-IDF).

An example of a claim c, for which 3 (k) documents D have been retrieved, is given below (Example 4.1). As can be noticed, all the retrieved documents discuss the claim.

Claim: Israel caused flooding in Gaza by opening river dams

Retrieved documents:

Doc.1 The Gaza Ministry of Interior said in a statement that civil defense services and teams from the Ministry of Public Works had evacuated more than 80 families from both sides of the Gaza Valley (Wadi Gaza) after their homes flooded as water levels reached more than three meters. “Israel opened water dams, without warning, last night, causing serious damage to Gazan villages near the border,” General Al-Saudi told Al Jazeera. ...

Doc.2 Hundreds of Palestinians left homeless after Israel opens river dams and floods houses General Al-Saudi said that the dams were opened without warning. The suffering is compounded by the fact that Israel has maintained a complete siege over Gaza for the last eight years, severely limiting electricity and the availability of fuel for generators. It has also prevented the displaced from rebuilding their homes, as construction materials are largely banned from entering. ...

Doc.3 The Daily Mail published a story on Monday that originally accused Israel of intentionally opening dams in southern Israel in order to flood Gaza. The only problem is, ... there are no dams in southern Israel. Honest Reporting, an NGO that according to its website “monitors the news for bias, inaccuracy, or other breach of journalistic standards in coverage of the Arab-Israeli conflict,” took screen shots of the article before amendments were made. Even more embarrassing ... the Daily Mail’s article attempted to connect the flooding in Gaza with the Israel Electric Company’s decision to cut power to the West Bank cities. ...

Example 4.1: Document retrieval for automated fact-checking

4.2 Related work

Document retrieval is a sub-branch of *information retrieval* (Manning et al., 2010; Baeza-Yates et al., 2011) and is the task of identifying free-text documents given a multi-word or multi-sentence query. Document retrieval systems, also referred to as search engines, can be based in lexical search (Robertson et al., 1995), non-lexical (semantic) search (Guha et al., 2003; Bast et al., 2016) or the mixture of the two. Another distinction can be made by differentiating between machine learning based and non-machine learning-based document retrieval systems. Whereas non-machine learning based systems only extract features from the document and the query in order to measure their similarity for document ranking, machine learning based systems learn a ranking function using examples of query-document pairs.

Document retrieval systems typically use an inverted index in order to keep track which term is contained in which document. Thus, when typing a number of keywords into the *search-box*, all documents containing the keywords can be retrieved. For the ranking of the retrieved documents, a number of different approaches have been proposed. Traditional ranking approaches are based on the *Okapi BM25* algorithm (Robertson et al., 1995) and/or the Google's *Page Rank* algorithm (Page et al., 1999). Okapi BM25 is a lexical, non-machine learning-based algorithm which makes use of Term Frequency - Inverse Document Frequency (TF-IDF) in order to measure the similarity between the query and a document. A document and a query are thereby often represented as TF-IDF weighted bag-of-words vectors and their cosine similarity is taken as the ranking score. Page Rank uses the information about the cross-references between documents in order to determine the documents' importance. The more links point to a document, the more important it is and the higher its ranking position.

In contrast to Okapi BM25, which only considers lexical overlap between the query and the documents, semantic (non-lexical) non-machine learning ranking algorithms are based on explicit semantic representations (Guha et al., 2003; Ruotsalo, 2012), such as open linked data (Bizer et al., 2011), semantic web ontologies (Huiping, 2010), or knowledge graphs (Singhal, 2012)).

Machine learning based semantic ranking approaches make use of learned representations (Kumar et al., 2012; SanJuan et al., 2007), topic models (Eickhoff and Neuss, 2017), and deep learning approaches (Guo et al., 2016; Mitra and Craswell, 2017; McDonald et al., 2018).

Semantic and machine-learning-based ranking approaches are widely used in today's search engines. However, document retrieval systems for practical NLP applications still often only rely on TF-IDF similarity for ranking because the approach is effective and easy to implement.

Stab et al. (2018a) developed *ArgumenText* which is a search engine for retrieving arguments. ArgumenText is based on an index generated by Elasticsearch (Gormley and Tong, 2015) and the Okapi BM25 ranking algorithm. In a first step, ArgumenText retrieves documents that potentially contain arguments for a given query. In the second step, the arguments within the documents are identified. Argument retrieval or argument search is a task defined within the field of argumentation mining (Section 1.2.3) and it is similar to the two steps document retrieval and evidence extraction in the context of automated fact-checking. The retrieved arguments are similar to evidence, as they often represent credible statements, such as a conclusion of a scientific study or an expert opinion (Section 1.2.2), that either support or attack a given text query. However, whereas in document retrieval for automated fact-checking the query is a claim, in argument search the query is a controversial topic. The retrieval problem is therefore significantly different. Since a topic is mostly only represented by a number of words (e.g. nuclear energy), keyword based (or Boolean) search (Frants et al., 1999) is often sufficient to find the relevant documents. In automated fact-checking, ideally, the claim as a whole needs to be understood by the search engine in order to find the relevant documents.

Another line of research relevant to document retrieval for automated fact-checking is open domain question answering. Also for this task, a document retrieval system is deployed in the first step of the pipeline.

Chen et al. (2017a) tackle open-domain factoid question answering using Wikipedia as a knowledge source. They consider text-spans in Wikipedia articles as answers for the factoid questions. To address the defined problem, Chen et al. (2017a) propose the *DrQA system*. The system first retrieves Wikipedia articles containing relevant information to a given question and then detects the answer text-spans in the retrieved articles using a *Recurrent Neural Network* (RNN). DrQA is based on the inverted index followed by TF-IDF ranking. Chen et al. (2017a) further improve the performance of the system using n-gram features that represent information about the local word order.

Since the first component of the DrQA system was developed to retrieve Wikipedia articles, Thorne et al. (2018a) used this component as a baseline for document retrieval in the FEVER shared task. In fact, for both tasks, in the first step, one needs to identify Wikipedia articles that contain relevant information to a given query. Whereas for the FEVER shared task the query is a claim, for question answering the query is a question. As both types of queries are sentences, it is expected that the document retrieval problems are similar.

Document retrieval for Wikipedia is a problem setting that provides additional information compared to a general document retrieval problem, and therefore, the task can also be approached in a different manner. A query often features one or multiple *entities* that form its main content. Wikipedia can be viewed as a knowledge base, where each article describes a particular *entity*, and the entity is represented by the article title. Thus, as originally proposed by Cucerzan (2007), document retrieval can be framed as an *entity linking problem*. More concretely, one can identify entity mentions in the claim and link them to the Wikipedia articles of this entity. The linked Wikipedia articles can then be used as the set of the retrieved documents. Nevertheless, even though the entity linking approach is often superior to classical document retrieval methods (see the analysis in Section 4.4), it is only applicable to encyclopedias where the article title corresponds to an entity which is described in the article body.

Since this chapter is concerned with the document retrieval problem defined in the FEVER shared task, we use the DrQA system as a baseline in our experiments. The performance of the DrQA system is compared to the results of two entity linking based document retrieval approaches.

4.3 Document retrieval systems

In this section, we present a method for linking entity mentions in a query to entities in the Wikidata knowledge base (Vrandečić and Krötzsch, 2014) that was proposed by Sorokin and Gurevych (2018), and an entity linking approach that we have developed for the FEVER shared task document retrieval problem. Since Wikidata is based on Wikipedia, the approach proposed by Sorokin and Gurevych (2018) is also suitable for the FEVER document retrieval problem.

4.3.1 Entity linking for Wikidata

Sorokin and Gurevych (2018) introduced an entity linking approach for question answering based on the Wikidata knowledge base (Vrandečić and Krötzsch, 2014). In the first step of the question-answering task, entities in the input question have to be linked to entities in the knowledge base. A jointly optimized neural architecture is used for the *detection* of entity mentions in the input question and their *disambiguation*. In the detection phase, token n-grams need to be identified, which correspond to the entity mention, and in the disambiguation phase, entities in the knowledge base need to be found to which the detected entity mention refers. As features, Sorokin and Gurevych (2018) use the surrounding context of the candidate n-gram on different levels of granularity. A token-level system component extracts higher-level features from the whole question context, and a character-level system component builds lower-level features for the candidate n-gram. E.g. for the question: *Where was Barack Obama born?* the token n-grams "where, where-was, where-was-barack, ... " and the character n-grams "wh, ere, was, bar, ack, ..." are extracted as features.

As described in Section 4.2, entity linking approaches can be leveraged for retrieving Wikipedia articles, whereby entities in the query are matched with the titles of the Wikipedia articles. Since the approach developed by Sorokin and Gurevych (2018) was designed for the Wikidata knowledge base, which is constructed on the basis of Wikipedia, the method is suitable for document retrieval based on Wikipedia. In fact, claims in the **FEVER18** dataset also feature entities that can be linked to Wikipedia articles. Thus, in the experiments in the subsequent section, we evaluate the performance of the system on the FEVER document retrieval task.

4.3.2 Entity linking based document retrieval for Wikipedia

The system introduced by Sorokin and Gurevych (2018) is designed for linking entity mentions in a question to entities in the Wikidata knowledge base. Even though the system can also be applied for linking entity mentions in a claim to Wikipedia articles, it was not developed for this purpose. In fact, after analyzing the **FEVER18** dataset, we have found that other features, which are not used by Sorokin and Gurevych (2018), are also important for document retrieval for this dataset. We have observed that instead of using n-gram features, we can reach higher performance if we use noun-phrases and the subject of the claim as entity candidates (see discussion below). Thus, we have developed our own entity linking approach that is particularly tailored for retrieving articles from Wikipedia in the context of the FEVER shared task.

The main challenge in entity linking is the processing of ambiguous entity mentions. Since a claim is only a single sentence, which does not provide much context for disambiguation, we base our system on entity linking approaches for short texts (Guo et al., 2013; Sorokin and Gurevych, 2018). These approaches focus on the extraction and modeling of the entity mentions.

In our problem setting, we need to find entities in the claims that match the titles of Wikipedia articles. For instance, given the claim "*Robin Thicke has worked with Pharrell Williams*" the two named entities in the claim can be linked to the Wikipedia article titles "*Robin Thicke*" and "*Pharrell Williams*". The article "*Robin Thicke*" is the Wikipedia article that contains supporting evidence sentences for the given claim, and would be therefore the desired document. Following the typical entity linking pipeline, we develop a document retrieval component that has three main steps, which are described below. The three steps are also illustrated in Figure 4.1.

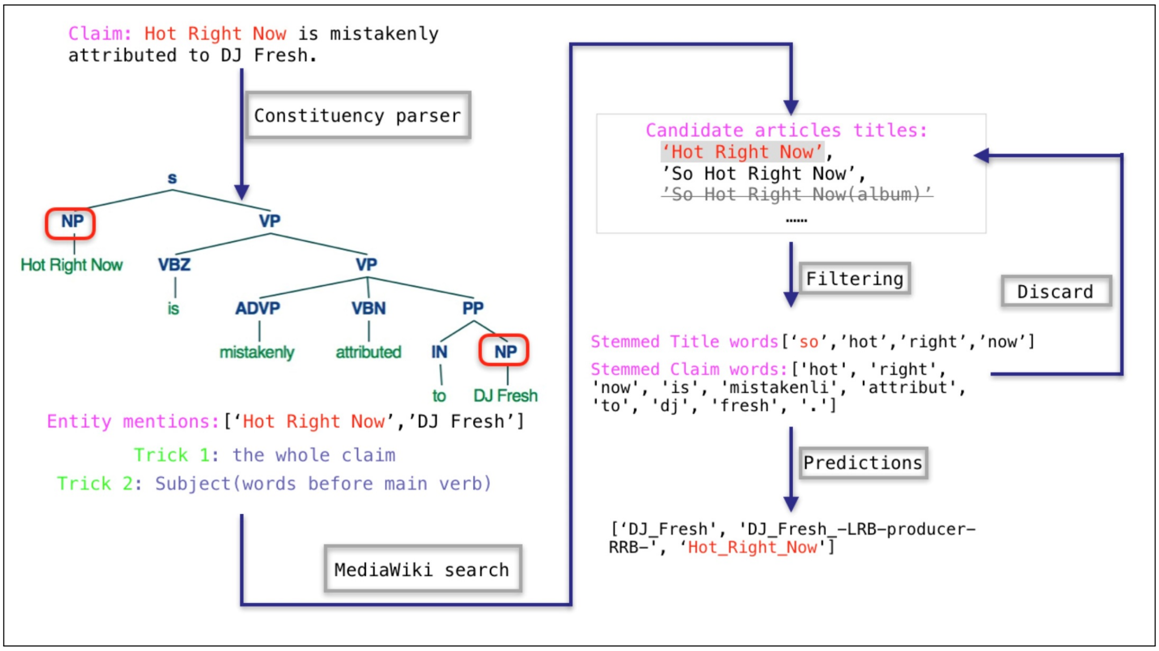

Claim: Hot Right Now is mistakenly attributed to DJ Fresh.
Constituency parser
S
NP
VP
Hot Right Now
VBZ
VP
is
ADVP
VBN
PP
mistakenly
attributed
IN
NP
to
DJ Fresh
Entity mentions:['Hot Right Now','DJ Fresh']
Trick 1: the whole claim
Trick 2: Subject(words before main verb)
MediaWiki search
Candidate articles titles:
'Hot Right Now',
'So Hot Right Now',
~~'So Hot Right Now(album)'~~
......
Filtering
Discard
Stemmed Title words['so','hot','right','now']
Stemmed Claim words:['hot', 'right', 'now', 'is', 'mistakenli', 'attribut', 'to', 'dj', 'fresh', '.']
Predictions
['DJ_Fresh', 'DJ_Fresh_-LRB-producer-RRB-', 'Hot_Right_Now']

Figure 4.1: Entity linking for the FEVER shared task document retrieval problem (source of figure (Zhang, 2018))

Mention extraction. In general, for mention extraction pre-trained *named entity recognition* models, such as those contained in Stanford CoreNLP (Manning et al., 2014) or AllenNLP (Gardner et al., 2018) library, can be used. However, these models focus on the three main types of named entities (Organization, People and Location), which represent only a subset of entities present in the claims of the **FEVER18** dataset. Thus, we develop our own mention extraction approach. In order to find entities of different categories, such as movie or song titles, which are numerous in the claims of **FEVER18**, we employ the constituency parser from AllenNLP (Gardner et al., 2018). The constituency parser breaks a claim into phrases or constituents in the form of a tree structure (see example in Figure 4.1). After parsing the claim, we consider every *noun phrase* as a potential entity mention that can be linked to a Wikipedia article. Nevertheless, a movie or a song title may be an adjective or any other type of syntactic phrase. To account for such cases, we consider the set of words in the claim before the main verb and the whole claim itself also as potential entity mentions. For example, a claim "*Down With Love is a 2003 comedy film.*" contains the noun phrases "*a 2003 comedy film*' and "*Love*". Neither of these two noun phrases is the correct entity mention. But all words before the main verb "*is*" form an entity "*Down With Love*".

Candidate article search. We use the MediaWiki API[2] to search through the entire English Wikipedia in order to find Wikipedia article titles that potentially correspond to the entity mentions found in the claim in the previous step. The MediaWiki API uses the Wikipedia search engine to find matching articles. Given an entity mention as a query, it returns about ten Wikipedia articles. The order of the articles reflects the similarity between the entity mention and the named entity that the Wikipedia article describes. This means that the higher the article is placed in the ranking, the more likely is it that the article describes the entity mentioned in the claim. The MediaWiki API uses the online version of Wikipedia, and therefore, there are some discrepancies between the 2017 dump provided by the shared task organizers and the latest Wikipedia version. To account for the difference, we also perform a search over all Wikipedia articles in the dump, where we consider only exact matches between the extracted entity mentions and the Wikipedia article titles. We add these results to the set of the retrieved articles.

Candidate filtering: The MediaWiki API retrieves articles whose titles overlap with the entity mention query. The results may therefore contain articles with a title longer or shorter than the entity mention used as the query, and thus, these articles can refer to different entities. As illustrated in the example in Figure 4.1, using *Hot Right Now* (song name) as a query, the article with the title *So Hot Right Now* is among the search results. However, this article refers to a different song that is unrelated to *Hot Right Now*. To address this problem, we remove articles with the headlines that are longer than the entity mention and do not overlap with the rest of the claim, e.g. since *So Hot Right Now* is longer than *Hot Right Now* and the word *So* is not contained in the claim, this article is discarded. To evaluate the overlap, we first remove the content in parentheses from the Wikipedia article titles (used for disambiguation) and stem the remaining words in the titles and the claim. Then, we discard a Wikipedia article if its stemmed article title is not completely included in the stemmed claim. We collect all retrieved Wikipedia articles for all identified entity mentions in the claim after filtering and supply them to the next step in the fact-checking pipeline, which is evidence extraction.

[2] `https://www.mediawiki.org/wiki/API:Main_page`

4.4 Experiments

In Table 4.1, we illustrate how the introduced heuristics affect the performance of our document retrieval system on the **FEVER18** development set. Only using noun phrases as queries for the MediaWiki API already leads to high performance, as we are able to retrieve 88.37% of the gold-standard documents when considering the 7 highest-ranked articles (recall @7). By adding entire *claims* and tokens before the main verb (*subject*) to the retrieved entity mentions, the document recall @7 could be further improved by 2%.

candidate entity type	accuracy	recall @7
NPs	92.24	88.37
NPs + claims	92.69	89.40
NPs + claims + subject	**93.55**	**90.33**

Table 4.1: The influence of the selection of different entity mention candidates on the performance of the document retrieval system on the **FEVER18** development set (7 highest ranked articles considered)

Table 4.2 shows the performance of our document retrieval system when retrieving different numbers of the highest-ranked Wikipedia articles. The results show that the more articles we retrieve, the better the accuracy and recall of the system. However, since the overall number of sentences retrieved for a claim is increased, this does not necessarily improve the performance of the subsequent evidence extraction system. In fact, when using 10 articles, we have a larger number of irrelevant candidate evidence sentences and the performance of the evidence extraction component is reduced (see analysis in Section 6.4.1). Thus, in the results reported in the rest of the thesis, we only use the system which retrieves 7 documents.

# docs	accuracy	recall
1	89.75	84.63
3	92.60	88.90
5	93.30	89.94
7	93.55	90.33
10	**93.66**	**90.49**

Table 4.2: Performance of our retrieval system when using different numbers of search results on the **FEVER18** development set

In Table 4.3, we compare the performance of our document retrieval systems to the results obtained by the baseline system, which was provided by the shared task organizers (Thorne et al., 2018a), and the entity linking approach proposed by Sorokin and Gurevych (2018). As the results demonstrate, we were able to substantially outperform both systems. The FEVER shared task baseline is only based on the inverted index and TF-IDF ranking and does not make use of the additional information provided by the Wikipedia article headlines. The entity linking approach proposed by Sorokin and Gurevych (2018) is designed to link entities in questions with entities in the Wikidata knowledge base, which is not well suited for retrieval Wikipedia documents. Compared to questions, the claims in the FEVER shared task corpus exhibit a different structure. We have exploited this structure in our entity linking approach by identifying candidate entity mentions using a constituency parses and filtering the retrieved articles based on heuristics. As demonstrated by the results, the introduced additional modifications allow us to identify the Wikipedia articles with higher recall, and we can therefore substantially increase the performance of the system.

system	recall
Thorne et al. (2018a)	70.20
Sorokin and Gurevych (2018)	73.70
our system	**90.49**

Table 4.3: Results of the compared entity linking approaches on the **FEVER18** development set

4.5 Error analysis

We have identified the following causes for the errors made by our system:

Spelling errors. A word in the claim or in the article title is misspelled. E.g. "*Homer Hickman wrote some historical fiction novels.*" vs. "*Homer Hickam*". If spelling errors occur, our document retrieval system discards the article during the filtering phase. In order to address this problem, a document retrieval system could be based not on exact matches between tokens but on the cosine similarity between word or/and character embeddings. Since a misspelled word would have a similar word or character embedding, the performance of the system for these cases could be improved.

Missing entity mentions. The title of the article that needs to be retrieved, is not related to any entity mention in the claim. E.g. Article title to be retrieved: "*Blue Jasmine*"; Claim: "*Cate Blanchett ignored the offer to act in Cate Blanchett.*". This problem is difficult to solve, as there is no lexical overlap between the title and the claim. However, also in this case, embeddings could potentially help to improve performance, as the embeddings to the entities in the claim and the embedding of the article title might be semantically similar.

Search failures. Some Wikipedia articles refer to different entities but they have the same titles. For the disambiguation, they contain a category name in parentheses, e.g., Michael Jordan (basketball). Since the correct entity among the different alternatives needs to be retrieved, this makes it additionally difficult to find articles using the MediaWiki API. E.g. for the claim "*Alex Jones is apolitical*" the article "*Alex Jones (radio host)*" needs to be retrieved, but the MediaWiki only returns Wikipedia articles that refer to other people: "*Alex Jones (actor)*", "*Alex Jones (baseball)*", "*Alex Jones (basketball)*", ... A future document retrieval system should be able to identify all article headlines that match the entity and then disambiguate between those.

4.6 Chapter summary

In this chapter, we analyzed the problem of retrieving documents in the context of automated fact-checking. Thereby, we mostly focused on the document retrieval sub-task of the FEVER shared task. We first discussed the document retrieval problem setting and provided an example. Next we presented related work for the task of retrieving documents, from which in a subsequent step, evidence for a claim or answers to a question need to be identified. We thereby presented a document retrieval approach for open-domain question answering which served as a baseline in the FEVER shared task.

In the second part of the chapter, we presented two entity linking based approaches for retrieving Wikipedia articles. These methods are based on the observation that entities in the claims of the FEVER shared task often match headlines of Wikipedia articles. The first approach we have presented was proposed by Sorokin and Gurevych (2018) and was originally designed for linking entities in questions to entities in the Wikidata knowledge base. However, having observed that claims in the **FEVER18** dataset have a different structure compared to questions, we have developed our own entity liking approach that is particularly tailored for matching entities in the claim of the **FEVER18** dataset to Wikipedia articles.

After presenting the two entity linking approaches, we discussed the experiments, which we have performed with the systems. We have shown that the different components, which we have introduced in our system, are beneficial and help to increase performance. As a result, we are able to substantially outperform the FEVER shared task baseline system, as well as the entity linking approach proposed by Sorokin and Gurevych (2018).

In the last part of the chapter, we presented the results of the error analysis that we have conducted for our entity linking approach, in order to help to improve the performance of the system in future research. In the analysis we have identified three major causes of errors: Spelling errors in the claim or the Wikipedia articles, missing lexical overlap between the claim and the titles of the correct Wikipedia articles, and failures of the used search engine to retrieve the required Wikipedia articles.

Chapter 5

Stance detection

In this chapter, we analyze the stance detection problem which, after document retrieval, represents the second task in the fact-checking pipeline (Section 1.2.2). We define stance detection as the problem of identifying the relative perspective of a document with respect to a given claim. The detection of the stance of a document provides valuable information for the fact-checker operating the system as the *agreement/disagreement* between the retrieved documents and the claim can be made explicit. Moreover, this information can potentially be helpful for the subsequent tasks in the fact-checking pipeline: *evidence extraction* and *claim validation.*

We investigate the stance detection problem by running experiments on the corpora **FNC2017**, **ARC2017**, and **Snopes19**. Here, the stance of a *news article* (**FNC2017**), a *user-post* (**ARC2017**) or an *Evidence Text Snippets (ETS)* (**Snopes19**) needs to be identified with respect to a *claim* (**ARC2017**, and **Snopes19**) or an *article headline* (**FNC2017**). Even though *news articles*, *user-posts*, and *ETSs* vary in size, they in general consist of multiple sentences, and for the sake of simplicity, we will refer to them as *documents.*

The analysis of the stance detection problem in this chapter is structured as follows. In the first part (Section 5.1), we discuss the stance detection problem and provide an example. The second part (Section 5.2) gives an overview of related work in the field of stance detection. In the third part (Section 5.3), we analyse the Fake News Challenge (FNC) stance detection task[1] (**FNC2017**) and perform experiments on the **ARC2017** corpus for comparison. In the fourth part (Section 5.4), we conduct experiments with three different models on our newly introduced corpus **Snopes19**.

The contributions of this chapter are the following.[2]
(7) We present our stance detection system for determining the stance of a document with respect to a given headline, which was deployed in the Fake News Challenge.
(8) We analyze the top three systems of the Fake News Challenge, and the Fake News Challenge problem setting itself. We evaluate the performance of the features of the top three models on **FNC2017**, as well as the performance of a new set of features.
(1) We evaluate the generalizability of all analyzed FNC models by testing them

[1] `http://www.fakenewschallenge.org/`

[2] The complete list of contributions ranging from 1 to 10 is given in Section 2.1.1.

on a second corpus **ARC2017** and performing cross-domain experiments between **FNC2017** and **ARC2017**.
(4) We conduct experiments on the stance detection task of the **Snopes19** corpus with systems that reach high performance in similar problem settings and discuss the results. Moreover, we conduct an error analysis in order to identify challenging instances that can be addressed in future work.

5.1 Problem setting

We define stance detection as the problem of identifying the relative perspective of a document towards a claim. Our definition of a document is broad, and we consider *news articles*, *user blog posts*, and *Evidence Text Snippets* (ETSs) also as documents. Moreover, in the **FNC2017** corpus, the *anchor*, with respect to which the stance of the documents needs to be determined, is a *news article headline* and not a *claim*. Nevertheless, the headlines are often framed as claims and the FNC problem setting is therefore similar to the stance detection task defined for the other two corpora. This broad definition of the problem setting allows us to analyze how the stance detection problem differs for each corpus and how different methods perform across different corpora.

In this chapter, we consider two stance labeling schemes, the FNC problem setting: *agree, disagree, discuss, unrelated*, and our definition of the stance detection problem for the **Snopes19** corpus: *agree, refute, no stance.*

An example of the annotation of the stance of a number of documents with respect to a claim is given below in Example 5.1. Whereas the green-colored documents *agree* with the claim, the red-colored document *disagrees* with the claim.

Claim: Israel caused flooding in Gaza by opening river dams

Retrieved documents:

Doc. 1 The Gaza Ministry of Interior said in a statement that civil defense services and teams from the Ministry of Public Works had evacuated more than 80 families from both sides of the Gaza Valley (Wadi Gaza) after their homes flooded as water levels reached more than three meters. "Israel opened water dams, without warning, last night, causing serious damage to Gazan villages near the border," General Al-Saudi told Al Jazeera. ...

Doc. 2 Hundreds of Palestinians left homeless after Israel opens river dams and floods houses General Al-Saudi said that the dams were opened without warning. The suffering is compounded by the fact that Israel has maintained a complete siege over Gaza for the last eight years, severely limiting electricity and the availability of fuel for generators. It has also prevented the displaced from rebuilding their homes, as construction materials are largely banned from entering. ...

Doc. 3 The Daily Mail published a story on Monday that originally accused Israel of intentionally opening dams in southern Israel in order to flood Gaza. As writer and Lydia Willgress has learned the hard way, there are no dams in southern Israel. Honest Reporting ... took screen shots of the article before amendments were made. Even more embarrassing ... the Daily Mail's article attempted to connect the flooding in Gaza with the Israel Electric Company's decision to cut power to the West Bank cities. ...

Example 5.1: Stance detection for automated fact-checking

5.2 Related work

Previous works in stance detection mostly focused on *target-specific* stance prediction, where the stance of a piece of text with respect to a topic or a named entity is determined. Corpora concerned with document level stance detection are the three corpora **FNC2017**, **ARC2017**, and **Snopes19**, which we use for the experiments in this chapter, and the datasets introduced by Hasan and Ng (2013) and Faulkner (2014). Below, we first briefly discuss previous work on *target-specific* stance prediction and then work on document level stance detection. Since in this chapter we analyze the FNC problem setting in more detail, we discuss related work on **FNC2017** in a separate subsection.

5.2.1 Target-specific stance detection

Target-specific stance detection has been done for tweets (Mohammad et al., 2016; Augenstein et al., 2016; Zarrella and Marsh, 2016) and online debates (Walker et al., 2012; Somasundaran and Wiebe, 2010; Sridhar et al., 2015). Such target-specific approaches are based on structural (Walker et al., 2012), linguistic and lexical features (Somasundaran and Wiebe, 2010) and use probabilistic soft logic (Sridhar et al., 2015) or neural models (Zarrella and Marsh, 2016; Du et al., 2017) with conditional encoding (Augenstein et al., 2016) to predict the stance. Stance prediction in tweets (Mohammad et al., 2016; Augenstein et al., 2016; Du et al., 2017) and in online debates (Hasan and Ng, 2013) differs from the problem setting we are analyzing in this chapter, as the considered text pieces are represented by only few sentences instead of an entire document. Moreover, the stance is determined with respect to a target, which is a single word or multi-word expression, and not a statement in natural language, such as a claim or a headline, as in our case.

5.2.2 Document-level stance detection

Faulkner (2014) tackled document-level stance detection in student essays, and Hasan and Ng (2013) considered document-level stance detection in online debates. Faulkner (2014) proposed a classifier based on lexica of polarity words, dependency sub-trees, and prompt topic words in the target that invoke a response in the essay. Hasan and Ng (2013) developed a sentence modeling approach to improve document-level stance classification. They hypothesize that sentences with neutral stance should not play a role in determining the document's stance. However, this pipeline approach highly depends on the sentence-level stance classification to be accurate.

5.2.3 FNC stance detection problem

The FNC stance detection task is inspired by Ferreira and Vlachos (2016), who classify the stance of a headline (summarizing a news article in a sentence) towards a specific claim. In the FNC, however, the task is document-level stance detection, which requires the classification of an entire news article (document) relative to the headline (see also the discussion in Section 3.1.2).

There has been much work done on the **FNC2017** dataset and not all of the studies and methods can be discussed here in detail. Thus, below, only the studies are outlined that we consider to be most significant.

The top performing system in the FNC was developed by the team *SOLAT in the SWEN* (Sean et al., 2017) from Talos Intelligence (`Talos` henceforth). They used a combination of a deep convolutional neural network with gradient-boosted decision trees with lexical features. Our system `Athene` (Hanselowski et al., 2017) won the second place with an ensemble of five Multi-Layer Perceptrons (MLPs) and handcrafted features. *UCL Machine Reading* (`UCLMR`) (Riedel et al., 2017) were placed third using a multi-layer perceptron with bag-of-words features. The three systems are presented in Section 5.3.1 in more detail. Other works on FNC use a two-step logistic regression-based classifier (Bourgonje et al., 2017) and a stacked ensemble of five classifiers (Thorne et al., 2017) and achieve the 9th and 11th places respectively.

Although many groups performed experiments on the **FNC2017** corpus, in our analysis in Section 5.3, we focus only on the top three systems due to the availability of source code and our goal of analyzing what contributes most to good performance. More recent studies on the FNC presented below have been published after our analysis, and are therefore beyond the scope of this chapter.

Mohtarami et al. (2018) introduced an end-to-end memory network to address the FNC stance detection task. The memory module of the model is based on convolutional and recurrent neural networks. The model uses a similarity matrix in order to extract text snippets evidence that is most indicative of the stance. The model stands out among other models developed for the FNC task as it allows the user to identify sentences expressing a stance towards the headline that can then be used to reason about the veracity of the headline. However, the model does not outperform the top three systems from the FNC.

Zhang et al. (2018) framed the FNC stance detection task as a ranking problem. For this purpose, they defined a max-margin hinge-loss objective function that maximizes the difference between the ranking score of the correct stance prediction and the ranking scores for the other three classes. This approach is superior to the classification-based method used by the three winning systems of the FNC and reaches a new state-of-the-art on **FNC2017**.

Other studies which are concerned with the FNC are the following (Thorne et al., 2017; Bourgonje et al., 2017; Stanford, 2017).

5.2.4 Summary

As we have outlined in this section, there was not much work on document-level stance detection prior to the FNC. Most of the related work focused on target-specific stance detection where the stance of a relatively short piece of text (one or several sentences), with respect to a multi-word expression (target) is determined. The FNC document-level stance detection problem is more challenging, as the stance of an entire document with respect to a news article headline needs to be determined. A document typically contains many neutral sentences, which do not express a stance, and can include sentences with an opposing stance, that is, sentences agreeing and sentences disagreeing with the headline. In the latter case, it is particularly difficult to determine the stance of the article as a whole. Nevertheless, most of the successful systems on the **FNC2017** corpus do not model the stance of individual sentences, but directly classify the stance of the entire document mostly relying on lexical features. These systems are analyzed in the rest of this chapter in more detail.

5.3 Analysis of the FNC stance detection task

Even though the FNC has received much attention in the NLP community with 50 participating teams, there was no overview or analysis paper on FNC similar to the shared task on detecting stance in twitter (Mohammad et al., 2016; Derczynski et al., 2017; Taulé et al., 2017). To demonstrate the best scientific practices and achieve research transparency, we closed this gap by performing a detailed analysis of the FNC stance detection task and systematically reviewing the top-ranked systems at FNC. More specifically, in this section, we present the top three participating systems of the FNC, reproduce and analyze the results and investigate the performance of different features for the FNC problem setting. Based on the analysis, we propose an optimized set of features and a new model for the FNC task. In order to analyze the generalizability capabilities of the models developed for the FNC, we evaluate their performance on the **ARC2017** dataset.

5.3.1 Participating systems

In our experiments, we consider FNC's three top-ranked systems which we abbreviate as `Talos`, `Athene`, and `UCLMR`.

Talos. Talos Intelligence's SOLAT in the SWEN team (Sean et al., 2017) won the FNC using their weighted average model (`TalosComb`) of a deep convolutional neural network (`TalosCNN`) and a gradient-boosted decision trees model (`TalosTree`). `TalosCNN` illustrated in Figure 5.1 uses pre-trained word2vec embeddings.[3] The embeddings are passed through several convolutional layers followed by three fully-connected and a final softmax layer for classification. `TalosTree` displayed in Figure 5.2 is based on word count, TF-IDF, sentiment, and singular-value decomposition features, and word2vec embeddings. The combined model (`TalosComb`) is illustrated in in Figure 5.3.

Athene. Our model `Athene`[4] (Hanselowski et al., 2017) came in second. We proposed a Multi-Layer Perceptron (MLP) inspired by the work of Davis and Proctor (2017). We extend the original model structure to six hidden and one softmax layer

[3] https://code.google.com/archive/p/word2vec/

[4] The model architecture was mainly developed by Benjamin Schiller, who was a member of our team participating in the Fake News Challenge (Schiller, 2017).

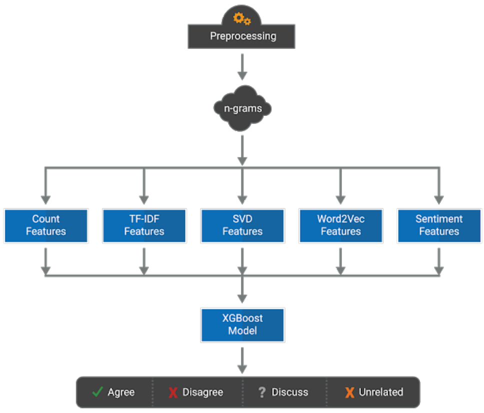


Figure 5.1: `TalosTree` (source: (Sean et al., 2017))

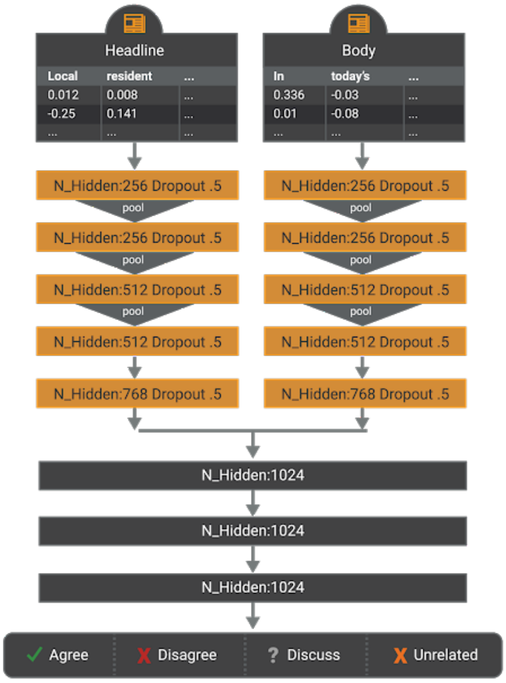


Figure 5.2: `TalosCNN` (source: (Sean et al., 2017))

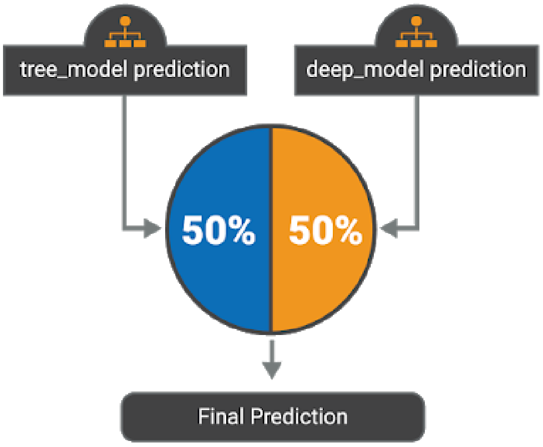


Figure 5.3: Talos Intelligence combined model (`TalosComb`) (source: (Sean et al., 2017))

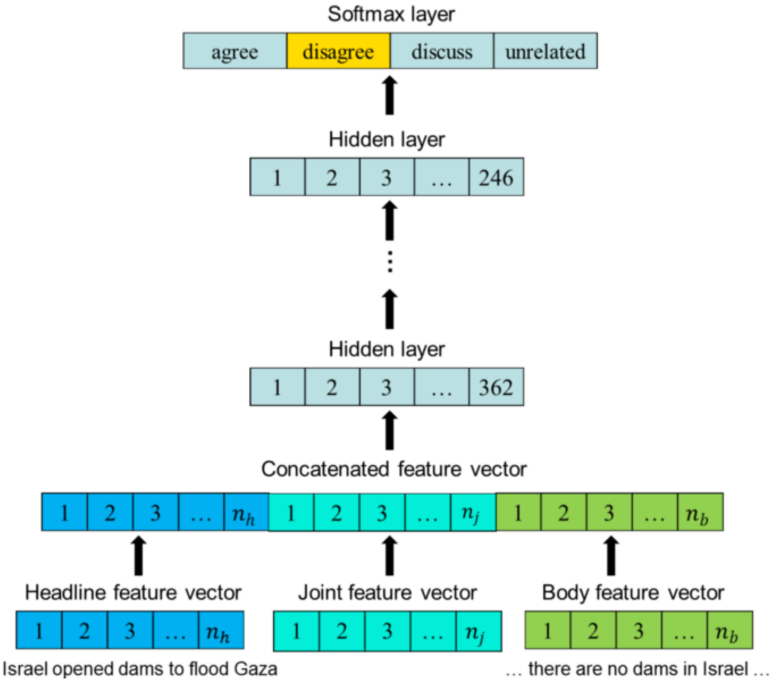


Figure 5.4: Our multi-layer perceptron (`Athene`)

and incorporated a number of hand-engineered features: unigrams; the cosine similarity of word embeddings of nouns and verbs of the headline and the document; topic models based on non-negative matrix factorization (NNMF), latent Dirichlet allocation (LDA), and latent semantic indexing (LSA); and baseline features provided by the FNC organizers. Depending on the feature type, we either formed separate feature vectors for document and headline, or a joint feature vector. In the competition, we used an ensemble of five MLPs that have been trained with different random seeds, and for the prediction of the stance we used hard voting. The model architecture is displayed in Figure 5.4.

UCLMR. The UCL Machine Reading group proposed an MLP model (Riedel et al., 2017) which was ranked third. In contrast to our MLP, they used only one single hidden layer. As features, Riedel et al. (2017) used a term frequency vectors of unigrams of the 5,000 most frequent words for the headlines and the documents. Additionally, they computed the cosine similarity between the TF-IDF vectors of the headline and the document. The feature vectors of the headline and the document are concatenated along with the cosine similarity of the two feature vectors and then fed into the MLP. The model architecture is displayed in Figure 5.5.

5.3.2 Reproduction of the results

Following the instructions from the GitHub repositories of the teams `Talos`[5] and `UCLMR`[6] we could successfully reproduce the results reported in the competition without significant deviations. In Table 5.1, we compare these results to our model `Athene`. Since Talos use a combination of two models, we have also included the re-

[5] https://github.com/Cisco-Talos/fnc-1

[6] https://github.com/uclmr/fakenewschallenge

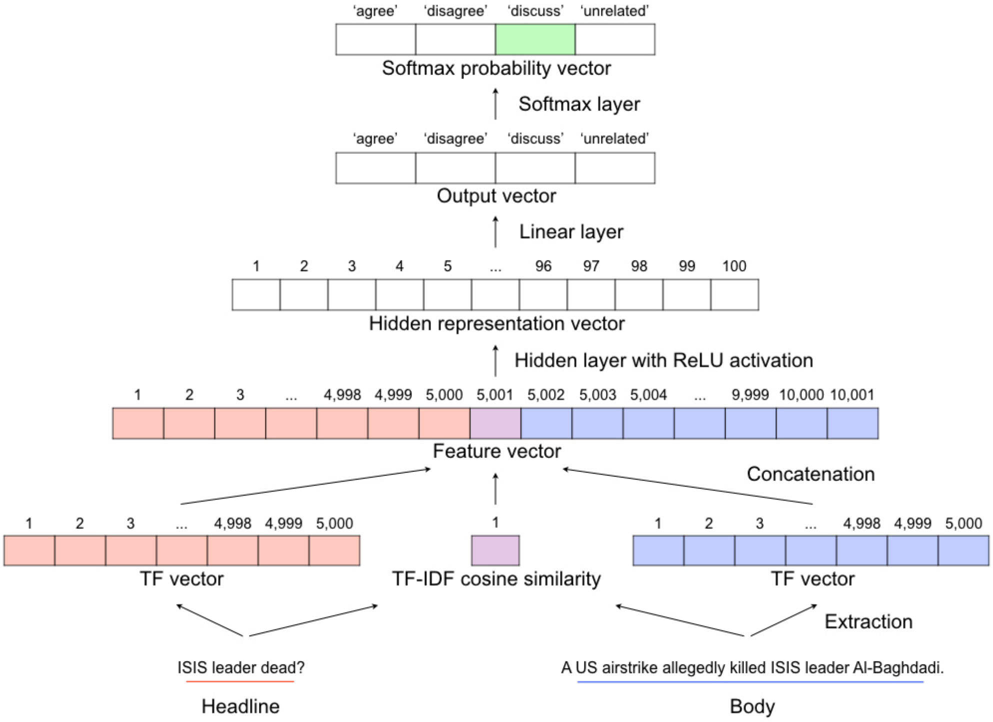


Figure 5.5: Multi-layer perceptron introduced by (Riedel et al., 2017) (UCLMR) (source: (Riedel et al., 2017))

sults of TalosCNN and TalosTree in the table. The models featMLP and stackLSTM illustrated in the table will be introduced later in the chapter.

The first interesting finding is that TalosTree even outperforms the combined model (TalosComb), since TalosCNN's performance is relatively low. We have also observed that basically, all the models perform about 20% worse on the testing set than on the development set. This is most likely because the test dataset is based on 100 new topics, which have not been seen during training (see the discussion in Section 3.1.2). Thus, the evaluation on the test-set can be considered as an out-of-domain prediction. To understand further merits and drawbacks of the systems, we analyze the performance metrics and the features used in the following sub-sections.

5.3.3 Evaluation of the FNC metric

The FNC organizers proposed a hierarchical metric for the evaluation of the participating systems, which is illustrated in Figure 5.6. The metric first awards 0.25 points if a document is correctly classified as *related* (i.e., *agree, disagree, discuss*) or *unrelated* to a given headline. If the document is related, 0.75 additional points are assigned if the model correctly classifies the document as *agree, disagree,* or *discuss.* The goal of this weighting schema is to balance out the large number of unrelated instances in **FNC2017**.

Nevertheless, the metric fails to take into account the highly imbalanced class distribution of the three related classes *agree, disagree,* or *discuss* as illustrated in Table 3.3. Thus, models, which perform well on the majority class and poorly on

Systems	*FNC-metric*	F1m	agree	disagree	discuss	unrelated
Upper bound	.859	.754	.588	.667	.765	.997
stackLSTM	.821	**.609**	.501	**.180**	.757	.995
featMLP	.825	.607	.530	.151	.766	.982
TalosComb	.820	.582	**.539**	.035	.760	.994
TalosTree	**.830**	.570	.520	.003	.762	.994
TalosCNN	.502	.308	.258	.092	0.0	.882
Athene	.820	.604	.487	.151	**.780**	**.996**
UCLMR	.817	.583	.479	.114	.747	.989
Majority vote	.394	.210	0.0	0.0	0.0	.839

Table 5.1: *FNC-metric*, $F_1 macro$ (F1m), and class-wise F_1 scores for the analyzed models

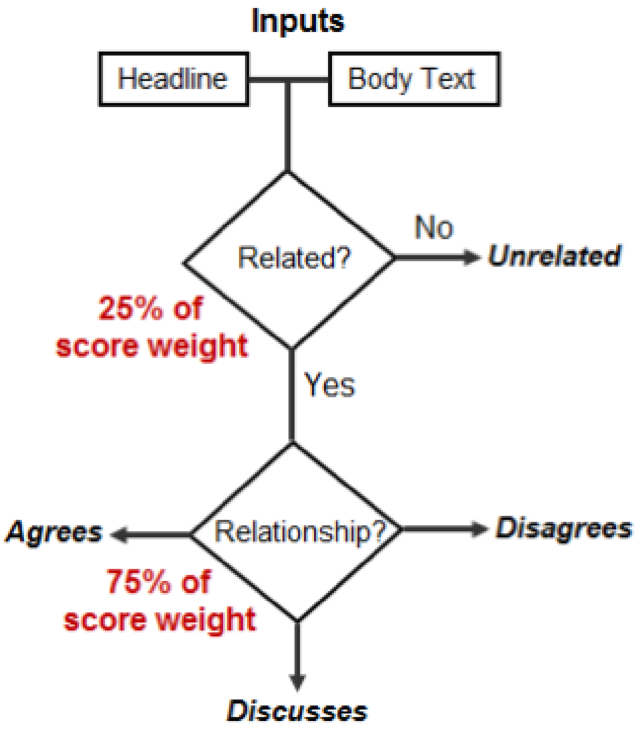


Figure 5.6: (source www.fakenewschallenge.org) FNC metric for the evaluation of systems in the FNC

the minority classes are favored. Because it is not difficult to separate related from unrelated instances (the best systems reach about $F_1 = 0.99$ for the *unrelated* class), a classifier that reaches 100% on related vs. unrelated classification and then just randomly predicts one of the three related classes would already achieve a high *FNC-metric* score. A classifier that always predicts the majority class *discuss* for the related documents even reaches 0.833 on the *FNC-metric*, which is higher than the top-ranked system. Thus, only using this simple approach one would be able to win the FNC.

We therefore argue that the *FNC-metric* is not appropriate for validating the document-level stance detection task defined for **FNC2017**. Instead, we propose the class-wise F_1 and the macro-averaged $F_1 macro$ as new metrics for this task that are not affected by the large size of the majority class. The class-wise F_1 scores are the harmonic means of the precisions and recalls of the four classes, which are then averaged to the $F_1 macro$ metric. The naïve approach of perfectly classifying *unrelated* and always predicting *discuss* for the related classes, would achieve only $F_1 macro = 0.444$, which is clearly a lower score compared to the $F_1 macro$ scores of top three systems (see Table 5.1). By averaging over the individual classes' scores, $F_1 macro$ also allows for a fairer comparison to other datasets, which have a different class distribution than **FNC2017**.

As the results in Table 5.1 show, the three top-ranked systems reach only about $F_1 macro = 0.6$. The results reveal that `TalosCNN` does not predict the *discuss* class yielding an F_1 score of zero for this class. Also, the overall performance of this model is low according to the *FNC-metric*. In contrast to `TalosCNN`, `TalosTree` performs exceptionally well in terms of the *FNC-metric*, but it returns almost no predictions for the *disagree* class. Since there are only a few *disagree* instances and the model predicts more often for the majority class *discuss*, the overall performance of this model appears high. As a result, `TalosTree` would even outperform `TalosComb`.

Considering the FNC results according to our proposed $F_1 macro$ metric, the ranking of the systems changes. The `TalosComb` and `TalosTree` systems are slightly outperformed by `UCLMR` and clearly outperformed by `Athene`. This is because the two Talos models benefit from the *FNC-metric* definition, favoring the prediction of the majority classes *unrelated* and *discuss*. On smaller classes, such as *disagree*, they perform much worse than `Athene` and `UCLMR`. Using $F_1 macro$ as a metric, the `Athene` system would be ranked first, as it outperforms the second-ranked `UCLMR` by 2.1 percentage points. In addition to that, `Athene` also works best on the *agree, disagree,* and *discuss* class.

5.3.4 Analysis of models and features

In this subsection, we first perform an error analysis for the top three models in the FNC in order to find out what the models are learning and in which cases they fail. In order to address the identified drawbacks, we subsequently conduct a systematic feature analysis and derive two alternative models based on our findings.

Error analysis

We identified four major causes of errors: lexical overlap, synonyms, cue words, and indirect mention of disagreement. These causes are discussed below in detail.

Lexical overlap. If there is lexical overlap between the headline and the document, the models classify the instance as one of the related classes, even in cases in which the two are unrelated.
Example case 1. (ground truth: *unrelated*, system predicts: *agree*)

Headline: CNN: Doctor Took Mid-Surgery Selfie with Unconscious Joan Rivers
Document: ... “A TEENAGER woke up during brain surgery to ask doctors how it was going. Iga Jasica, 19, was having an op to remove a tumour at when the anaesthetic wore off and she struck up a conversation with the medics still working on her.” ...

Synonyms. If the document-headline pair is related, but only contains synonyms rather than the same tokens, the model often misclassifies the instance as *unrelated.*
Example case 2. (ground truth: *agree*, system predicts: *unrelated*)
Headline: Three Boobs Are Most Likely Two Boobs and a Lie
Document: ... The woman who claimed she had a third breast has been proved a hoax. ...

Cue words. If keywords like *reports*, *said*, or *allegedly* are detected, the systems often classify the pair as *discuss.*
Example case 3. (ground truth: *disagree*, system predicts: *discuss*)
Headline: Woman pays 20,000 for third breast to make herself LESS attractive to men
Document: ... The woman who reported that she added a third breast was most likely lying. ...

Indirect mention of disagreement. The *disagree* class is especially difficult to determine, as only few lexical indicators (e.g., *false*, *hoax*, *fake*) are available as features. The disagreement is often expressed in complex terms that demands more sophisticated machine learning techniques.
Example case 4. (ground truth: *disagree*, system predicts: *agree*)
Headline: Disgusting! Joan Rivers Doc Gwen Korovin’s Sick Selfie EXPOSED — Last Photo Of Comic Icon, When She Was Under Anesthesia
Document: If the bizarre story about Joan Rivers’ doctor pausing to take a “selfie” in the operating room minutes before the 81-year-old comedienne went into cardiac arrest on August 29 sounded outlandish, that’s because it was.

Our analysis shows that the models exploit the similarity between the headline and the document in terms of lexical overlap. Lexical cue words, such as *reports*, *said*, *false*, *hoax* play an important role in classification. However, the systems fail when semantic relations between words need to be taken into account, complex negation instances are encountered, or the understanding of propositional content in general is required. This is not surprising since the three models are based on n-grams, bag-of-words, topic models and lexicon-based features instead of capturing the semantics of the text.

Feature analysis

Throughout our feature analysis, we use the `Athene` model, which performed best in terms of $F_1 macro$ and allows for a large number of experiments due to its fast computation. All tests are performed on the **FNC2017** training set with 10-fold cross-validation. Below, we first discuss and evaluate the performance of each feature individually and then conduct an ablation test for groups of similar features. Detailed feature descriptions are included in the appendix (Section A.1). Figure 5.7 shows the performance of `Athene` if combined with individual features that are described below.[7]

FNC baseline features. The FNC organizers provided a gradient-boosting baseline using the co-occurrence (COOC) of word and character n-grams in the headline and the document, as well as two lexicon-based features, which count the number of refuting (REFU) and polarity (POLA) words based on small word lists. Figure 5.7 indicates that COOC performs well, whereas both lexicon-based features are on par with the majority vote baseline.

Challenge features. The top three FNC systems rely on combinations of the following features: Bag-of-words (BoW) unigram features, topic model features based on non-negative matrix factorization (NMF-300, NMF-cos) (Lin, 2007), Latent Dirichlet Allocation (LDA-cos) (Blei et al., 2001), Latent Semantic Indexing (LSI-300) (Deerwester et al., 1990), two lexicon-based features using NRC Hashtag Sentiment (NRC-Lex) and Sentiment140 (Sent140) (Mohammad et al., 2013), and word similarity features which measure the cosine similarity of pre-trained word2vec embeddings of nouns and verbs in the headlines and the documents (WSim). All topic models use 300 topics. Besides the concatenated topic vectors (NMF-300, LSI-300), we also consider the cosine similarity between the topics of document and headline (NMF-cos, LDA-cos). The BoW features perform best in terms of $F_1 macro$. While LSI-300, NMF-300, and NMF-cos topic models yield high scores, LDA-cos and WSim fall behind.

Novel features. We also analyze a number of novel features for the FNC task that have not been used in the challenge. The performance of these features is also illustrated in Figure 5.7. Bag-of-character 3-grams (BoC) represent sub-word information and show promising results in our setup. Structural features (STRUC) include the average word lengths of the headline and the document, the number of paragraphs in the document and their average lengths. The low performance of these features indicates that the structure of the headline and the documents is not indicative of the stance. A lexical diversity (LexDiv) feature combining the type-token-ratio (TTR) and the measure of textual lexical diversity (MTLD) (McCarthy, 2005) also shows a much lower score compared to the baseline. Furthermore, we test readability features (READ) which estimate the complexity of a text. Less complex texts could be indicative of deficiently written false news. We tried the following metrics for headline and document as a concatenated feature vector: SMOG grade (Mc Laughlin, 1969), Flesch-Kincaid grade level and Flesch reading ease (Kincaid et al., 1975), Gunning fog index (Štajner et al., 2012), Coleman-Liau index (Mari and Ta Lin, 1975), automated readability index (Senter and Smith, 1967), LIX and RIX (Jonathan, 1983), McAlpine EFLAW Readability Score (McAlpine, 1997), and Strain Index (Solomon, 2006). However, in the present problem setting, these

[7] The feature analysis presented in this sub-section was conducted by Benjamin Schiller (Schiller, 2017).

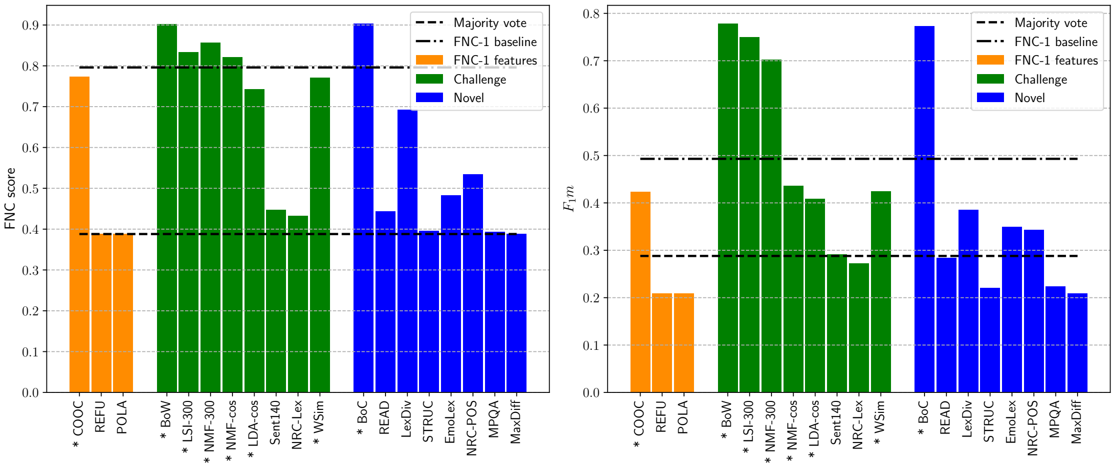


Figure 5.7: Performance of the system based on individual features (*indicates the best performing features)

features show only a low performance. The same is true for the lexical diversity (LexDiv) metrics, type-token ratio, and the measure of textual diversity (MTLD) (McCarthy, 2005). We finally analyze the performance of features based on the following lexicons: MPQA (Wilson et al., 2005), MaxDiff (Kiritchenko et al., 2014), and EmoLex (Mohammad and Turney, 2010). These features are based on the *sentiment*, *polarity*, and *emotion* words included in the headlines and documents, which might be good indicators of an author's opinion. However, our results show that these lexicon-based features are not helpful. Even though the considered lexicons are important for fake-news detection (Shu et al., 2017; Horne and Adali, 2017), for the FNC stance detection task, the properties captured by the lexicon-based features are not very useful.

Feature ablation test. We first remove all features that are more than 10% below the FNC baseline, since they mostly predict the majority class and thus harm the F_1*macro* score. In Figure 5.7, we mark the remaining high-performing features with an asterisk (*). To quantify the contribution of the high-performing features, we conduct an ablation test across three groups of related features: (1) BoW and BoC (Bo*), (2) LSI-topic, NMF-topic, NMF-cos, LDA-cos (Topic), and (3) NRC-POS and WSim (Oth). Table 5.2 shows the results of our ablation test. The BoW and BoC features have the biggest impact on the performance. While the topic models yield further improvements, the NRC-POS and WSim features are not helpful. Hence, we suggest BoW, BoC, and the four topic model-based features as the most promising feature set. We evaluate this feature set on the test-set of **FNC2017**. We call the model equipped with this feature set `featMLP` and display the results in Table 5.1. Although `featMLP` with the revised feature selection outperforms the best performing FNC system `Athene` in terms of F_1*macro* and *FNC-metric* score, the margin is not significant. We suspect this is because many of the features are exploiting a similar pattern and are therefore to some extent redundant.

	Baselines		**Only**			**All without**				
	maj.	FNC	Bo*	Topic	Oth	-Bo*	-Topic	-Oth	All*	All
agr.	0.0	.241	**.772**	.637	0.0	.665	.714	.722	.713	.675
dsg.	0.0	.047	.601	.571	0.0	.530	.598	**.616**	.573	.455
dsc.	0.0	.738	.874	.838	.731	.841	.863	**.87**6	.870	.835
unr.	.835	.970	.991	.983	.964	.982	.989	**.995**	.993	.989
F1m	.209	.499	.796	.757	.425	.754	.791	**.802**	.787	.738

Table 5.2: Results of the feature ablation test. Baseline FNC uses gradient boosting classifier with all FNC baseline features. * states that only the pre-selected features are used (see Figure 5.7). (agr. = agree, dsg. = disagree, dsc. = discuss, unr. = unrelated, maj. = majority vote, FNC = FNC baseline, F1m = $F_1 macro$)

Analysis of models

In order to further push the performance on **FNC2017**, we conducted a number of experiments with different models in various settings.[8] We performed experiments with an ensemble of the three models `featMLP`, `TalosComb`, and `UCLMR` using hard voting. However, we could not significantly improve the results, which suggests that all three models exploit the same patterns in the dataset. Furthermore, since all models struggle with the minority class *disagree*, we have applied different under- and over-sampling techniques to balance the class distribution. Nevertheless, also this technique did not yield improved results.

In the error analysis presented above, we observed that the feature-based models lack semantic understanding. Therefore, we combine a feature-based model with an *LSTM* based model that is better able to capture the semantics of the headlines and the documents using word embeddings and sequential encoding. Sequential processing of information is important in order to get the meaning of the entire sentence, e.g. "*It wasn't long ago that Gary Bettman was ready to expand NHL.*" VS. "*It was long ago that Gary Bettman wasn't ready to expand NHL.*" In Figure 5.8, we introduce the `stackLSTM` model that combines the best feature set found in the ablation test with a stacked LSTM network (Hermans and Schrauwen, 2013). We use 50-dimensional GloVe word embeddings[9] (Pennington et al., 2014) in order to generate sequences of word vectors of a headline–document pair. We combine the token embeddings of the headline and the document with the maximum length of 100 tokens (the remaining tokens of the documents are neglected). These embedded word sequences $v_1, v_2, \ldots, v_n$ are fed through two stacked LSTMs with a hidden state size of 100 with a dropout (Hinton et al., 2012) of 0.2 each. The last hidden state of the second LSTM is concatenated with a vector representing the feature set and is fed into a 3-layer neural network with 600 neurons each. Finally, we add a dense layer with four neurons and a softmax activation function in order to obtain the class probabilities.

[8] The analysis of the models presented in this sub-section was conducted by Benjamin Schiller (Schiller, 2017).

[9] `http://nlp.stanford.edu/data/glove.twitter.27B.zip`

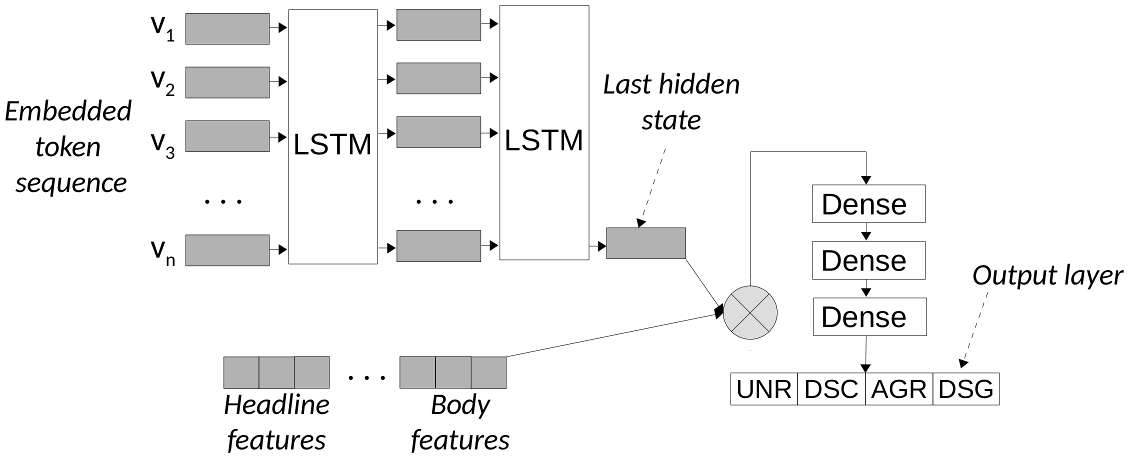


Figure 5.8: Model Architecture of the feature-rich stackLSTM

Headline: NHL expansion ahead? No, says Gary Bettman

Document: It wasn't very long ago that NHL commissioner Gary Bettman was treating talk of expansion as though he was being asked if he'd like an epidemic of Ebola. But recently the nature of the rhetoric has changed so much that the question is becoming not if, but when. ...

Example 5.3: Stance detection for automated fact-checking

Table 5.1 shows the performance of `stackLSTM`. Even though this model outperforms all other methods in terms of $F_1 macro$, the difference to `Athene` and `featMLP` is not significant. Nevertheless, an important advantage of `stackLSTM` is its improved performance for the *disagree* class, which is the most difficult class to predict due to the low number of instances. This means that `stackLSTM` correctly classifies a larger number of complex negation instances. The difference on the *disagree* class between `stackLSTM` and all other methods is statistically significant (using Student's t-test). The model predicts more often for the *disagree* class and gets more of these examples correct without compromising the overall performance. One challenging *disagree* instance, which was correctly classified by the `stackLSTM`, is given in Example 5.3.

5.3.5 Analysis of the generalizability of the models

In order to test the robustness of the models (i.e. how well they generalize to new datasets), we conduct experiments with the models on the **ARC2017** corpus (Section 3.1.2). We first perform in-domain experiments, where we train and test the models on **ARC2017**, and then cross-domain experiments, where we train on **FNC2017** or **ARC2017** and then predict for the other corpus.

In-domain experiments on the ARC2017 corpus: The in-domain results for the **ARC2017** corpus listed in Table 5.3 show that the overall performance of all

Systems	**ARC2017-ARC2017**					
	FNC metric	*F_1macro*	*agree*	*disagree*	*discuss*	*unrelated*
Upper bound	.796	.773	.710	.857	.571	.954
stackLSTM	.685	.524	.451	.518	**.194**	.935
featMLP	.690	.526	.526	.506	.144	.934
TalosComb	**.725**	**.573**	**.593**	**.598**	.160	**.944**
Athene	.680	.548	.516	.482	.190	.933
UCLMR	.667	.519	.517	.503	.121	.932
Majority vote	.430	.214	0.0	0.0	0.0	.857

Table 5.3: *FNC metric*, *F_1macro*, and class-wise F_1 scores for the analyzed models on in-domain experiments on **ARC2017**

models decreases. Because the models have been constructed to perform well on the **FNC2017** dataset, this is not surprising. Nevertheless, for the **ARC2017** corpus, the models are better able to distinguish between *agree* and *disagree* instances, than for the **FNC2017** corpus. We assume this is because the number of *disagree* instances in **ARC2017** is substantially larger than the number of *disagree* instances in **FNC2017**. Moreover, for **ARC2017** the number of *disagree* instances roughly corresponds to the number of *agree* instances (see Table 3.5). The classification of the *discuss* instances, on the other hand, turns out to be more challenging for **ARC2017**. When analyzing the issue, we have found that this is because the user-posts related to the claim often do not explicitly refer to it. Thus, the classifier misses the connection between the user-post and the claim in these cases.

The model `TalosComb` is better able to generalize to the new data than other models as it achieves the best results on **ARC2017**. Even though the `stackLSTM` is again better on the more difficult minority class (in this case *discuss*), the structure and features of `TalosComb` seem to be more appropriate for this problem setting.

Cross-domain experiments: In the cross-domain setting, we train on the training set of one corpus and evaluate on the testing set of the other corpus. Table 5.4 shows the performance of the models when they are trained on the **FNC2017** dataset and then predict for the **ARC2017** dataset. In Table 5.4, the results for the reverse experiments are displayed (training on **ARC2017** and testing on **FNC2017**).

The experiments show that the performance of the models is substantially better than the majority vote baseline. We therefore conclude that the stance detection problems for the two corpora **FNC2017** and **ARC2017** are related and exhibit a common structure. The results also suggest that `TalosComb` is best able to learn from the **ARC2017** corpus, as it is superior in the **ARC2017-ARC2017** and **ARC2017-FNC2017** settings. The `stackLSTM`, on the other hand, yields best results when trained on the **FNC2017** corpus as the **FNC2017-ARC2017** and **FNC2017-FNC2017** settings suggest. We assume that the particular model architecture and the features used for `TalosComb` are more appropriate for the **ARC2017** dataset. In order to further narrow down the cause for the different performance of

Systems	**FNC2017-ARC2017**					
	FNC metric	*F_1macro*	*agree*	*disagree*	*discuss*	*unrelated*
Upper bound	.796	.773	.710	.857	.571	.954
stackLSTM	**.591**	**.401**	.321	.191	.182	.910
featMLP	.586	.389	.321	.159	.171	.906
TalosComb	.584	.365	.336	0.0	.195	**.929**
Athene	.523	.340	**.340**	**.244**	.138	.894
UCLMR	.557	.358	.271	.064	**.201**	.896
Majority vote	.430	.214	0.0	0.0	0.0	.857

Table 5.4: *FNC metric*, *F_1macro* and class-wise F_1 scores based on cross-domain experiments **FNC2017-ARC2017**

Systems	**ARC2017-FNC2017**					
	FNC metric	*F_1macro*	*agree*	*disagree*	*discuss*	*unrelated*
Upper bound	.859	.754	.588	.667	.765	.997
stackLSTM	**.613**	.373	**.343**	.116	.082	.950
featMLP	.585	.351	.322	.111	.033	.939
TalosComb	.607	**.388**	.279	**.183**	**.113**	**.977**
Athene	.548	.321	.277	.097	.028	.882
UCLMR	.482	.288	.234	.109	.080	.728
Majority vote	.394	.210	0.0	0.0	0.0	.839

Table 5.5: *FNC metric*, *F_1macro* and class-wise F_1 scores based on cross-domain experiments **ARC2017-FNC2017**

the models on the two datasets, a deeper analysis is required. This, however, was outside the scope of the current study.

5.3.6 Discussion

Our analysis of the FNC stance detection task shows that the defined problem setting remains challenging. Even though relatively high scores according to the *FCN-metric* have been reported, we are sceptical about the performance of the systems. Because the *FCN-metric* favors systems that perform well on the majority class, we conclude that the metric is not appropriate to validate the performance of the models on **FNC2017**. Our *F_1macro* metric, on the other hand, highlights this problem, as only models that reach good performance across all classes score high. The metric also shows that the best models still reach a relatively low score of 0.6 *F_1macro*. We have found that the low performance is achieved because the models mostly rely on the lexical overlap between the headline and the document. We therefore conclude that in order to reach a higher performance on the stance detection task, more sophisticated machine learning techniques are needed that have a deeper semantic understanding. The cross-domain experiments show that the stance detection problem defined by **FNC2017** and **ARC2017** are similar, because models trained on **FNC2017** and evaluated on **ARC2017** and vice versa outperform the random baseline. Nevertheless, the performance gains are relatively small.

5.4 Experiments on Snopes19

In this section, we analyze the stance detection task defined by our **Snopes19** corpus. In this problem setting, the stance of an ETS towards a claim needs to be identified. According to our annotation framework (Section 3.2.1), an ETS can *support*, *refute* or expresses *no stance* with respect to the claim. We conduct experiments with a number of different models on the corpus. We evaluate the performance of the `featMLP` introduced in the previous section, since it reaches similar performance on the **FNC2017** corpus as the best performing model `stackLSTM`, but is structurally considerably simpler. In order to evaluate how commonly used deep learning models perform on the **Snopes19** dataset, we conduct experiments with two neural architectures: a novel model based on the *universal sentence encoder* (Cer et al., 2018) and *decomposable attention* (Parikh et al., 2016). The experiments are followed by a detailed discussion of the results, and in order to identify outstanding challenges, we conclude the section with a detailed error analysis.

5.4.1 Model architectures

Universal Sentence Encoder with Attention (USEAtt)

We introduce a new model for the stance detection task that is based on the *Universal Sentence Encoder* (USE) (Cer et al., 2018). USE is a neural model pre-trained on a number of text classification tasks, and it reaches good performance across different problem settings. We use the *Deep Average Network* version of the USE (illustrated in Figure 5.9) as a building block in our model. The parameters of the USE can either be kept fixed or fine-tuned during training. In our analysis, we have found that the latter option yields better performance, and therefore, in our experiments reported below, the parameters of the USE are always updated during training. As illustrated in Figure 5.10, we use the USE as one component of our model. A detailed description of the entire model is given below.

Using the USE we compute c, which is the representation of the claim, and s_1, ..., s_i, ..., s_n, which are representations of the sentences of the ETS. In order to compress the sentence representations of the ETS derived by the USE into one vector, we use an attention mechanism, which is inspired by (Yang et al., 2016). The evidence sentences and the claim are first fed through a single layer MLP:

$$\begin{aligned} \hat{s}_i &= \sigma(W\, s_i + b) \quad \forall i \in [1, n], \\ \hat{c} &= \sigma(W\, c + b). \end{aligned} \tag{5.1}$$

In order to compute the attention weights a_i, in the next step, we take the cosine similarity between the claim $\hat{c}$ and each sentence $\hat{s}_i$, and apply the softmax function to ensure that the weights add up to one:

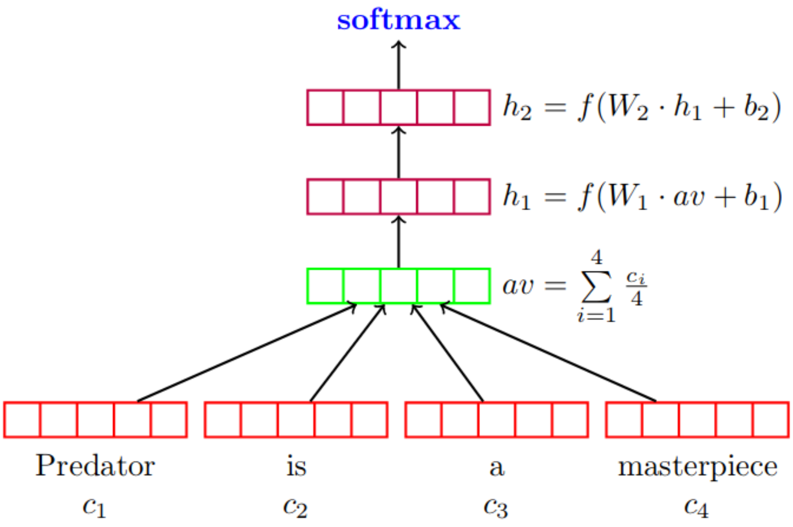

softmax
$h_2 = f(W_2 \cdot h_1 + b_2)$
$h_1 = f(W_1 \cdot av + b_1)$
$av = \sum_{i=1}^{4} \frac{c_i}{4}$
Predator
c_1
is
c_2
a
c_3
masterpiece
c_4

Figure 5.9: Universal sentence encoder deep average network (source (Cer et al., 2018))

$$\begin{aligned} \alpha_i &= \frac{\hat{s}_i \cdot \hat{c}}{|\hat{s}_i| \cdot |\hat{c}|} \quad \forall i \in [1, n], \\ a_i &= \frac{exp(\alpha_i)}{\sum_{i=1}^{n} exp(\alpha_i)} \quad \forall i \in [1, n]. \end{aligned} \tag{5.2}$$

We then apply weighted pooling to reduce the sentence representations of the ETS to a single vector r:

$$r = \frac{\sum_{i=1}^{n} a_i\, s_i}{n}. \tag{5.3}$$

In order to combine the information of the claim c with the aggregated representation of the evidence sentences r, we derive a vector that is based on different combinations of the two vectors:

$$v = [c, r, c - r, c \circ e], \tag{5.4}$$

where $\circ$ represents the Hadamard product. Such a combination has been shown to work better in practice (especially for the SNLI (Bowman et al., 2015) dataset) than a simple vector concatenation. The resulting vector v is fed to a classifier, which is a MLP, in order to predict the stance.

The motivation behind the attention mechanism is that when predicting the stance of the ETS the model should learn to focus on sentences, that are most indicative of the stance of the ETS. In fact, as described in Section 3.2.1, ETSs often contain additional background information that is not directly related to the claim and can mislead the model.

Decomposable Attention (DecAtt)

Decomposable Attention (`DecAtt`) is a model originally proposed by Parikh et al. (2016) for the Stanford Natural Language Inference (SNLI) task (Bowman et al.,

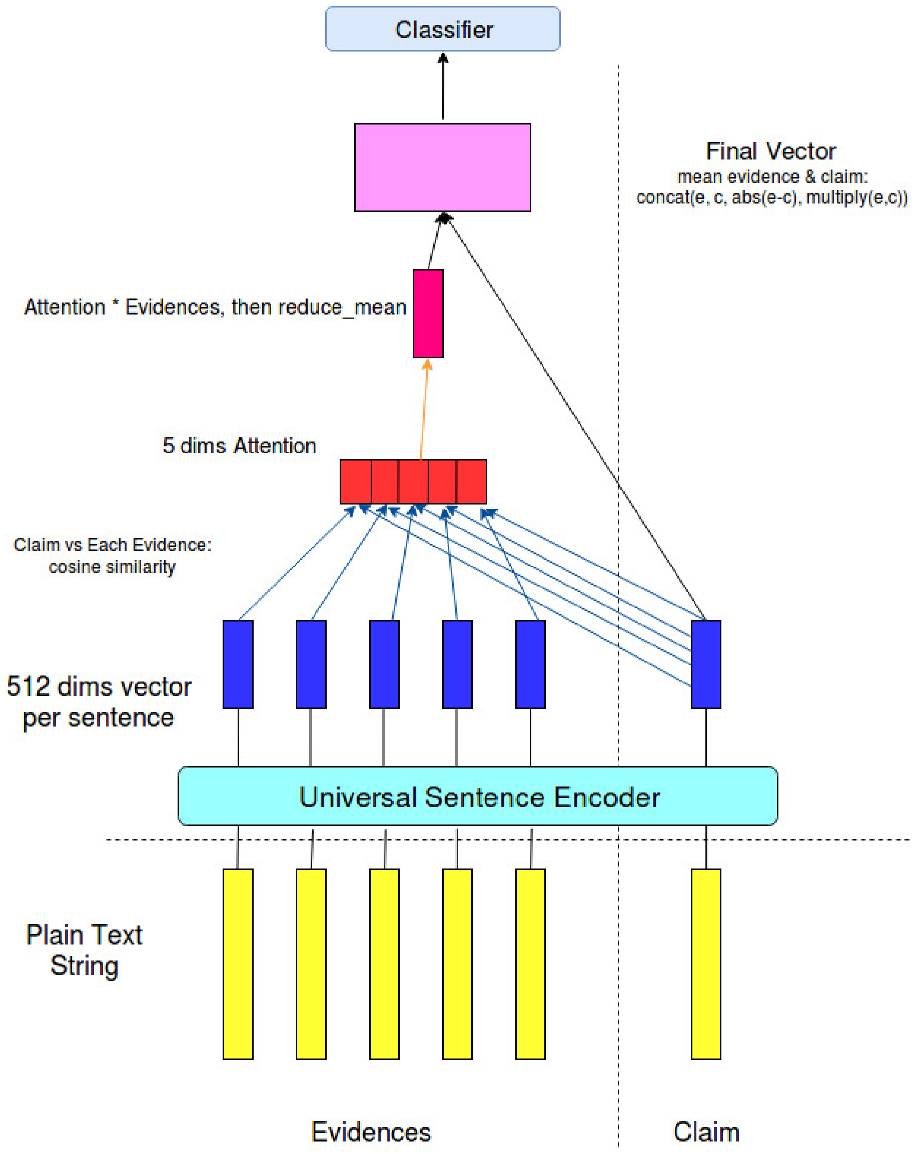


Figure 5.10: Universal Sentence Encoder with Attention (`USEAtt`) (source (Li, 2018))

2015). The model has a low number of parameters and relies on a relatively simple attention mechanism instead of more complex neural network modules like LSTMs or CNNs. However, the model still reached state-of-the-art results on SNLI in 2016. Even though this model was originally designed for the SNLI task, the model can also be used for other tasks such as stance detection.

Standard Siamese sentence encoders (Bowman et al., 2015) applied to the SNLI have the problem that the meaning of the *hypotheses* sentence and the *premise* sentence is encoded into two separate vectors. This represents an information bottleneck as the meaning of the individual tokens of the sentences is *convoluted* in the two resulting representations.

The `DecAtt` model mitigates this issue, by comparing the tokens of the hypotheses and the premise and not encoded sentences. In the process, parts of the hypotheses and premise are first aligned using an attention mechanism and the information about the aligned phrases is then aggregated for the classification of the

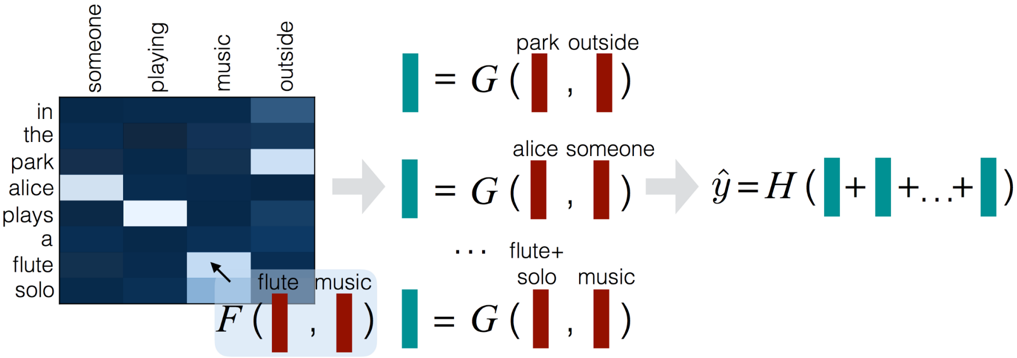


Figure 5.11: Decomposable attention model (source (Parikh et al., 2016))

entailment relation. Parikh et al. (2016) give in the introduction of their paper the following example to describe the intuition behind the approach:

1. Bob is in his room, but because of the thunder and lightning outside, he cannot sleep.

2. Bob is awake.

"*When aligning the second sentence with the first one can easily conclude that it is entailed by the first since* ***Bob*** *can be aligned with* ***Bob*** *and* ***cannot sleep*** *with* ***awake*** *which are basically synonyms*" (Parikh et al. 2016, page 1).

The model solves the entailment problem using the alignment approach in three subsequent steps which are described below. These steps are also illustrated in Figure 5.11.

Attend: In this step, a matrix of attention weights w_{ij} is computed, which represents the attention of each token in the first sentence h_i (hypotheses) to every token p_i in the second sentence (premise). Then, the *sub-phrases*, to which each token attends, are collected:

$$\begin{aligned} \rho_i &= \sum_{j=1}^{n} w_{ij}\, p_i, \\ \eta_j &= \sum_{i=1}^{n} w_{ij}\, h_j. \end{aligned} \tag{5.5}$$

In contrast to the encoding of a sentence using a sentence encoder, ρ_i and η_j are sub-phrase representations that are based on the information of the two sentences. Thus, the interaction of the two sentences is taken into account.

Compare: The representations of the sub-phrases η_j, ρ_i and the token embeddings h_i, p_i are feed through a neural network G in order to compare the two kinds of representations:

$$\begin{aligned} v_{1,i} &= G(a_i, \rho_i), \\ v_{2,j} &= G(p_j, \eta_j). \end{aligned} \tag{5.6}$$

As a result, the entailment relations on the token-sub-phrase level can be determined which are represented by $v_{1,i}, v_{2,j}$.

Aggregate: The vectors $v_{1,i}, v_{2,j}$, which indicate the entailment relation of subphrases, are aggregated, and fed through a neural network H in order to determine the entailment relation between the two entire sentences:

$$\begin{aligned} v_1 &= \sum_{i=1}^{n} v_{1,i}, \\ v_2 &= \sum_{j=1}^{n} v_{2,j}, \\ Y &= H(v_1, v_2), \\ y &= \operatorname{argmax}(Y). \end{aligned} \tag{5.7}$$

Here Y represents a vector of unnormalized scores for each class (entailment relation), and y represents the highest scoring class which is also the predicted class (*entail, contradict,* or *neutral*).

Even though the model was designed for the SNLI task, the problem is structurally similar to the considered stance detection task. For both problems, two different textual inputs are given and it is required to predict relations of the two inputs. The three labels for the two tasks roughly correspond to each other: *entail* ≈ *agree*, *contradict* ≈ *disagree*, *neutral* ≈ *no stance.* However, whereas the premise in the SNLI task is only one sentence, an ETS in the **Snopes19** corpus consists of 6.5 sentences on average. In order to be able to feed an ETS into the model, we simply treat the entire snippet as one long sentence by concatenating all of its sentences. The individual tokens are thereby represented by GloVe (Pennington et al., 2014) word embeddings.

5.4.2 Experiments

The performance of the models discussed in the previous sections on the **Snopes19** corpus is presented in Table 5.6. The comparison shows that `FeatMLP` is superior to the two neural models. This result is similar to the outcome of the FNC, in which feature-based models outperformed neural networks based on word embeddings (Hanselowski et al., 2018a). Moreover, `DecAtt` reaches a substantially higher scores than `USEAtt`. The superior token-based alignment scheme of `DecAtt` compared to the sentence-level-based approach of `USEAtt` is probably the reason for the higher performance. As the comparison of the results to the human upper bound suggests, there is still a substantial room for improvement in future work, as the human upper bound is about 20% higher than the best performing model `FeatMLP`.

model	recall	precision	F1m
upperBound	0.770	0.837	0.802
random baseline	0.333	0.333	0.333
majority vote	0.150	0.333	0.206
FeatMLP	**0.585**	**0.607**	**0.596**
DecAtt	0.510	0.560	0.534
USEAtt	0.380	0.505	0.434

Table 5.6: Stance detection models and baselines (F1m = $F_1 macro$, *upperBound* = *human upper bound*)

model \ gold	support	refute	no stance
support	**472**	**86**	175
refute	41	80	51
no stance	141	74	**531**

Table 5.7: Stance confusion matrix for `FeatMLP`

5.4.3 Error analysis

In order to identify potential future research directions for the improvement of the models, we perform an error analysis for the best-scoring model `FeatMLP`. As described in the previous section, the model is based on bag-of-words representations, character n-grams, and topic models. This indicates that the lexical overlap between the claim and the ETS is an important feature for stance prediction.

The error analysis shows that *supporting* ETS are mostly classified correctly if there is a significant lexical overlap between the claim and the ETS. If the claim and the ETS use different wording, or if the ETS implies the validity of the claim without explicitly referring to it, the model mostly classifies the snippets as having *no stance*. A concrete example of a misclassified instance because of missing lexical overlap is given in Example 5.2. Even though the ETS agrees with the claim, the lexical overlap is relatively low. Most likely for this reason, the model predicts *refute* instead of *agree*.

Moreover, as the distribution of the classes in Table 3.8 shows, *support* and *no stance* instances are more dominant than the *refute* cases. The model is therefore biased towards these two classes and is less likely to predict *refute*. This can also be observed in the confusion matrix illustrated in Table 5.7.

Our analysis of the misclassified *refute* ETSs shows that the contradiction is often expressed indirectly and the model therefore was not able to correctly classify these cases, e.g. "*the myth originated*", "*no effect can be observed*", "*there is no evidence*". Another frequently encountered type of miss-classified *refute* instances is the case, in which the refutation is expressed by pointing out that another statement contradicting the claim is true (implicit refutation henceforth). For instance, the claim "*There is a new punishment for reading the Bible in Saudia Arabia*" is contradicted by the ETS sentence "*Saudi authorities accept the private practice of*

Correct stance: *agree*; **model prediction**: *refute*

Claim: The Reuters news agency has proscribed the use of the word 'terrorists' to describe those who pulled off the September 11 terrorist attacks on America.

ETS: Reuters' approach doesn't sit well with some journalists, who say it amounts to self-censorship. "Journalism should be about telling the truth. And when you don't call this a terrorist attack, you're not telling the truth," says Rich Noyes, director of media analysis at the conservative Media Research Center. They also argue that it's inaccurate. "A news organization's responsibility is to find the facts ... not to play politics with its reporting."

Example 5.2: A misclassified ETS of the Snopes corpus

religions other than Islam" but the contradiction is not explicitly stated.

In addition to the qualitative error analysis described above, we also determine the number of occurrences of different classes of errors. For this purpose, we divide the errors into different classes according to the reason of the misclassification (which we consider to be most likely). The results illustrated in Table 5.8 show that the misclassification of *agree* instances because of the missing lexical overlap is most dominant and represents 24% of all errors. Ambiguous ETSs, which are difficult to classify even for a human, are the second largest group with 22%. False *agree* predictions because of the lexical overlap are also frequently encountered with 20%. Misclassified *refute* instances due to complex refutation or implicit refutation are also numerous with 16% and 10%, respectively.

error type	%
miscl. *agree* because of missing lexical overlap	24
ambiguous ETS	22
false *agree* predictions because of lexical overlap	20
complex refutation	16
implicit refutation	10
other errors	8

Table 5.8: Different types of errors in stance detection

Summary. The presented error analysis is in agreement with the error analysis performed for the **FNC2017** dataset presented in Section 5.3.4. The low performance of deep learning models is most likely because of the relatively low number of samples (16,509 ETSs). The feature-based models, which mostly exploit lexical overlap, perform best on relatively small datasets but have a number of issues that need to be addressed. They are not able to classify the ETSs correctly if the agreement or disagreement is expressed in different terms (for example if synonyms are used). Thus, more elaborate models are required which have a better understanding of the semantics of the text instead only measuring the lexical overlap. In order to further improve performance on **FCN2017** and **Snopes19**, models that are based on self-supervised pre-training or multitask learning may prove superior. These kinds of models are able to reach high performance in new text classification tasks if they are fine-tuned only on a small number of examples (Devlin et al., 2018; Subramanian et al., 2018).

5.5 Chapter summary

This chapter was concerned with the stance detection problem and we focused on two stance detection problem settings. In the first part of the chapter, we conducted a thorough analysis of the FNC stance detection task. Given that the challenge has attracted much attention in the NLP community with 50 participating teams, a detailed analysis is valuable as it provides insights into the problem setting and lessons learned for upcoming competitions. In our analysis, we evaluated the performance of the three top-scoring systems of the FNC, critically assessed the experimental setup, and performed a detailed feature analysis. In the feature analysis, we have found that features based on word n-grams, character n-grams and topic models perform best. We conducted an error analysis for the top three FNC models and found that they mostly rely on lexical overlap for classification. To assess how well the models generalize to a similar problem setting, we run experiments on a second corpus **ARC2017**. We have found that models are to some extent able to generalize across the two corpora, that is, when training on **FNC2017** and predicting for **ARC2017** and vice versa. Since the challenge's metric is highly affected by the imbalanced class distribution of the test data, we also proposed a new evaluation metric based on F_1 scores. Using this evaluation setup, the ranking of the top three systems changes, favoring models that reach good performance across all classes.

In the second part of this chapter, we analyzed the stance detection problem defined by our **Snopes19** dataset. We conducted experiments with three models on the dataset in which a features-based MLP came out on top. As in the FNC stance detection problem, feature-based models showed better performance than pure deep-learning-based approaches. In a subsequent error analysis, we found that also for our corpus **Snopes19**, the feature-based models mostly rely on lexical overlap for the classification.

In our analysis of both problem settings, we found that even the best performing models are not yet able to resolve difficult cases. We therefore concluded that the investigated stance detection problem is challenging, and more sophisticated machine learning techniques are needed. These methods must have a deeper semantic understanding, and are able to determine the stance of a piece of text on the basis of propositional content instead of relying on lexical features.

Chapter 6

Evidence extraction

This chapter is concerned with the evidence extraction problem, which is the next step in the fact-checking pipeline (Section 1.2.2) after stance detection. We define evidence extraction as the problem of finding evidence sentences in the retrieved documents that either support or refute the given claim. The evidence sentences serve as a basis for claim validation which is the next step in the pipeline.

The analysis of the evidence extraction problem in this chapter is structured as follows. We first discuss the evidence extraction problem setting in more detail and give an illustrative example (Section 6.1). We then present related work, where we outline relevant studies on evidence extraction in the area of argumentation mining, interactive evidence detection, and open-domain question answering (Section 6.2). In the next section (Section 6.3), we introduce a number of evidence extraction systems that we have explored in the course of the FEVER shared task. The subsequent section (Section 6.4) discusses the results of the experiments that we have performed with the introduced systems on the FEVER shared task corpus (**FEVER18**) and on the Snopes corpus (**Snopes2019**). The last part of the chapter (Section 6.5) is devoted to the error analysis where we analyze the predictions of the best-performing systems on the two corpora.

The contributions of this chapter are the following.[1]
(9) We propose a new deep learning model, which together with the document retrieval system introduced in Section 4.3.2 reaches the best performance on the FEVER evidence extraction task.
(4) We conduct evidence extraction experiments on the **Snopes19** corpus, with systems that reach high performance in similar problem settings, discuss the results, and perform an error analysis.

6.1 Problem setting

We define evidence extraction as the problem of identifying sentence-level evidence in the retrieved documents. A valid piece of evidence is considered to be a sentence that provides relevant information for the validation of a claim and either supports or refutes the claim. Thus, minor details about the topic of the claim or

[1] The complete list of contributions ranging from 1 to 10 is given in Section 2.1.1.

additional background information is not considered as evidence. Ideally, the evidence originates from a credible source, such as a scientific study or an expert's account. Nevertheless, we also consider statements from the retrieved documents that restate the claim or contradict it as evidence, because we grant the authors of the documents some authority on the issue. In the claim validation process, it could be found that some of the evidence sentences are contradicted by other more substantial evidence, and thus, these evidence sentences are neglected in the claim validation process. However, this is not an issue that we try to resolve in the evidence extraction step, as the goal is only to aggregate all available information that is relevant for the validation of a claim, and not to reason about its validity. More information on this issue is also given in Sections 1.2.2 and 3.2.1.

The evidence extraction problem can either be framed as a *classification task* (Stab et al., 2018b; Stahlhut et al., 2018) or as a *ranking task* (Thorne et al., 2018b; Hua and Wang, 2017; Aharoni et al., 2014; Rinott et al., 2015). In the classification problem setting, evidence extraction is considered as binary classification. We define this task formally as follows. Determine whether the sentence s_{ij} in the document d_i from the retrieved documents D belongs to the set of evidence sentences E_c for the claim c. In other words, evaluate for every $s_{ij} \in d_i \in D$ whether $s_{ij} \in E_c$.

In the evidence ranking setting, all sentences in the retrieved documents are ranked according to their relevance to the claim, i.e. how valuable the sentences for the validation of a given claim are. After ranking, the top-k sentences are then taken as evidence. We define this task formally as follows. Rank all sentences s_{ij} in the documents $\{d_1, ..., d_i, ..., d_n\} = D$ according to the ranking function $f(c, s_{ij})$ and take the top-k sentences s_l as evidence $E_c = \{s_1, ..., s_l, ..., s_k\}$.

An example of extracted evidence sentences for a given claim is presented below (Example 6.1). The evidence sentences are highlighted in the documents by the italic font.

Claim: Israel caused flooding in Gaza by opening river dams

Retrieved documents with highlighted evidence:

Doc.1 The Gaza Ministry of Interior said in a statement that civil defense services and teams from the Ministry of Public Works had evacuated more than 80 families from both sides of the Gaza Valley (Wadi Gaza) after their homes flooded as water levels reached more than three meters. *"Israel opened water dams, without warning, last night, causing serious damage to Gazan villages near the border," General Al-Saudi told Al Jazeera. ...*

Doc.2 *Hundreds of Palestinians left homeless after Israel opens river dams and floods houses General Al-Saudi said that the dams were opened without warning.* The suffering is compounded by the fact that Israel has maintained a complete siege over Gaza for the last eight years, severely limiting electricity and the availability of fuel for generators. It has also prevented the displaced from rebuilding their homes, as construction materials are largely banned from entering. ...

Doc.3 *The Daily Mail published a story on Monday that originally accused Israel* *of intentionally opening dams in southern Israel in order to flood Gaza. The only problem is, … there are no dams in southern Israel. Honest Reporting … took screen shots of the article before amendments were made.* Even more embarrassing … the Daily Mail's article attempted to connect the flooding in Gaza with the Israel Electric Company's decision to cut power to the West Bank cities. …

Example 6.1: Evidence extraction for automated fact-checking (identified evidence sentences are highlighted with the italic font)

6.2 Related work

There is a considerable amount of work related to our definition of evidence extraction in the areas of argumentation miningand open-domain question answering. Below, we present these studies in three different sub-sections.

6.2.1 Argumentation mining

Arguments (or premises) are often indistinguishable from our definition of evidence (see for instance the definition by Rinott et al. (2015)). Thus, work on *argument detection* (Hua and Wang, 2017; Stab et al., 2018b; Levy et al., 2014) is closely related or our definition of the evidence extraction problem. Nevertheless, whereas in argument detection the query is a controversial topic or a controversial claim, for evidence extraction in the area of automated fact-checking, the query is a factual claim (Section 1.2.3).

Aharoni et al. (2014) introduced a benchmark dataset for automated detection of *claims* and *evidence* in the context of *controversial topics*. They give the following definition for the three types of argument elements: Topic: "*a short phrase that frames the discussion.*" Claim: "*a general, concise statement that directly supports or contests the topic.*" Context-dependent evidence: "*a text segment that directly supports a claim in the context of the topic.*" (Aharoni et al. 2014 p. 64-65). An example for the three argument elements is given in (Rinott et al. 2015 p. 440): Topic: "*use of Performance Enhancing Drugs (PEDs)*"; Claim: "*PEDs are bad for health*"; context-dependent evidence: "*a 2006 study shows that PEDs have psychiatric side effects*". The corpus created by Aharoni et al. (2014) consists of 2,683 argument elements collected from Wikipedia in the context of 33 controversial topics. It must be noted that in contrast to our definition of the evidence extraction problem, the framework suggested by Aharoni et al. (2014), and consequently the corpus, only features supporting and no refuting evidence.

Levy et al. (2014) presented a supervised learning approach for detecting *context-dependent claims* that either support or refute a given controversial topic. For the supervision, they used the dataset introduced by Aharoni et al. (2014). The presented approach is designed as a cascade of three classifiers that are based on handcrafted features. The system receives as an input the topic along with relevant articles and outputs context-dependent claims contained therein. The purpose of the classifier cascade is to gradually focus on smaller text segments, while filtering out irrelevant text. Levy et al. (2014) defined the task as a ranking problem, whereby they retrieve context-dependent claims on a sub-sentence level. As the claims support or refute a given topic, the problem is similar to the evidence extraction in automated fact-checking. In fact, an evidence sentence is also often formulated as a claim, e.g. an expert opinion.

Rinott et al. (2015) introduced another dataset for *context-dependent evidence detection* in documents which is similar to the corpus constructed by Aharoni et al. (2014) and contains parts of it. The claims of the corpus (Aharoni et al., 2014) are contain in the corpus (Rinott et al., 2015), but the evidence differs. The dataset introduced by Rinott et al. (2015) is based on 274 articles, it covers 39 topics, it contains 1,734 claims and 3,057 context-dependent evidence. Rinott et al. (2015) additionally provided annotations for three different evidence types that they define as follows: "***Study result*** *of a quantitative analysis of data, given as numbers, or as conclusions.* ***Expert Testimony*** *by a person / group / committee / organization with some known expertise / authority on the topic.* ***Anecdotal evidence:*** *A description of an episode(s), centered on individual(s) or clearly located in place and/or in time.*" (Rinott et al. 2015 p. 440-441). Rinott et al. (2015) developed a classifier that is able to identify sentence-level evidence in the articles and distinguish between the three types of evidence. The classifier is based on the following features: lexicons, named entities, regular expressions and subjects of the evidence sentences.

Lippi and Torroni (2016c) introduced MARGOT, which is an online argumentation mining (argument search) system. They developed a model for evidence extraction and a model for the detection of boundaries for claims and evidence. The goal of Lippi and Torroni (2016c) was to construct a generally applicable system for *context-independent claim and evidence detection*, by using cross-topic applicable features instead of contextualized features. For instance, they consider the structure of parse trees of candidate sentences as containing valuable information for the detection of arguments. MARGOT is trained on the two corpora introduced by Aharoni et al. (2014); Rinott et al. (2015), but in contrast to machine learning approaches previously applied on these corpora (Levy et al., 2014; Rinott et al., 2015), topic-specific features are neglected.

Stab et al. (2018a) developed *ArgumenText*, which is an argument search engine for retrieving arguments from heterogeneous sources. The system consists of thee steps, where first documents for a topic are collected, then arguments in the documents are identified, and in the last step the stance of the arguments with respect to the topic is determined. The annotation framework (argument model) used for ArgumentText is significantly simpler compared to the framework proposed by Levy et al. (2014) that was used for MARGOT. Stab et al. (2018a) only consider *topics* and *arguments*, for which they give the following definitions: "*We define an* ***argument*** *as a span of text expressing evidence or reasoning that can be used to either support or oppose a given topic. ...* ", "*A* ***topic****, in turn, is some matter of controversy for which there is an obvious polarity to the possible outcomes ...*" (Stab et al. 2018b p. 3665). They argue that the advantage of this approach is that "*it allows annotators to classify text spans without having to read large amounts of text and without having to consider relations to other topics or arguments.*(Stab et al. 2018b p. 3666)" This allows them to apply the annotation framework to diverse types of documents, and they use crowd workers to annotate arguments on the sentence level in the documents.

For the automated identification of arguments using machine learning techniques, Stab et al. (2018b) framed the problem as a binary classification, that is, a sentence in the retrieved documents is either an evidence or not (see also the definition in Section 6.1). They proposed two models for the task: a Bidirectional Long Short-Term Memory Network (BiLSTM) and a BiLSTM model that additionally uses features from the topic. The presented problem setting is closely related to our evidence extraction problem, as our goal is also to identify sentence-level text units that either support or oppose a given statement.

6.2.2 Open-domain question answering

The identification of candidate answers in documents for open-domain question answering is a problem setting which is also similar to the identification of evidence in large collections of text. Thus, similar approaches can be used for both tasks.

Wang et al. (2017) tackled open-domain question answering by aggregating evidence (candidate answers) from multiple text passages. Their motivation is that "*the correct answer is often suggested by more passages repeatedly*" and that "*sometimes the question covers multiple answer aspects, which spreads over multiple passages.*" (Wang et al. 2017 p. 1). For the solution of the problem, they proposed the following approach. In the first step, they use an information retrieval system to extract passages from documents that potentially contain the answer. They then generated a preliminary ranking by collecting top-k candidate answers (which they consider as evidence), based on the probabilities computed by standard question answering systems. For the final ranking, they propose two models: *strength-based re-ranker* and *coverage-based re-ranker*, which are based on their motivation. This problem setting is closely related to the evidence ranking problem setting described in Section 6.1.

The retrieval component of the *DrQA system* presented in Section 4.2 was developed by Chen et al. (2017a) to retrieve and rank documents potentially containing answers for open-domain questions. However, the system can be also be used to rank sentences according to their relevance to an arbitrary query. Thorne et al. (2018a) have therefore used the system as a baseline in the FEVER shared task in order to extract evidence sentences on the basis of a claim. The system selects sentences using bi-gram TF-IDF with binning. This system will be considered as a baseline in our experiments presented in the following sections.

Even though similar approaches are used in open-domain question answering and automated fact-checking, important differences exist. Whereas in automated fact-checking the query is a factual claim, in question answering the query is a question. Moreover, evidences also differ from candidate answers. In question answering the question can often be answered by identifying the correct entity in a text and in automated fact-checking, sentence-level or clause-level evidence need to be found for the validation of the claim.

6.2.3 Summary

The work on argument detection in argumentation mining is similar to our evidence extraction problem setting, as the goal is to find premises or claims that support or refute a topic or a claim. Nevertheless, whereas in argumentation mining evidences for a controversial topic or a claim need to be aggregated, in automated fact-checking evidences for a factual claim need to be found.

Open-domain question answering is a task similar to evidence extraction and for both tasks, similar approaches are used. However, whereas in automated fact-checking the query is a factual claim, in question answering, the query is a question. Moreover, evidences for a claim are different to candidate answers as they are typically represented by entire sentences or clauses, whereas answers in question answering can also be multi-word expressions, such as named entities.

6.3 Machine learning approaches

In our experiments in this chapter, we consider evidence extraction as a ranking problem and evaluate the performance of a number of models on this task. In this section, we present the architectures of these models, the approach of how the models are trained, and the configuration of the models during testing.

6.3.1 Model architectures

We have performed experiments with three models, a newly developed model `BiLSTM` and two existing models, `DecAtt` and `ESIM`, which we modified to rank the candidate evidence sentences on the basis of a given claim. Below, we only present `BiLSTM` and `ESIM` since `DecAtt` was already introduced in Section 5.4.1.

Stacked-BiLSTM (BiLSTM)

We have developed a Siamese model, which has two encoders, one for the claim and the other one for the evidence candidate sentence. The model architecture is illustrated in Figure 6.1.

The two encoders are identical and share the same parameters. The encoders are based on two stacked Bidirectional LSTM layers (BiLSTMs) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997). We take the last hidden state of the upper BiLSTM as the representation of the claim or the candidate evidence sentence respectively. The representations of the claim and the candidate evidence sentence are concatenated and fed through a Multi-Layer Perception (MLP). In the last layer of the MLP, there is only one single neuron that predicts the ranking score. The ranking score indicates how relevant the current candidate evidence sentence for the validation of the claim is. The resulting model will be called `BiLSTM`.

Enhanced Sequential Inference Model (ESIM)

The Enhanced Sequential Inference Model (`ESIM`) (Chen et al., 2017b) was originally developed for determining the entailment relation between a hypothesis sentence and a premise sentence in the SNLI task (Bowman et al., 2015). The model is similar to `DecAtt` introduced in Section 5.4.1, since the two sentences are compared on the token level before the entailment relation is predicted. However, whereas `DecAtt` only relies on the token interaction, `ESIM` additionally uses LSTMs. Chen et al.

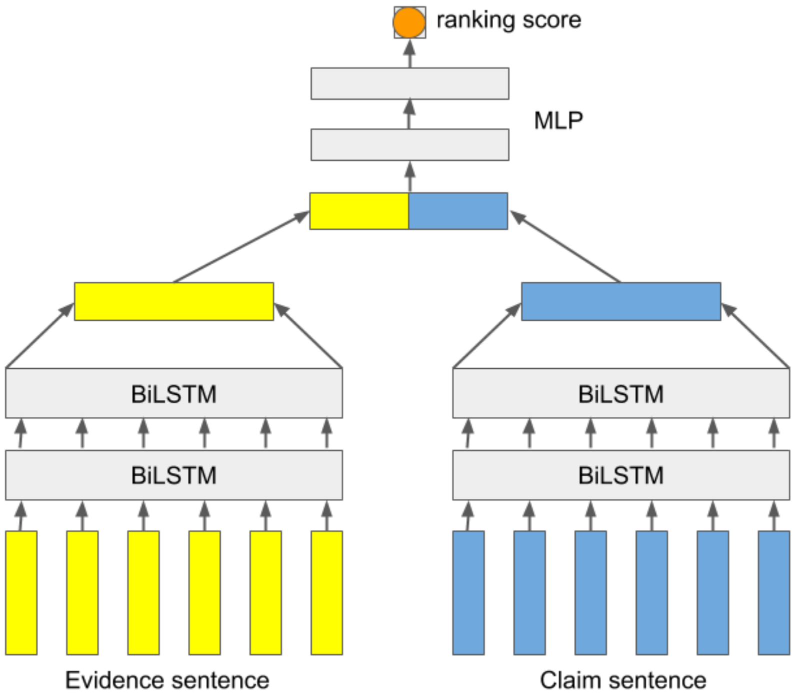


Figure 6.1: Siamese stacked BiLSTM model (**BiLSTM**)

(2017b) proposed two versions of the `ESIM`, both of which are illustrated in Figure 6.2. The model displayed on the left-hand side computes contextual representations of the tokens based on the sequence of tokens in the sentence using BiLSTMs. The model depicted on the right-hand side computes token representations based on the syntactic parse tree of the sentence which is traversed by Tree-LSTMs. In our experiments, we are only using the BiLSTM version of the model depicted on the left-hand-side of the Figure. Below, we give a short description of the BiLSTM-ESIM by presenting its three levels of computation.

Input encoding: The embeddings of the tokens of the two sentences a_j and b_i (in our case the candidate evidence sentence and the claim) are feed through a BiLSTM, and output representations $\hat{a}_j$ and $\hat{b}_i$ of the individual tokens are obtained. Since a BiLSTM takes the information about the tokens from both sides of the sentence into account, the computed token representations $\hat{a}_i$ and $\hat{b}_i$ are enhanced with contextual information from neighboring tokens.

Local inference modeling: Each token of one sentence $\hat{a}_j$ is used to compute attention weights with respect to each token $\hat{b}_i$ in the other sentence giving rise to an attention weight matrix w_{ij}. Then, each contextualized token representation $\hat{a}_j$ and $\hat{b}_i$ is multiplied by all of its attention weights, and weighted pooling is applied in order to compute a single representation for each token (α_j or β_i) on the basis of all the tokens in the other sentence:

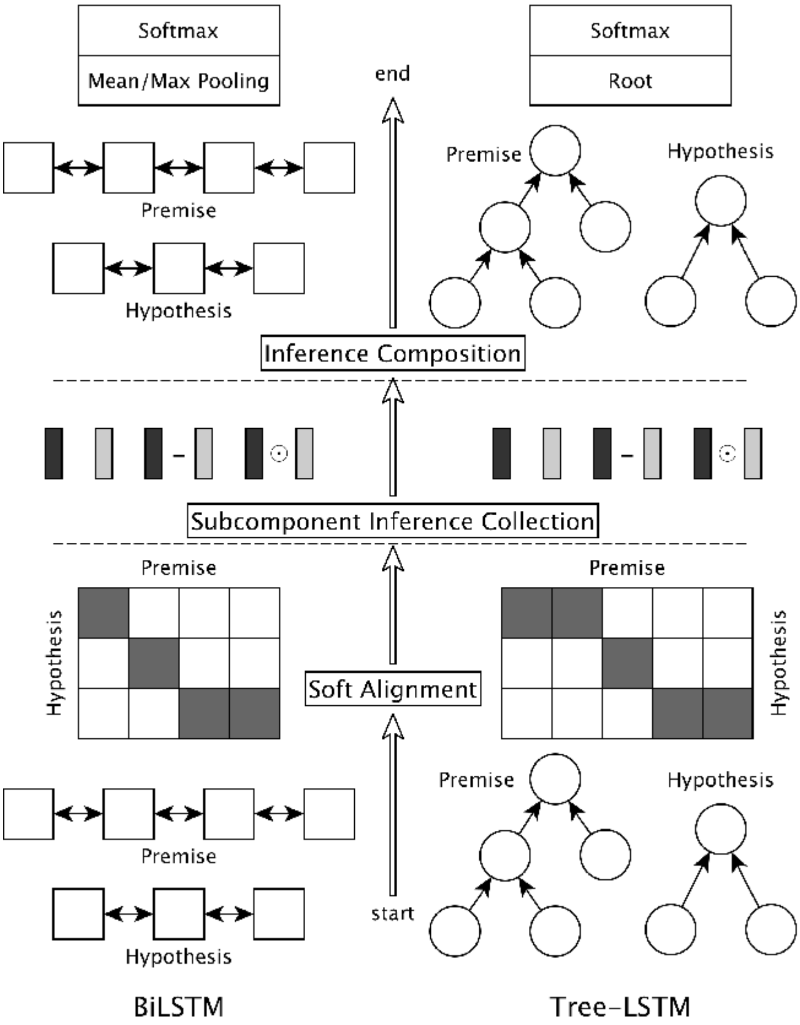


Figure 6.2: Enhanced Sequential Inference Model (ESIM) (source: (Chen et al., 2017b))

$$\begin{aligned} \beta_i &= \sum_{j=1}^{n} w_{ij}\, \hat{b}_i, \\ \alpha_j &= \sum_{i=1}^{n} w_{ij}\, \hat{a}_j. \end{aligned} \tag{6.1}$$

This step is identical to **Attend** step for the `DecAtt` model, but the attention operation is performed with *contextualized* embeddings instead of *global* embeddings of the tokens.

Inference composition: The two token sequences α_j and β_i are fed through another BiLSTM, which again computes sequences of representations for each sentence $\hat{\alpha}_j$ and $\hat{\beta}_i$. Maximum and average pooling is then applied to the two sequences in order to derive two vectors $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$. These vectors are combined in three different ways [$\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\beta}}$; $\hat{\boldsymbol{\alpha}}$ - $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\alpha}} \otimes \hat{\boldsymbol{\beta}}$] giving rise to the last hidden state of the `ESIM`. This vector is then fed into an MLP for the classification of the entailment relation.

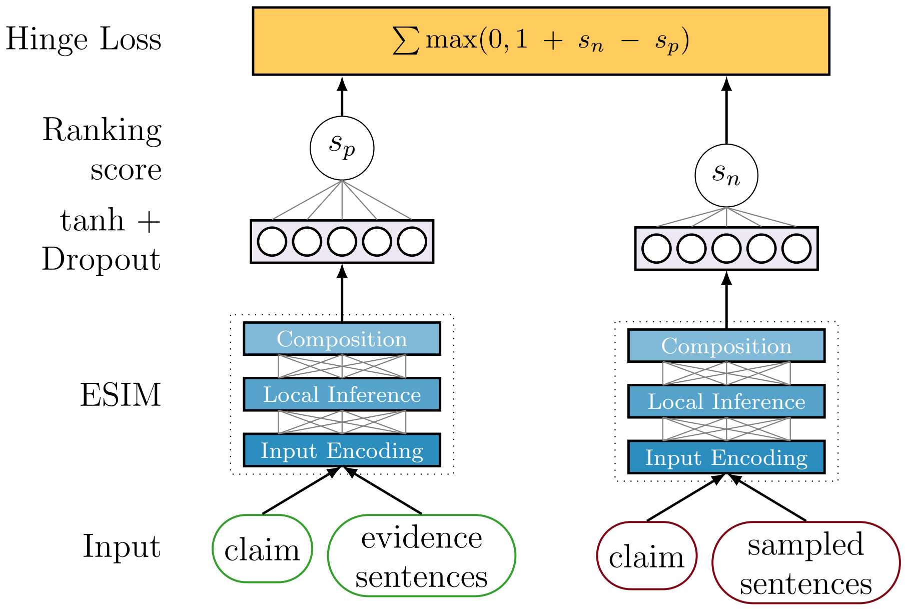


Figure 6.3: Training configuration of the evidence extraction model

6.3.2 Training configuration

The three models `BiLSTM`, `DecAtt` and `ESIM` represent different approaches to combine two sentences in order to classify their relation. Even though `DecAtt` and `ESIM` were originally designed for classification, they compute a rich representation of the two compared sentences which can also be used for ranking. Thus, we use these models as encoders in order to derive a joint representation of a candidate evidence sentence and the claim that can be used to compute a ranking score. This ranking score shall indicate how relevant the given sentence is for the validation of the claim. The training configuration of the models is illustrated in Figure 6.3, where we depicted `ESIM` as the encoder of the system. When using `DecAtt` or `BiLSTM` as an encoder, we only need to replace the `ESIM` with the other encoder in the system.

The encoder takes as input a claim and a candidate evidence sentence and outputs a vector that represents the interaction between the two sentences. The output vector is then fed through a hidden layer that is connected to the single neuron predicting the ranking score. The models is trained to predict the ranking score for positive and negative samples. As a loss function, we use a modified hinge loss with negative sampling: $\sum max(0, 1 + s_n - s_p)$, where s_p indicates the ranking score of a positive sample and s_n represents the ranking score of a negative sample. To obtain s_p, we feed the network with a claim and a ground truth evidence sentence. To compute s_n, we take all documents, from which the ground truth evidence sentences for the claim originate, randomly sample one sentence (not including ground truth evidence sentences), and feed this sentence with the claim into the encoder. With our modified hinge loss function, we are able to maximize the margin between positive and negative samples. The ranking score for the positive samples is thereby pushed towards one and the ranking score of the negative samples towards minus one.

6.3.3 Testing configuration

At test time, a claim is combined with each sentence in the retrieved documents and the pair is fed into the trained model to predict the ranking scores. Then, the sentences are ranked in the descending order based on their ranking scores, and we choose the top-k highest-ranked sentences as evidence. The three resulting ranking models will be named `rankingESIM`, `rankingDecAtt`, and `rankingBiLSTM` henceforth.

6.4 Experiments and results

In order to evaluate the performance of the three models in the evidence ranking setup, we measure the precision and recall on the five highest ranked sentences (precision @5 and recall @5). This setting is identical to the evaluation procedure used in the FEVER shared task to evaluate the evidence ranking models. As encouraged by the FEVER evaluation metric, we also believe that a model, which is able to retrieve most of the evidence sentences with relatively low precision is more valuable than a model that is only able to retrieve a subset of evidence sentences, though with high precision. Thus, recall @5 is considered as the decisive evaluation metric. We evaluate the performance of the models using this metric on the **FEVER18** and the **Snopes19** datasets and present the results below in two separate sub-sections.

6.4.1 Experiments on FEVER18

In this subsection, we perform a number of experiments with the introduced ranking models in order to evaluate their performance and find the best configuration for the best performing model. Moreover, we discuss the results of the FEVER evidence extraction sub-task.

Comparison of different ranking models

In our experiments discussed below, we compare the performance of our three ranking models to the FEVER baseline `TF-IDF`. The results of the different systems on the development set of **FEVER18** are illustrated in Table 6.1. As can be noticed, `rankingESIM` performs best from all the compared systems. However, the FEVER baseline `TF-IDF` also reaches good results, and outperforms the more complicated models `rankingDecAtt` and `rankingBiLSTM`. This indicates that the lexical overlap between the claim and the candidate evidence sentence is an important feature for evidence ranking. Since `rankingBiLSTM` outperforms `rankingDecAtt`, we assume that the contextual information computed by LSTMs is important in the evidence ranking task. Since the `rankingESIM` outperforms all other models, we use this model in the experiments below.

Evidence ranking with rankingESIM

In Table 6.2, we show the performance of our document retrieval system (presented in Section 4.3.2) and the `rankingESIM` when retrieving different numbers of the highest-ranked Wikipedia documents. The results show that both systems benefit from a

model	precision @5	recall @5
rankingESIM	**0.839**	**0.862**
TF-IDF (baseline)	0.764	0.859
rankingBiLSTM	0.541	0.832
rankingDecAtt	0.415	0.719
random baseline	0.206	0.602

Table 6.1: Comparison of different models in the evidence ranking problem setting of the **FEVER18** corpus

larger number of retrieved documents. However, whereas for document retrieval the performance always improves when more documents are considered, for evidence extraction, the recall slightly decreases if we retrieve 10 Wikipedia documents. This problem arises from the fact that when 10 documents are retrieved, there is a larger number of sentences from which the evidence needs to be extracted. Since there is an increased amount of *noise*, in terms of a larger number of sentences that are somehow related to the claim but are not considered as evidence, the problem becomes more difficult. Thus, because we reach the best performance for 7 documents, we will use this setting in the following experiments.

# docs	doc. recall	evidence recall @5
1	0.8463	0.8208
3	0.8890	0.8537
5	0.8994	0.8602
7	0.9033	**0.8624**
10	**0.9049**	0.8615

Table 6.2: Performance of our document retrieval system and `rankingESIM` when using different numbers of MediaWiki search results

As mentioned above, in the FEVER shared task, recall @5 was considered as an evaluation metric. However, we also analyze, how a different number of the selected evidence affect the performance, that is, recall @4, recall @3, ... The results in Table 6.3 show that whereas recall increases if more evidence sentences are considered, the precision and the F1 score decrease. This indicates that even though we are able to retrieve a larger number of correct evidence sentences, the noise in the retrieved set of evidence increases. In fact, since the **FEVER18** corpus in many cases provides only one or two evidence sentences, the selected set of five sentences necessarily includes unrelated sentences.

FEVER shared task evidence extraction sub-task

In order to maximize the performance in the FEVER shared task, we used an ensemble of 10 `rankingESIM` models for sentence ranking. To construct the ensemble, we trained 10 `rankingESIM` models with different randomisation seeds. At testing

# sentences (k)	precision @k	recall @k	F1 @k
1	**0.7855**	0.7216	**0.7522**
2	0.4857	0.7216	0.6075
3	0.3576	0.8421	0.5020
4	0.2862	0.8599	0.4294
5	0.2402	**0.8624**	0.3767

Table 6.3: Performance of the `rankingESIM` on the **FEVER18** corpus when different numbers of sentences are selected

User	Team Name	precision @5	recall @5	F1 @5
chaonan99	UNC-NLP	0.4227	0.7091	0.5296
tyoneda	UCL	0.2216	0.8284	0.3497
littsler	Athene UKP	0.2361	**0.8519**	0.3697
papelo		**0.9218**	0.5002	**0.6485**
chidey		0.1848	0.7539	0.2969
Tuhin	ColumbiaNLP	0.2302	0.7589	0.3533
Wotto		0.1209	0.5169	0.1960
FEVER base.		0.1826	0.4422	0.1866

Table 6.4: Results of the evidence ranking part of the FEVER shared task

time, the claim is individually combined with all candidate evidence sentences and fed into each `rankingESIM` in order to compute 10 ranking scores for each claim-sentence pair. Then, the mean score of each claim-sentence pair over all 10 models of the ensemble is calculated. The candidate evidence sentences are ranked according to these scores, and the five highest-ranked sentences are then taken as an output of the model.

The results of the FEVER evidence extraction task are illustrated in Table 6.4. It must be noted that the performance of the ranking models also depend on the document retrieval system, and thus, the results are indicative of the performance of both systems combined. As the table illustrates, our document retrieval and evidence ranking systems reach the highest recall @5, which was the official metric for the FEVER shared task evidence extraction problem. In fact, as can be observed in Table 6.4, only the team *UCL* reaches a similar performance. The recall score of their model is only about 2.4% lower than the score of our model. The user *papelo*, only predicted one sentence as evidence, and was therefore able to reach the best precision @5. Nevertheless, this strategy leads to low recall @5, since often two evidence sentences had to be identified.

6.4.2 Experiments on Snopes19

As described in Section 3.2.1, for the **Snopes19** corpus, we define evidence extraction as the identification of Fine-Grained Evidence (FGE) in Evidence Text Snippets (ETS). This problem setting is similar to the identification of evidence sentences for

model	recall m	precis. m	F1 m
upperBound	0.769	0.725	0.746
random baseline	0.500	0.500	0.500
majority vote	0.343	0.500	0.407

Table 6.5: Baselines and the human upper bound for evidence extraction task of the **Snopes19** corpus, if the problem is considered as a classification task (m = macro)

model	precision @5	recall @5
rankingBiLSTM	0.451	**0.637**
rankingDecAtt	0.420	0.627
TF-IDF	**0.627**	0.601
rankingESIM	0.288	0.507
random baseline	0.296	0.529

Table 6.6: Comparison of different models on the evidence ranking problem for the **Snopes19** corpus

the **FEVER18** corpus, where sentences of the introductory sections of Wikipedia articles need to be ranked according to their relevance for the validation of the claim. Since ETSs are similar in size compared to introductory sections of Wikipedia articles, the FEVER problem setting will be considered as a reference task. We therefore frame evidence extraction for the **Snopes19** corpus also as a ranking problem. Nevertheless, in Table 6.5, we provide the human upper bound, the random baseline, and majority vote scores for evidence extraction as a *classification problem* for future reference.

In Table 6.6, we show the performance of the four discussed models on the **Snopes19** corpus. Compared to the performance of the models on the **FEVER18** corpus illustrated in Table 6.1, the results are very much different. In terms of recall @5, the neural networks with a small number of parameters `rankingBiLSTM` and `rankingDecAtt` perform best. The `TF-IDF` model reaches best results in terms of precision. The `rankingESIM` reaches a relatively low score and is not able to beat the random baseline. We assume this is because the model has a large number of parameters and requires more training instances than provided by **Snopes19**.

6.5 Error analysis

6.5.1 Error analysis FEVER18

For **FEVER18**, we performed an error analysis for the `rankingESIM` as the highest-scoring model. Nevertheless, in a number of cases, the correct evidence sentences could not be retrieved not because of the shortcomings of the sentence ranking model but for other reasons. One common error, which is independent of the ranking model, is that the document retrieval system is not able to retrieve all documents containing the evidence sentences. As shown in Table 6.2, the best document retrieval system is still not able to identify all the required documents for almost 10% of the claims. This reduces the upper bound of the evidence ranking model.

Furthermore, the `rankingESIM` retrieves in many cases a reasonable set of evidence sentences, but these sentences have not been labeled as evidence by the annotators. For example, the claim "*The Bee Gees wrote music.*" can be supported by the identified sentence "*The Bee Gees wrote all of their own hits, as well as writing and producing several major hits for other artists.*" which was not annotated as evidence. In fact, Wikipedia is too large to be exhaustively explored for evidence by a small group of annotators. This was also acknowledged by the FEVER shared task organizers. Thus, in the evaluation phase of the FEVER shared task, human annotators evaluated the evidence sentences predicted by the competing systems. It was therefore possible to give credit to systems that identified valid evidence sentences, but which have not yet been labeled as such.

Another frequent case, in which the correct evidence sentences could not be identified, is when one entity mention in the claim did not occur in the annotated evidence sentences. E.g. for the claim "*Daggering is nontraditional.*" there is only one annotated evidence sentence "*This dance is not a traditional dance.*". Here, *this dance* refers to *daggering*, but this information is not given. Thus, this case cannot be resolved by our model. For some claims, one of the evidence sentences is less related to the claim and it is therefore not identified by our model. E.g. the claim "*Herry II of France has three cars.*" has the two evidence sentences: "*Henry II died in 1559.*" and "*1886 is regarded as the birth year of the modern car.*". The second sentence is important for the validation of the claim, but it has a low lexical and semantic overlap with the claim, and it is therefore ranked very low by our model.

Summary. As our analysis shows, the encountered errors can often be traced back to failures of the document retrieval system and issues with the dataset. They can therefore not be directly addressed by improving the evidence extraction model. Nevertheless, we have also found instances, where the evidence sentences could in principle be identified, if a more sophisticated model was used. For instance, coreference resolution approaches would help to find the evidence sentence in the *daggering dance* example. If the model had some kind of world knowledge and was able to link semantically distant sentences that in combination could help to validate the claim, then the claim in the *Henry II* example could be solved.

6.5.2 Error analysis Snopes19

For **Snopes19**, we perform an error analysis for the `rankingBiLSTM` and the `TF-IDF` system, as they reach the highest recall and precision, respectively. The `TF-IDF` system achieves best precision because it only predicts a small set of sentences that have lexical overlap with the claim. The model therefore misses FGE, in which the claim is paraphrased and synonyms instead of the same words are used. The `rankingBiLSTM` is better able to capture the semantics of the claim and the FGE, and we believe that it was therefore able to reach a higher recall. In fact, we have found that in order to identify the correct FGE, the model had to make the connection between semantically related words, such as "*Israel*" - "*Jewish*", "*price*"-"*sold*", "*pointed*"-"*pointing*", "*broken*"-"*injured*". Nevertheless, the model fails when the relationship between the claim and the FGE is more elaborate, e.g. if the claim is not paraphrased, but reasons for it being true are given. We have also found that refuting FGE is more difficult to identify since the semantic and lexical overlap between the FGE and the claim is less pronounced. Moreover, as illustrated by the Example 6.2 below, the model wrongly assigns a high ranking score to a sentence, when the topic of the sentence is similar to the topic of the claim but the sentence is not relevant for the validation of the claim.

Claim: *The Department of Homeland Security uncovered a terrorist plot to attack Black Friday shoppers in several locations.*
Sentence: *Bhakkar Fatwa is a small, relatively unknown group of Islamic militants and fanatics that originated in Bhakkar Pakistan as the central leadership of Al Qaeda disintegrated under the pressures of U.S. military operations in Afghanistan and drone strikes conducted around the world.*

Example 6.2: A sentence wrongly assigned a high ranking score because of semantic relatedness with the claim

Another frequently encountered error is when several sentences only in combination form a consistent narrative that supports or refutes the claim. For example, the claim "*Vintage color photograph from 1941 shows the Japanese attack on Pearl Harbor.*" has the ground truth sentences "*The 7 December 1941 Japanese raid on Pearl Harbor was one of the great defining moments in history.*" and "*Watch these Amazing color photos of the attack.*" In isolation, the second sentence is only marginally related to the claim, and the model therefore fails to identify this sentence as evidence.

Summary. As our analysis shows, the evidence ranking system fails in more difficult cases, for which features, such as lexical overlap or semantic relatedness, are not sufficient. In fact, we have found that compared to the errors of the `rankingESIM` in the FEVER shared task discussed in the previous section, the errors of the `rankingBiLSTM` and the `TF-IDF` system discussed above are more rudimentary. The `rankingESIM` is more elaborate and has more parameters, which, on the one hand, leads to improved performance, but on the other hand, requires a larger corpus for training. Since the **Snopes19** corpus does not contain sufficient instances for training the model, pre-trained language models, such as BERT (Devlin et al., 2018) or XLNet (Yang et al., 2019b), could be explored in the future. Such models are trained on large collections of unlabeled text in a self-supervised manner and can then be fine-tuned on a small corpus. After fine-tuning, the models in general reach a high performance on the small corpus.

6.6 Chapter summary

In this chapter, we analyzed the evidence extraction problem, which is the task of identifying evidence sentences in documents using a claim as a query. The task can be either considered as binary classification, where sentences in the retrieved documents are classified as evidence or not evidence, or as a ranking problem. In the ranking problem setting, sentences from all the retrieved documents are first ranked according to their relevance for the validation of the claim and then the k-highest ranked sentences are taken as evidences. Similar to the FEVER shared task, we have considered evidence extraction as a ranking problem. For the ranking of the evidence sentences, we have presented three neural models, which have different levels of complexity, and compared their performance to the FEVER shared task TF-IDF baseline. We have evaluated the performance of the models on the two corpora: **FEVER18** and **Snopes19**.

For **FEVER18**, our neural model based on the ESIM architecture reached best results. For **Snopes19**, a simpler BiLSTM based model and the TF-IDF baseline performed best. We conclude that the results for the two corpora are different because **Snopes19** is an order of magnitude smaller than **FEVER18**. The more sophisticated model based on the ESIM has probably too many parameters to be effectively trained on the smaller **Snopes19** corpus.

Our error analysis on both corpora has shown that the correct evidence sentences cannot be identified for a number of reasons: (1) The documents containing the required evidence may not be retrieved by the document retrieval system. (2) The correct evidence sentences are lexically and semantically different from the claim, and because the models rely on these features, such sentences could often not be identified. (3) The evidence sentences contain anaphoric pronouns and the models therefore often fail to make the connection between these sentences and the claim. (4) World knowledge is required to infer the relevance of a sentence for the validation of the claim.

Since the error sources are very diverse, we have concluded that different strategies need to be followed to improve performance. For the FEVER evidence ranking problem, models are required which can resolve anaphoric pronouns and are able to incorporate world knowledge in the ranking process. Since the models already achieve a relatively high score on the FEVER dataset, and world knowledge cannot be easily incorporated in practice, significant performance gains for the **FEVER18** can probably not be achieved.

The errors occurring on the **Snopes19** corpus are, on the other hand, more rudimentary, and therefore, more elaborated approaches could help improve performance for this corpus. Most promising in this case are pre-trained language models such as BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019b). These models are trained on large collections of text and are able to generalize to a new task if fine-tuned only on a small corpus.

Chapter 7

Claim validation

Claim validation is the problem of determining the verdict for a claim on the basis of collected evidence, and it represents the last step in our fact-checking pipeline (Section 1.2.2). A system capable of validating a claim with high accuracy would be of enormous value, as it will be possible to validate a large number of claims emerging on the web and thus, help to control the spread of false information (see also the discussion in Section 1.2.2). However, the task is very difficult as the system needs to understand the often complex relationship between the evidence and the claim and be able to weight contradicting evidence against each other. Moreover, the accumulation of errors through the pipeline makes it additionally more difficult to determine the verdict. It is not guaranteed that the correct set of evidence was retrieved, let alone there is such a set in the queried document collection. As described in Section 1.2.2, the goal of the proposed fact-checking pipeline is therefore not to take over the fact-checking task entirely, but to assist the fact-checker in order to speed up the process. Thus, our objective in this chapter is to investigate in which cases we can achieve good results for claim validation, and what the outstanding challenges are.

The chapter is structured as follows. In Section 7.1, we discuss the claim validation problem setting and provide an example. Section 7.2 gives an overview of work related to claim validation. In Section 7.3, we analyze the *Recognizing Textual Entailment* (RTE) problem of the FEVER shared task which is similar to our definition of the claim validation problem. We introduce a new model for the RTE task and perform an error analysis for the misclassified instances of this model. In Section 7.4, we present experiments with a number of different claim validation models on the **Snopes19** corpus, perform an error analysis, and discuss outstanding challenges.

The contributions of this section are the following.[1]
(10) We present our model for claim validation that reached the third rank for *recognizing textual entailment* in the FEVER shared task, and analyze the performance of the model in an error analysis.
(4) We perform a large number of experiments on our newly created Snopes corpus with models from the FEVER shared task and other successful approaches suitable for the claim validation sub-tasks. Based on the conducted experiments, we perform an analysis of the claim validation problem setting defined by our corpus **Snopes19**

[1] The complete list of contributions ranging from 1 to 10 is given in Section 2.1.1.

and compare it to the claim validation problem defined by **FEVER18**.

7.1 Problem setting

As described above, we define claim validation as the problem of predicting the verdict for a claim on the basis of a set of aggregated evidences. The set of evidences is represented by a number of sentences that provide valuable information for the validation of the claim (see also the definition in Section 6.1). We analyze the task on the basis of the two corpora **FEVER18** and **Snopes19**. For both corpora, we are given one or several evidence sentences based on which the verdict needs to be determined. In our experiments, we use for both problem settings the FEVER shared task classification scheme, that is, a claim needs to be classified as *supported*, *refuted*, or *not enough info*. The labels *supported*, and *refuted* correspond to the verdicts *true* and *false* respectively. The *not enough info* label is given if the collected evidence sentences do not provide sufficient information to label the claim as *supported* or *refuted*.

A challenging claim validation problem is presented in Example 7.1. The given claim needs to be validated on the basis of three sets of evidence sentences, each of which is extracted from a different document. The classification problem is difficult as the claim is false, but there are two sets of sentences which *support* the claim and only one sentence set which *refutes* the claim. Thus, a deeper analysis of the case is required. The first two sentences sets are from the same source, that is *General Al-Saudi*, and they are therefore not independent of each other and can be treated as one piece of evidence. The evidence set three attacks the claim by pointing out that a story published by the Daily Mail, which supports the claim (most likely the origin of one of the two other sentence sets), is false because there are *no dams in southern Israel*. It further states that the original Daily Mail story was altered (supposedly after noticing the mistake). All these are hints that the claim is false, however, without additional contextual information, the case is difficult to resolve even for humans. We will discuss such instances in more detail in the analysis Section 7.4.2 of this chapter. It must be noted that even though we have discussed the role of sentence sets (evidence sentences coming from the same source) and the stance of the sentences in the claim validation process, this kind of information is not considered in the claim validation problem setting discussed below. As we have pointed out in the previous paragraph, the task is only to classify the claim on the basis of the collected evidence sentences.

7.2 Related work

There has been a considerable amount of work in the area of claim validation, even though the task is often given a different name and it is framed slightly differently compared to our definition of the problem. In the following, we present work which we consider to be most relevant to our definition of the claim validation problem. We split the related work section into three sub-sections: early work on claim validation, advanced claim validation approaches, and the FEVER RTE problem setting.

Claim: Israel caused flooding in Gaza by opening river dams

Verdict: **refuted**

Collected evidence sentences:

Evidence set 1 *"Israel opened water dams, without warning, last night, causing serious damage to Gazan villages near the border," General Al-Saudi told Al Jazeera.*

Evidence set 2 *Hundreds of Palestinians left homeless after Israel opens river dams and floods houses General Al-Saudi said that the dams were opened without warning.*

Evidence set 3 *The Daily Mail published a story on Monday that originally accused Israel of intentionally opening dams in southern Israel in order to flood Gaza. The only problem is, ... there are no dams in southern Israel. Honest Reporting ... took screen shots of the article before amendments were made.*

Example 7.1: Claim validation on the basis of collected evidence

7.2.1 Early work on claim validation

Early work in claim validation was done on relatively small corpora and the problem setting was often very restricted. Wang (2017) performed experiments on the **PolitiFact17** corpus (Section 3.1.1) in two different settings: predicting the verdict only on the basis of the claim itself, or additionally considering the meta-information (such as the name of the speaker, his/her party affiliation etc.) for the prediction.

Derczynski et al. (2017) have introduced the dataset **RumEval17** (Section 3.1.1) for the RumourEval shared task. The organizers suggested two variants of the task: prediction of the verdict only on the basis of rumor (claims), or using additional information, such as Wikipedia, for the prediction. Nevertheless, for the second setting, no additional annotations were provided.

Another study concerned with the validation of rumor was performed by Dungs et al. (2018). They proposed a Hidden Markov Model (HMM) to predict the verdict for rumors on Twitter. As features, Dungs et al. (2018) only used the stance of the tweets that referred to the rumor and the timestamps of these tweets.

bar introduced the corpus **CLEF-2018** (Section 3.1.1) for the CLEF-2018 shared task. In the second part of the shared task, participants had to classify claims as *true*, *false*, or *half-true*.

The corpora of the studies discussed above are relatively small and in most cases, no evidence for the validation of the claims is considered. From our point of view, such a definition of claim validation is problematic. If no evidence is considered, the models most likely only learn biases with respect to the words in the claim rather than to differentiate between true and false claims. Moreover, since the number of claims in all corpora is relatively low, most models trained on these corpora reach a relatively low performance, and they are unlikely to be able to generalize to other datasets.

7.2.2 Advanced claim validation approaches

Popat et al. (2017) constructed the substantially larger corpus **Snopes17** containing 4,956 claims from the Snopes and Wikipedia web-pages. However, even though they provide additional documents for the validation of the claims, the documents have been collected not by human experts but using the Google search engine. The documents can therefore not be considered as a gold standard. Popat et al. (2017) proposed two models for the claim validation task which make use of hand-engineered features: a model based on *Distant Supervision* and a *Joint Model* based on a Conditional Random Field. Even though the Joint Model is more sophisticated, it is outperformed by Distant Supervision by 2% (80% vs. 82%).

Popat et al. (2018) proposed a neural network architecture `DeClarE` for *"debunking false claims"*. The claim's verdict is predicted on the basis of the claim, its source, and the evidence documents with their sources (URLs of the web-documents). The model makes use of an attention mechanism that should increase performance, and because it allows highlighting passages in the documents, it should also help to explain why a particular prediction was made. Popat et al. (2018) perform experiments on the corpora **Snopes17**, **RumEval17**, and two datasets based on the PolitiFact[2] and the NewsTrust[3] websites. On three of these corpora, the model is able to achieve a new state-of-the-art.

7.2.3 FEVER RTE problem setting

The corpus **FEVER18** provided by the FEVER shared task organizers is substantially larger than all previous fact-checking datasets. It therefore allows training machine learning models with larger complexity. which shall lead to higher performance. As described in Section 3.1.2, in the FEVER shared task, a pipeline had to be constructed for three fact-checking sub-tasks. For the Recognizing Textual Entailment (RTE) sub-task (which corresponds to claim validation) the FEVER organizers proposed two baselines: `DecAtt` (Section 5.4.1), which was developed for the SNLI (Bowman et al., 2015) task, and the `UCLMR` MLP introduced for the FNC stance detection task by Riedel et al. (2017) (see Section 5.3.1). Even though the two models were designed for different tasks (NLI and stance detection), because the problem settings are structurally identical to claim validation, both models can be used for the FEVER RTE task. Since `DecAtt` is superior to `UCLMR`, this model serves as a baseline for the experiments in this chapter. We consider the FEVER shared task problem setting as the most comprehensive definition of the automated fact-checking problem, and thus, in Section 7.4, we use the FEVER RTE sub-task as a reference for our analysis of the claim validation problem defined for the **Snopes19** corpus.

There has been much work on the FEVER RTE sub-task, which is beyond the literature review given in this section. Thus, in the following, we only discuss studies that are most relevant to our analysis in this chapter.

The UNC-NLP team (Nie et al., 2019) won the RTE sub-task reaching an accuracy score of 67.98%. Nie et al. (2019) called their system *Neural Semantic Matching*

[2] `https://www.politifact.com/`

[3] `https://news.trust.org/`

Network which takes as input the concatenated evidence sentences and the claim and predicts the verdict. For the representation of the tokens of the claim and the evidence sentences they used GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018) embeddings, and other features. The two input sequences representing the claim and the evidence sentences are feed through a network that is similar to the ESIM (Section 6.3.1). More specifically, BiLSTM layers are used in order to enrich the tokens with contextual information and an attention mechanism allows the model to identify important information in the evidence sentences on the basis of the claim and vice versa.

UCL Machine Reading Group (Yoneda et al., 2018) came in second in the RTE sub-task reaching a verdict accuracy of 67.44%. They constructed a model based on ESIM (Chen et al., 2017b) and pre-trained it on the SNLI (Bowman et al., 2015) dataset, before fine-tuning it on **FEVER17**. The model is designed in such a way that evidence sentences are individually combined with the claim and are feed into different ESIMs. Each ESIM computes a probability distribution for the three classes. These predictions for each claim-sentnnce pair are multiplied by the ranking scores of their sentence selection model (for the current sentence) and then feed through a MLP for the final classification.

Our system `Athene` (Hanselowski et al., 2018b) achieved a verdict accuracy of 65.22% and came in third. Similar to the UCL Machine Reading Group, we used the ESIM as a building blog in our network architecture. We also fed sentence-claim pairs into individual ESIMs, but in contrast to weighting the predictions by the scores of the evidence ranking model, we used an attention mechanism to weight the individual outputs of the ESIM for each sentence-claim pair (see Section 7.3.1).

Yin and Roth (2018) have not participated in the FEVER shared task but have also proposed a pipeline for the three FEVER sub-tasks. They introduced a model that in a multitask setting jointly learns to select the evidence sentences and predict the verdict for the claim. Moreover, the model has a *two channel mechanism* meaning that during the classification process an attention mechanism is not only applied between the claim and the evidence sentences (as by the top three models in the FEVER RTE sub-task), but also between a given evidence sentence and other remaining evidence sentences. This shall enable the model to predict the verdict on the basis of several facts distributed across different evidence sentences (see Example 3.1 in Section 3.1.2). The resulting model reaches an accuracy score of 75.99% on the unofficial test-set released before the shared task. Even though the results are not directly comparable, as the top three systems are evaluated on the official test-set, the performance gains over the three model are substantial.

7.2.4 Summary

Early work on claim validation was done on relatively small corpora and the problem setting was often very restricted. The corpora did not provide annotation of evidence and the verdict for the claim was often only predicted on the basis of the claim itself. This problem setting is problematic, because if no evidence is provided, the trained systems can only exploit regularities in the dataset for the prediction of the verdict. Moreover, the systems have been trained only on single-domain corpora, which means that they are unlikely to generalize to other domains.

Popat et al. (2017, 2018) used a number of substantially larger multi-domain corpora for training claim validation systems. For the validation of claims, they explored elaborated machine learning approaches, such as distant supervision, conditional random fields, and attention based neural networks. These approaches are more promising compared to early work on claim validation as they reach relatively high accuracy scores of about 80%.

The corpus **FEVER18** constructed by the FEVER shared task organizers is substantially larger than all previous fact-checking datasets, and it provides more annotations for the claim validation problem than previous corpora. The corpus received much attention in the NLP community, and a number of different neural network architectures have been proposed for the FEVER claim validation problem.

In this chapter, we present our approach to tackle the FEVER claim validation problem that reached the third rank in the FEVER shared task. Moreover, we compare claim validation experiments on our **Snopes19** corpus to the experiments on the FEVER corpus as the most comprehensive dataset introduced so far.

7.3 Experiments on FEVER18

In this section, we first present the model that we have developed for the FEVER claim validation problem (RTE sub-task) and discuss the results of the FEVER shared task. The performance of our model is then evaluated in an error analysis. We conclude the section with a brief discussion of the major takeaways from the FEVER shared task.

7.3.1 Model architecture

Our model illustrated in Figure 7.1 is an extension of the ESIM (Section 6.3.1). In contrast to the ESIM, which can only predict the entailment relation between a hypotheses sentence and a premise sentence, our model can predict the relation between *multiple* input sentences and the claim as required in the FEVER RTE sub-task.

For the prediction of the verdict (*supported*, *refuted* or *not enough info*), we use the five sentences retrieved by our sentence selection model (Section 6.3.1). For the representation of the tokens of the claim and the evidence sentences, we concatenate the GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017) embeddings. Since both types of embeddings are pretrained on Wikipedia, they are particularly suitable for the FEVER RTE problem setting. To process the five input sentences using the ESIM, we combine the claim with each sentence and feed the pairs into 5 ESIMs. The pairs are analogous to the hypotheses-premise pairs in the original SNLI setting. The last hidden states of the ESIMs computed for the five individual claim-sentence pairs are compressed into one vector using an attention mechanism and pooling operations.

The attention mechanism is based on representations of the claim and the five evidence sentences. These representations are obtained by average pooling over the outputs of the first BiLSTM of the ESIM (Section 6.3.1). For each claim-sentence pair, the sentence representation and the representation of the claim are individually

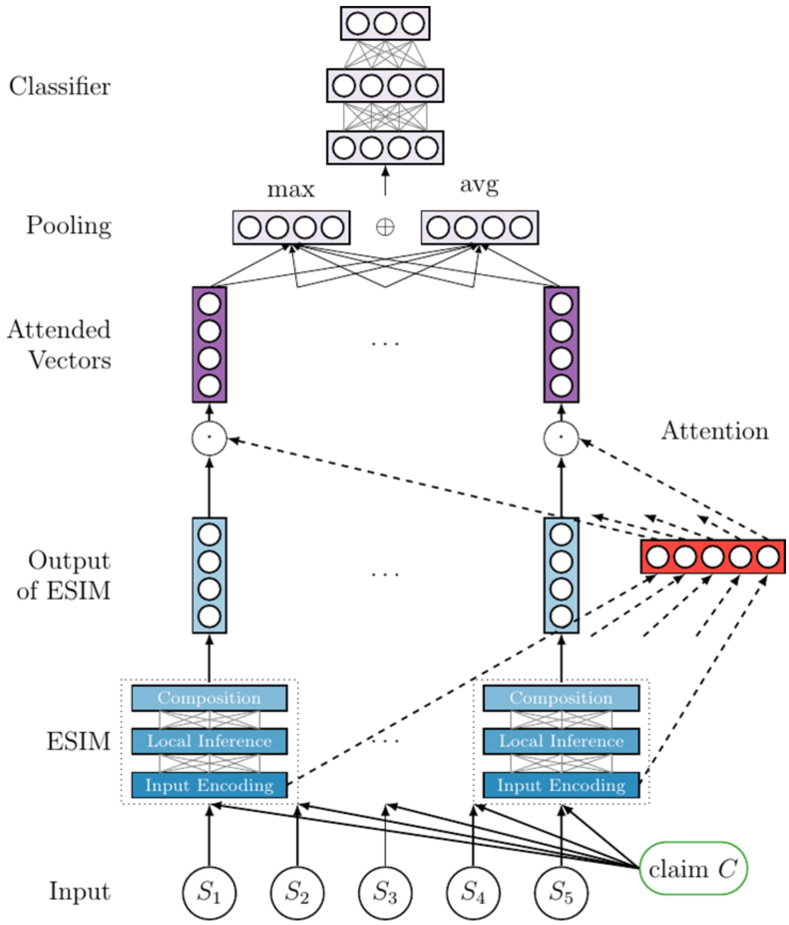


Figure 7.1: Extension of the ESIM for the FEVER RTE task

fed through a single layer perceptron giving rise to two vectors. The cosine similarity of the two vectors is then used as an attention weight.

The five output vectors of all ESIMs are multiplied with their respective attention weights and we apply average and max pooling on these vectors in order to reduce them to two representations. Finally, the two representations are concatenated and fed through a 3-layer perceptron for the prediction of the verdict (three way classification). The motivation behind the attention mechanism is that it allows us to extract information from the five sentences that is most relevant for the classification of the claim. In fact, since we are always considering five sentences but an evidence set in the FEVER corpus usually consists of only one or two sentences, the model needs to be able to focus on the most important information. Other teams participating in the FEVER shared task came up with similar sentence weighting approaches (Nie et al., 2019; Yin and Roth, 2018).

7.3.2 Experiments and results

As for the `rankingESIM` (Section 6.4.1), we explore the performance of our claim validation model for different numbers of evidence sentences. The results on the **FEVER18** development set illustrated in Table 7.1 demonstrate that our model performs best if all five highest ranked-sentences are used. In the table, *label accuracy* refers to the accuracy of our claim validation model. The *FEVER score* was the official metric for the evaluation of the systems in the FEVER shared task. It represents a restricted version of the label accuracy, where a claim is only considered as validated correctly, if the right verdict for the claim is predicted and the correct set of evidence for the claim is retrieved. It must be noted that the scores presented in the table are based on the experiments of the entire pipeline, that is, we do not use the gold evidence sentences for the verdict prediction, but sentences extracted by our sentence selection model from the documents that have been retrieved by our document retrieval system.

#sentence(s)	label accuracy (%)	FEVER score (%)
1	60.03	54.80
2	61.87	57.19
3	64.29	59.33
4	66.30	61.79
5	**68.49**	**64.74**

Table 7.1: Performance of our claim validation model using different numbers of sentences

In Table 7.2, we compare the performance of our three sub-systems developed for the FEVER shared task, as well as the full pipeline, to the baseline sub-systems and the entire pipeline implemented by the shared task organizers (Thorne et al., 2018a). It must be remarked that for sentence selection, we provide our model the documents from our document retrieval system, and the baseline sentence selection model receives the documents from the baseline document retrievals system. We use the same strategy to compute the label accuracy for the RTE task (the provided input sentences are from the respective sentence selection model). The results reported on the development set demonstrate that we were able to significantly improve upon the baseline in each sub-task. The performance gains over the entire pipeline add up to an improvement of about 100% with respect to the performance of the baseline pipeline.

In Table 7.3, the results of the top five systems (out of the 23 participating systems) of the FEVER shared task are reported on the held out test-set. As the results demonstrate, we reached the third rank. Since the three top systems have used similar network architectures (different versions of the ESIM attention mechanism), the results are not very much different. Team *Papelo* (Malon, 2018) used a Transformer network (Vaswani et al., 2017) and also reached good performance. With respect to the scores of the remaining systems, there is a substantial drop in performance, and the fifth-ranked team *SWEEPer* only reaches a score of 49.94%.

7.3.3 Error analysis

We have discovered that a large number of claims are misclassified due to the model's disability to interpret numerical values. For instance, the claim "*The heart beats at a resting rate close to 22 beats per minute.*" is not classified as *refuted* by the evidence sentence "*The heart beats at a resting rate close to 72 beats per minute.*". The only

Task (metric)	system	score (%)
Document retrieval (accuracy)	baseline (TF-IDF) our system (Section 4.3.2)	70.20 93.55
Sentence selection (recall @5)	baseline (TF-IDF) our system (Section 6.3.1)	44.22 87.10
Textual entailment (label accuracy)	baseline (DecAtt) our system (Section 7.3.1)	52.09 68.49
Full pipeline (FEVER score)	baseline our system	**32.27** **64.74**

Table 7.2: Comparison of the performance of our pipeline and the baseline pipeline (Thorne et al., 2018a) on the **FEVER18** development set

#	Team	label accuracy (%)	FEVER score (%)
1	UNC-NLP	**67.98**	**63.98**
2	UCL Machine Reading	67.44	62.34
3	TUDA UKP-Athene	65.22	61.32
4	Papelo	60.74	57.04
5	SWEEPer	59.64	49.86

Table 7.3: Top 5 systems of the FEVER shared task

information refuting the claim is the number, but we assume that neither GloVe nor FastText word vectors can embed numbers distinctly enough so that the model can identify the contradiction. Another problem is challenging *not enough info* cases. For instance, the claim "*Terry Crews played for the Los Angeles Chargers.*" (annotated as *not enough info*) is classified as *refuted*, given the sentence "*Crews played as a defensive end and linebacker in the National Football League (NFL) for the Los Angeles Rams, San Diego Chargers, and Washington Redskins, ...*". The sentence is related to the claim but does not exclude it, which makes this case difficult.

7.3.4 Conclusion

In order to achieve high performance in the FEVER RTE task, a developed pipeline needs to perform well across all the three sub-tasks. Without retrieving the correct set of documents, the required evidence sentences could not be identified, and consequently, no credit could be given for a validated claim even if the correct verdict was predicted. Obviously, the correct type of evidence also helps the RTE component to predict the correct verdict. The evaluation metric penalizes models that make the right prediction on the basis of wrong evidence, which also helps to prevent overfitting.

For the RTE sub-tasks, the top three teams converged to a similar network architecture that is based on the ESIM or on an ESIM like attention mechanism.

Even though good performance is reached by the top three systems, which almost achieve 70% accuracy for the three-way classification, it is expected that more recent model architectures based on language modeling (Devlin et al., 2018; Radford et al., 2019; Yang et al., 2019b) can further improve performance. As already shown by Yin and Roth (2018), further improvement gains can be achieved when considering sentence selection and RTE as a multitask problem.

The FEVER shared task has significantly progressed the work on automated fact-checking and inspired the development of pipeline systems in contrast to a single model fact-checking systems. Nevertheless, the FEVER fact-checking problem is to some extent simplified and still deviates from *real* fact-checking. The entire FEVER dataset was created syntactically only using Wikipedia as an information source. We address this issue in the following section by running experiments on our multi-domain corpus **Snopes19**, which is based on real fact-checking instances.

7.4 Experiments on Snopes19

We define the claim validation problem for the **Snopes19** corpus in such a way that we can compare it to the FEVER RTE problem setting discussed above. Thus, as illustrated in Table 7.4, we compress the different verdicts from the **Snopes19** corpus into three categories. In order to form the *not enough info* (NEI) class, we combine the claims with these three verdicts: *mixture*, *unproven*, *undetermined.* We have found that for these classes of claims, no clear decision can be made whether the claim is *true* or *false* even though Fine-Grained Evidence (FGE) is provided. Moreover, we entirely omit all the other instances with verdicts like *legend, outdated, miscaptioned, ...* since these cases are ambiguous and difficult to classify on the basis of the given FGE.

FEVER	Snopes
refuted:	false, mostly false
supported:	true, mostly true
NEI:	mixture, unproven, undetermined

Table 7.4: Compression of the Snopes verdicts to FEVER verdicts (NEI: not enough information)

7.4.1 Experiments

For the Snopes claim validation task, we consider models of different complexity:
`extendedESIM` is our extended version of the ESIM that we have developed for the FEVER RTE task (Section 7.3.1).
`BertSentEmb` is an MLP classifier that is based on BERT (Devlin et al., 2018) pre-trained sentence embeddings.[4] This model first derives representations of the claim and the FGE using BERT. The representations of the FGE are reduced to one vector

[4] `https://github.com/hanxiao/bert-as-service`

using an attention mechanism based on the claim (FGE more relevant to the claim is weighted higher). The compressed representation of the FGE is then concatenated with the representation of the claim and feed through a MLP for classification.

`DecAtt` is the Decomposed Attention model presented in Section 5.4.1. This model was used as a baseline in the FEVER shared task.

`BiLSTM` is a simple BiLSTM (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) architecture. For this model, individual FGE sentences and the claim are feed through a BiLSTM giving rise to one representation for each sentence. In order to obtain one representation for the FGE, average and max-pooling is applied to all representations of FGE computed by the BiLSTM. The resulting FGE representation is concatenated to the representation of the claim and feed through an MLP for classification.

`USE+MLP` is the Universal Sentence Encoder (USE) (presented in Section 5.4.1) combined with a MLP. The claim and the FGE combined into one long sentence are feed into the USE giving rise to two representations. These representations are concatenated and feed through a MLP.

`featureSVM` is a Support Vector Machine classifier based on bag-of-words, unigrams, and topic models.

`quantBaseline` is a simple quantitative baseline that weights the number of supporting ETSs versus the number refuting ETSs for a claim. If the number of refuting ETSs is larger than the number of supporting ETSs, the claim is classified as *refuted* and otherwise as *supported*. In the case of exact ties, it predicts *not enough info* (NEI).

The results illustrated in Table 7.5 show that `BertSentEmb`, `USE+MLP`, `BiLSTM`, and `extendedESIM` reach similar performance. The models `DecAtt` and `SVM` appear not to be suitable for the task and are even inferior to the quantitative baseline `quantBaseline`. `BertSentEmb` performs best, however, the score is still relatively low compared to the model's performance on FEVER RTE problem. This performance drop is discussed in the following section in more detail.

Labeling method	recall m	prec. m	F1 m
quantBaseline	0.415	0.377	0.395
random baseline	0.333	0.333	0.333
majority vote	0.198	0.170	0.249
BertSentEmb	0.477	**0.493**	**0.485**
USE+MLP	0.483	0.468	0.475
BiLSTM	0.456	0.473	0.464
extendedESIM	**0.561**	0.503	0.454
featureSVM	0.384	0.396	0.390
DecAtt	0.336	0.312	0.324

Table 7.5: Claim validation results (m = macro)

predicted \ gold	supported	refuted	NEI
supported	36	26	13
refuted	**38**	**203**	**53**
NEI	18	42	27

Table 7.6: Confusion matrix for claim validation `BertSentEmb`

7.4.2 Error analysis and further investigations

We performed an error analysis for `BertSentEmb` as the highest-scoring model. The confusion matrix for the predictions of the model is illustrated in Table 7.6. The class distribution of the verdicts for **Snopes19** is highly biased towards *refuted* (false) claims (see Table 3.7 in Section 3.2.2), and therefore, as the confusion matrix shows, claims are frequently labeled as *refuted* even though they belong to one of the other two classes. We observed that many of the FGE for claims, which have been wrongly classified as *refuted*, contain negating phrases, such as "*there was no*", "*instead of*", "*But there was*", "*has no*", "*must not*". We assume this was the main reason for misclassification. Our analysis has also shown that many of the instances are difficult to classify because they contain contradicting FGE. An example of such a case is given below (Example 7.2). Whereas the first two FGE sentences support the claim, the third FGE sentence contradicts it.

Claim: As a teenager, U.S. Secretary of State Colin Powell learned to speak Yiddish while working in a Jewish-owned baby equipment store.

Gold standard: *supported*; **Prediction**: *refuted*

FGE: *(1) As a boy whose friends and employers at the furniture store were Jewish, Powell picked up a smattering of Yiddish.*
(2) He kept working at Sickser's through his teens, earning 75 cents an hour and picking up a smattering of Yiddish.
(3) A spokesman for Mr. Powell said he hadn't heard about the spoof but confirmed that Gen. Powell does speak a little Yiddish.

Example 7.2: A misclassified instance due to contradicting ETS

The low performance of the model was reflected in the examples analyzed in the error analysis, as it was often difficult to determine, why the classifier misclassified a particular instance. As mentioned above, compared to the performance of the highest scoring models on the FEVER RTE task, which reach almost 0.7 accuracy, the performance of the best models on Snopes corpus is relatively low. The performance score reduces even further if we consider the claim validation problem not in isolation, but as a second step in the pipeline, that is, we first identify FGE using our `rankingBiLSTM` (Section 6.3.1) and then classify the claim on the basis of the collected FGE using `BertSentEmb`. In this case, the model reaches only 0.44 F1 macro.

verdict \ stance	support	refute	sum
supported	3.19	0.16	3.34
refuted	1.89	0.80	2.69
NEI	2.54	0.83	3.37

Table 7.7: The number of FGE (fine-grained evidence sentences) depending of the stance of FGE and the verdict of the claim

In order to determine the reason for the low performance, we have conducted a detailed investigation. We performed experiments with the models applied on **Snopes19** (listed in Table 7.5) on the FEVER RTE task using the ground truth evidence sentences. To eliminate the difference in size of the two corpora, we have reduced the number of training instances in the **FEVER18** corpus to the number of training instances in the **Snopes19** corpus and re-ran the experiments. Nevertheless, the best performing model `BertSentEmb`, could still reach a score of 0.54 F1 macro on the **FEVER18** corpus which is significantly higher than the performance of the model on the **Snopes19** corpus.

After eliminating the difference in size, we have analyzed the training instances of the two corpora more closely. Based on our analysis, we came to the conclusion that there are two main reasons for the performance gap:

(1) The **Snopes19** corpus is heterogeneous and thus, it is more challenging for a machine learning model to generalize across different text styles. In fact, we have performed additional experiments, in which we pre-trained models on the FEVER corpus and fine-tuned the parameters of the models on our Snopes and vice versa. In both experiments, no significant performance gains over the original training setup with only one corpus could be achieved. Models pre-trained on FEVER are probably tailored to the Wikipedia text and are not able to generalize to diverse forms of text in the Snopes corpus. The reverse experiment is probably not successful because the *simplified* form of the FEVER claim validation problem differs too much from the general claim validation setting defined for the **Snopes19** corpus. E.g. as discussed in Section 3.2.2, the structure of claims in **FEVER18** is much simpler compared to the claims in **Snopes19**.

(2) Another reason for the low performance on **Snopes19** is that it is difficult to infer verdict from the given FGE. Although the corpus is biased towards false claims, there is a large number of ETSs that support these false claims (see Table 3.8). As discussed in Section 3.2.2, this is because many of the retrieved ETSs originate from false news websites. Thus, in many cases, we have contradicting evidence, i.e. some of the evidence sentences support the claim and others refute it. In order to analyze this issue in more detail, we computed the distribution of the FGE (the number of the fine-grained evidence sentences) depending on the stance of the FGE and the verdict of the claim. The resulting matrix illustrated in Table 7.7 shows that even though a high number of supporting FGE is a strong indicator for a *supported* claim (3.19 vs. 0.16), more FGE are supporting *refuted* claims than refuting FGE (1.89 vs. 0.80). This further confirms our observation from Section 3.2.2, where we showed that the stance of the evidence is not a good indicator of the verdict of a claim.

7.4.3 Conclusion

As the results of our analyses in the previous sub-section have shown, our claim validation problem defined for the **Snopes19** corpus is more challenging than the FEVER shared task RTE problem. For the FEVER RTE problem, systems basically only need to classify the stance of the evidence with respect to the claim to predict the verdict (Section 3.2.2). For **Snopes19**, the relation between the evidence and the claims is more complicated. The stance in combination with information about the content of the FGE is potentially useful for the prediction of the verdict, but, as our correlation analysis in Section 3.2.2 shows, not in isolation.

Moreover, we believe that the Snopes problem setting is not only more challenging but also more realistic. Since the Snopes corpus is based on real fact-checking instances from the Snopes website, it is more similar to the task that fact-checkers encounter in practice. Thus, in order to make progress on the fact-checking problem, systems should be developed that reach high performance not on synthetic datasets but on real fact-checking instances. Since present approaches reach only low performance on such instances (see Table 7.6), alternative machine learning methods should be explored.

Based on our analysis, we conclude that a system is not able to validate claims in realistic fact-checking instances only on the basis of a small number of evidence sentences, and that additional contextual information is required. Thus, in future work, not only the FGE need to be considered for the validation of the claim, but also other additional information available in the **Snopes19** corpus, such as the stance of the FGE, FGE sources, and documents from the Snopes website that provide additional information about the claim. Here, the work done by Popat et al. (2017, 2018) can serve as a starting point, as they already performed claim validation experiments in which they considered the sources and the stance of the evidence. Moreover, as the Example 7.1 at the beginning of the chapter demonstrates, we need a system that explores the relation between the evidence. In particular, evidence sentences can not only refute the claim but also other evidence sentences. Thus, cross-references between the evidence should be taken into account in order to find a sub-set of evidence that forms a consistent narrative supporting or refuting the claim. This extended approach is expected to lead to better performance as much more information would be considered in the claim validation process as it is available in a small set of evidence sentences.

7.5 Chapter summary

In this chapter, we analyzed the claim validation problem which is the task of predicting the verdict for a claim on the basis of a number of evidence sentences. We presented a model that we have developed for the FEVER RTE sub-task (which corresponds to our definition of the claim validation problem). The model reached the third rank in the FEVER shared task among 23 competing systems. In a number of experiments, we analyzed the performance of the model in isolation and as a part of our fact-checking pipeline. In order to identify the weaknesses of our model that can be addressed in future work, we have performed an error analysis and identified challenging instances in which the model fails. We have found that the model fails if it needs to determine the verdict of the claim on the basis of numerical values or if world knowledge is required. Moreover, we have discussed the results of the FEVER shared task and in particular the outcome for the RTE sub-task. We concluded that even though relatively high scores for the task could be achieved, modern pretrained language models could potentially further increase performance on the task. Nevertheless, we have also remarked that the FEVER shared task problem setting is not realistic enough, as it is based on synthetically generated fact-checking instances that have been created only on the basis of the information in Wikipedia. In the second part of the chapter, we therefore analyzed the more realistic claim validation problem defined by the Snopes corpus. We have performed a large number of experiments on the corpus using models of different levels of complexity. We have found that compared to the results on the FEVER corpus, the models achieve a relatively low performance. In order to identify the reason for the lower scores, we have performed an error analysis for the best scoring model based on BERT embeddings, and conducted additional experiments. The error analysis has shown that the model is biased towards the majority class of the Snopes corpus and that the errors are very diverse. Our additional experiments have shown that the low performance of the models can only in part be attributed to the difference in size of the two corpora. A more significant factor is that the Snopes corpus is based on heterogeneous web documents with diverse text styles. We assume that the models struggle to generalize across the different domains of the corpus. Another reason for the low performance is that the evidence for a claim in the Snopes corpus is often contradictory, that is, while some of the evidence sentences support the claim, others refute it. In fact, since many of the evidence come from unreliable sources, the stance of the evidence is not necessarily indicative of the verdict for a claim, as for the FEVER corpus. We therefore concluded that in order to make progress on real fact-checking instances, not just the content and the stance of the evidence sentences need to be taken into account, but also the sources of the evidence and other contextual information available in the Snopes corpus.

Chapter 8

Conclusions and outlook

In this thesis, we explored the problem of automating the fact-checking process using machine learning techniques in order to help to control the growing amount of false information on the web. We proposed a pipeline approach for the validation of claims on the basis of evidence sentences. The pipeline consists of sub-systems for the following tasks: *document retrieval*, *stance detection*, *evidence extraction*, and *claim validation*. Because we believe that fact-checking is an AI-complete problem, meaning that human-level intelligence is required to solve the task, our pipeline approach is designed to assist fact-checkers in the fact-checking process rather than taking over the task entirely. The proposed pipeline is able to process raw text and thus make use of the vast amount of information contained in web documents. It is therefore superior to fact-checking methods based on knowledge bases that have a relatively low coverage. At the same time, our approach is more transparent than current end-to-end trained machine learning systems, as the fact-checker can observe the outputs of the intermediate sub-systems and thus trace back potential errors. The developed sub-systems of the pipeline have been tested in two competitive shared tasks, namely the Fake News Challenge and the FEVER shared task, and secured top three positions in these competitions.

In this thesis, we analyzed the sub-tasks of the fact-checking process in individual chapters and introduced a new, richly annotated corpus for training machine learning models for the sub-tasks. Below, we summarize the most important findings and contributions of this thesis and answer the research questions defined at the beginning of the thesis. We also discuss the impact of our work on the fact-checking community and highlight promising future research directions.

8.1 Summary of contributions and findings

Chapter 3. We introduced a new, richly annotated corpus for training machine learning systems for the four sub-tasks in the fact-checking process: *document retrieval, stance detection, evidence extraction,* and *claim validation.* The corpus is based on the Snopes website and provides annotations from real fact-checking instances. It is therefore superior to other *synthetic* datasets, such as the FEVER shared task dataset, which is only based on Wikipedia. We presented our corpus construction and annotation framework that allows for the efficient development of large corpora with high inter-annotator agreement. We also provided a detailed analysis of the corpus, discussed the dataset statistics, and described the biases of the corpus. On the basis of our analysis, we found that many evidence sentences come from unreliable sources, which makes it difficult to validate many of the claims in the corpus. Using a correlation analysis, we discovered that there is only a weak correlation between the stance of the evidence and the verdict for the claims. This means that the claim validation problem cannot be solved only using the information about the stance of the evidence. Our analyses showed that although the fact-checking problem defined by our corpus is more realistic, it is more challenging than the fact-checking problem posed by other available corpora such as the FEVER shared task corpus.

Chapter 4. We proposed a document retrieval system based on entity linking and hand-crafted rules that reaches high performance in the FEVER document retrieval problem setting. The system is based on our analysis of the FEVER dataset, wherein we discovered that the entities in the claims often correspond to titles of Wikipedia articles. Thus, it was possible to identify Wikipedia articles by matching the entity mentions in the claim to Wikipedia article titles. Our entity linking approach substantially outperformed traditional information retrieval systems based on the inverted index and TF-IDF ranking.

Chapter 5. In order to tackle document-level stance detection, we introduced a feature-based deep Multi-Layer Perceptron (MLP). The model was deployed in the Fake News Challenge (FNC) reaching the second rank out of 50 competing systems according to the official FNC metric. In fact, we found that the feature-based MLP is superior to LSTM-based deep neural networks that use word embeddings as features. In a retrospective analysis of the FNC stance detection task, we evaluated the FNC problem setting and discovered that the official FNC metric is biased towards the majority class and can therefore be exploited to maximize the score. F1 macro, on the contrary, is a more reliable indicator of the robustness of a system, as it penalizes systems that do not perform well across all classes. We therefore proposed the F1 macro as a metric for the evaluation of systems on the FNC corpus. We have shown that because our feature-based MLP performs well across all classes, according to F1 macro, our model is superior to other systems on the FNC corpus. In this chapter, we also performed stance detection experiments on our Snopes corpus and found that in this case, feature-based classifiers outperform LSTM-based neural networks using word embeddings.

Chapter 6. We introduced an evidence extraction system that is able to identify sentence-level evidence for a given claim in retrieved documents. The system is based on the ESIM (Chen et al., 2017b), which is a powerful encoder developed for the SNLI task (Bowman et al., 2015). We modified ESIM so that sentences from the retrieved documents can be ranked according to their relevance for the validation of the claim. After ranking, we took the top-k sentences as evidence. The evidence ranking system together with our document retrieval system was evaluated in the FEVER evidence selection sub-task. We were able to beat all 23 competing systems and win the shared task in this category.

Chapter 7. To address the claim validation problem, we again developed a new model on the basis of ESIM. We implemented an extended version of ESIM that allows us to determine the verdict for a claim on the basis of an arbitrary number of evidence sentences. The system was evaluated in the FEVER claim validation sub-task reaching the third rank. Our combined pipeline consisting of our systems for document retrieval, evidence extraction, and claim validation secured the third rank out of 23 other pipelines participating in the FEVER shared task. We applied our extended ESIM and other promising claim validation models to our newly constructed Snopes corpus and found that these models show substantially lower performance than their performance on the FEVER shared task corpus. Our subsequent analysis showed that the performance gap can be attributed to two major factors:
(1) The Snopes corpus is based on heterogeneous web sources. The text styles in the corpus are therefore very diverse, ranging from informal language in discussion forums to carefully written news articles of established news magazines. Because a system trained on the corpus needs to generalize across all different text styles, the performance is reduced.
(2) The evidence for a claim in the Snopes corpus is often contradictory, that is, whereas some of the evidence sentences support the claim, others refute it. In fact, we found that the correlation between the stance of the evidence and the verdict of the claim is very low. Thus, the verdict for a claim cannot be simply deduced from the stance of the evidence sentences as for the FEVER corpus. It is therefore often difficult even for humans to validate a claim only on the basis of the given evidence sentences.
Based on our analysis, we concluded that the more realistic fact-checking problem defined by our Snopes corpus is very challenging. Even though we can reach reasonable performance for some of the fact-checking sub-tasks, the problem as a whole, and particularly the claim validation problem, is far from being solved. Thus, other more elaborated methods are required in order to make progress on this task in future work.

Research questions. Based on our findings, we can now answer the research question posed in Chapter 2 of this thesis.

- Is it possible to design an annotation framework for the annotation of evidence in documents for a given claim and the annotation of the stance of the evidence with respect to the claim that leads to high inter-annotator agreement?
  ⇒ For the annotation of the stance of the Evidence Text Snippets (ETSs), we have been able to reach an inter-annotator agreement of 0.7 Cohen's Kappa which is considered to be *substantial.* For the annotation of Fine-Grained Evidence (FGE) (evidence on the sentence-level), we have been able to achieve a *moderate* inter-annotator agreement of 0.55 Cohen's Kappa. In fact, the annotation of sentences in documents is challenging, and related work often reports lower agreement scores. We therefore conclude that the annotation of the evidence and stance of the evidence was successful. Nevertheless, we acknowledge that further research is required in order to achieve higher agreement, in particular for evidence annotation.

- If we have stance annotated evidence for a claim (in the form of text snippets that support or refute the claim), is it possible to validate this claim only on the basis of the number of supporting and refuting ETSs without considering the textual content of the claim and the evidence?
  ⇒ As our correlation analysis in Section 3.2.2 demonstrates, for a realistic fact-checking corpus with evidence from diverse sources, the correlation between the veracity of the claim and the stance of the evidence is low. It follows that the stance of the evidence is only weakly indicative of the verdict of the claim. Therefore, we conclude that the stance information is not sufficient for the validation of the claim and that additional information is required.

- How well can current machine learning models for fact-checking, which perform well on existing datasets covering a single domain, generalize to multi-domain datasets, i.e. can we achieve a high performance for automated fact-checking on a multi-domain dataset, if the system is only training on a single-domain corpus?
  ⇒ Our analysis in Section 7.4.3 shows that the models that perform on a single-domain dataset, such as the FEVER corpus, do not reach high performance on heterogeneous multi-domain corpora such as our Snopes corpus. Cross-domain experiments have also shown that pre-training a model on a single-domain corpus and then fine-tuning the model on a multi-domain corpus is not helpful because no performance improvements could be observed. We therefore conclude that large heterogeneous corpora are required in order to train models that are able to generalize across different domains and to validate claims on the basis of evidence with diverse text styles.

- In most of the fact-checking problem settings defined so far, evidence for the validation of a claim is only provided in the form of one or several sentences. Is this information sufficient, or do we need additional contextual information in order to reach high performance on the claim validation task?
  ⇒ Our experiments in Section 7.4 demonstrate that the performance of the claim validation models, which only take the claim and a number of evidence sentences as input, is relatively low on a realistic multi-domain corpus. Our subsequent analysis has shown that the evidence sentences originate not only from reliable but also from unreliable sources. As a result, the evidence sentences for a claim are often contradicting each other. Thus, we conclude that in order to reach higher performance for claim validation, the information about the source of the evidence is required and the relationship between the evidence sentences needs to be taken into account.

8.2 Impact of the contributions

With our work, we have been able to significantly contribute to the fact-checking community. The repository of our stance detection sub-system evaluated in the Fake News Challenge was forked more than 110 times.[1] Our pipeline developed for

[1] `https://github.com/hanselowski/athene_system`

the FEVER shared task serves as an official baseline for the FEVER shared task.[2] Fact-checkers from FullFact[3] and Factmata[4] have requested the code of the pipeline in order to apply the system to real fact-checking instances.

Our publications are frequently cited in the context of automated fact-checking and in particular, the two studies (Hanselowski et al., 2018a,b) received much attention in the literature. Both works are frequently cited as related work, but more importantly, our systems were successfully combined with other models or reached a new state-of-the-art on a new corpus. Below we present some of these studies.

Follow-up work based on (Hanselowski et al., 2018a):
Our feature-based MLP for the FNC task was shown to reach best results on a new Arabic document-level stance detection corpus if the model is given the entire document and the most important passages for the classification in the document are highlighted (Baly et al., 2018). The model was able to outperform other top-ranking systems from the FNC, as well as a Memory Network.

Our stacked LSTM developed for the FNC was shown to reach the highest performance on article-claim stance classification for a new Persian corpus, thereby outperforming other feature-based classifiers (Zarharan et al., 2019).

Following our criticisms of the FNC metric, a number of studies acknowledged that the F1 macro is more appropriate for the evaluation of systems on the FNC corpus than the FNC metric (Conforti et al., 2018; Jwa et al., 2019; Chernyavskiy and Ilvovsky, 2019).

Follow-up work based on (Hanselowski et al., 2018b):
Because our document retrieval and evidence ranking systems reached the best results in the FEVER shared task, they have been frequently used by other researchers. The systems were used either to retrieve evidence for a new claim validation model, or as baselines for the development of novel document retrieval and evidence ranking approaches (Soleimani et al., 2019; Zhou et al., 2019; Liu et al., 2019b; Chernyavskiy and Ilvovsky, 2019).

8.3 Future research directions

Below we discuss a number of promising research directions that can be followed in order to improve the performance of the individual fact-checking sub-systems or the pipeline as a whole. Moreover, we discuss a number of popular recent machine learning approaches that can also be leveraged for automated fact-checking.

Improvement of the claim validation sub-system. Claim validation is at the core of the fact-checking process; however, it is also the most challenging sub-task. To improve the claim validation sub-system proposed in this thesis, several changes to the system can be made. The information about the stance and the source of the evidence can be taken as additional input to the system, which could help to

[2] `http://fever.ai/task.html`
[3] `https://fullfact.org/`
[4] `https://factmata.com/`

achieve further performance improvement. Relations and cross-references between the evidence could be explicitly modeled, which, potentially would also lead to better performance. In fact, in some fact-checking instances, a number of evidence sentences need to be combined to validate a claim (see, for instance, multi-hop reasoning for question answering (Yang et al., 2018) or solving syllogisms in logical argumentation). Moreover, in order to estimate how reliable the collected evidence sentences are, in addition to the information about the source of the evidence sentences, a second evidence extraction system can be developed that tries to find supporting evidence for a given evidence sentence. All these measures would potentially enable the system to find a consistent and reliable set of evidences that either support or refute a given claim.

Self-supervised models for automated fact-checking. Self-supervised approaches such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), XLM (Conneau and Lample, 2019), or XLNet (Yang et al., 2019b) have recently shown that substantial performance improvements across a large number of tasks can be achieved, if a model is pretrained in a self-supervised manner on a massive dataset and then fine-tuned on a small corpus for a particular *target* task. So far, we have only explored how a multi-layer perceptron, which is based on pretrained representations of BERT, performs for claim validation without fine-tuning BERT on the fact-checking corpora. Some further work was done by Yang et al. (2019a), who explored the GPT model (Radford et al., 2018) for rumor evaluation, or by Ning et al. (2019), who applied BERT to hyper-partisan news detection. More experiments are required in order to find out which of the pretrained models is best suited for the different fact-checking sub-tasks. Moreover, an entirely new self-supervised technique could be developed that is particularly suitable for automated fact-checking. Current self-supervised models are based on different training objectives, such as predicting the next word (Peters et al., 2018; Radford et al., 2018, 2019), next sentence prediction (Devlin et al., 2018), predicting masked words (Devlin et al., 2018), or word permutations (Yang et al., 2019b). One training objective (or *auxiliary*) task often benefits a particular *target* task that is framed in a similar manner. It would therefore also be worth exploring whether auxiliary tasks for the different fact-checking sub-tasks can be found that could be used to pretrain models for these tasks.

Multitask learning for automated fact-checking. Even though we have considered the four fact-checking sub-tasks independently of each other, there are synergies between the tasks that can be exploited if a joint model is trained in a multitask setting. A model simultaneously trained to select evidence sentences, determine their stance with respect to the claim, and predict the verdict for the claim, could benefit from the supervision at different levels and thus, achieve higher performance. In fact, work done by Popat et al. (2017); Yin and Roth (2018); Li et al. (2019) already suggests that multitask learning for automated fact-checking can be beneficial. Self-supervised pretrained models can be fine-tuned in a multitask setting, yielding further performance improvements (see, for instance, MT-DNN (Liu et al., 2019a) and ERNIE 2.0[5]). This framework can also be explored for automated fact-checking,

[5] http://research.baidu.com/Blog/index-view?id=121

where a model is first pretrained on an auxiliary task and then simultaneously fine-tuned on several fact-checking sub-tasks.

Multimodal automated fact-checking. In this thesis, we have restricted our scope to textual data only, that is, we have only considered the case where the claims and the evidence are presented as text. Nevertheless, evidence or misleading content often comes in the form of an image (fauxtography), a video (deep fakes), an audio signal (fake speech), or multiple modalities at once. It would therefore be promising to develop a system that can process information from different modalities and validate published content based on all signals in combination. For this purpose, recently introduced multi-modal self-supervised systems proposed by Sun et al. (2019) and Lu et al. (2019) could be explored.

Detecting automatically generated fake textual content. Recent progress in self-supervised learning led to significant progress in natural language generation. Deep neural networks pretrained in a self-supervised manner can enable bad actors to generate fake textual content in large quantities. Only by providing a headline as prompt, a pretrained generative model is able to generate a coherent piece of text (Radford et al., 2019). However, as shown by Zellers et al. (2019), pretrained models can be used not only to generate fake content but are also able to reliably detect artificially generated text. These are the first promising results indicating that the flood of false information generated by pretrained generative models can be controlled. Nevertheless, additional follow-up research is required. Investigation is needed on, whether artificially generated texts produced by different types of pretrained generative model can be reliably identified by a single pretrained *detector* model or whether a collection of pretrained *detector* models is required.

Appendix A

Appendix

A.1 FNC features: detailed description

BoW/BoC features We use bag-of-words (BoW) 1- and 2-grams with 5,000 tokens vocabulary for the headline as well as the document. For the BoW feature, based on a technique by Das and Chen (2007), we add a negation tag "_NEG" as prefix to every word between special negation keywords (e.g. "not", "never", "no") until the next punctuation mark appears. For the bag-of-characters (BoC) 3-grams are chosen with 5,000 tokens vocabulary, too. For the BoW/BoC feature we use the TF to extract the vocabulary and to build the feature vectors of headline and document. The resulting TF vectors of headline and document get concatenated afterwards. Feature *co-occurrence* (FNC-1 baseline feature) counts how many times word 1-/2-/4-grams, character 2-/4-/8-/16-grams, and stop words of the headline appear in the first 100, first 255 characters of the document, and how often they appear in the document overall.

Topic models We use non-negative matrix factorization (NMF) (Lin, 2007), latent semantic indexing (LSI) (Deerwester et al., 1990), and latent Dirichlet allocation (LDA) (Blei et al., 2001) to create topic models out of which we create independent features. For each topic model, we extract 300 topics out of the headline and document texts. Afterwards, we compute the similarity of headlines and bodies to the found topics separately and either concatenate the feature vectors (NMF, LSI) or calculate the cosine distance between them as a single valued feature (NMF, LDA).

Lexicon-based features These features are based on the NRC Hashtag Sentiment and Sentiment140 lexicon (Kiritchenko et al., 2014; Mohammad et al., 2013; Zhu et al., 2014), as well as for the MPQA lexicon (Wilson et al., 2005) and MaxDiff Twitter lexicon (Rosenthal et al., 2015; Kiritchenko et al., 2014). All named lexicons hold values that signal the sentiment/polarity for each word. The features are computed separately for headline and document, and constructed as proposed by Mohammad et al. (2013): First, we count how many words with positive, negative, and without polarity are found in the text. Two features sum up the positive and negative polarity values of the words in the texts and another two features are set by finding the word with the maximum positive and negative polarity value in the text. Finally, the last word in the text with negative or positive polarity is taken as a feature. Since the MaxDiff Twitter lexicon also contains 2-grams, we decide to take them into account as well, whereas for the other lexicons only 1-grams incorporated. Additionally, we base features on the EmoLex lexicon (Mohammad and Turney, 2010, 2013). For all its words, it holds up to eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, disgust), based on the context they frequently appear in. For headline and document respectively, the emotions for all words are counted as a feature vector. The resulting vectors for headline and document are then concatenated. Lastly, the baseline features *polarity words* and *refuting words* are added. The first one counts refuting words (e.g. "fake", "hoax"), divides the sum by two, and takes the remainder as a feature signaling the polarity of headline or document. The latter one sets a binary feature for each refuting word (e.g. "fraud", "deny") appearing in the headline or document.

Readability features We measure the readability of headline and document with SMOG grade (only document), Flesch-Kincaid grade level, Flesch reading ease, and Gunning fog index (Štajner et al., 2012), Coleman-Liau index (Mari and Ta Lin, 1975), automated readability index (Senter and Smith, 1967), LIX and RIX (Jonathan, 1983), McAlpine EFLAW Readability Score (McAlpine, 1997), Strain Index (Solomon, 2006). The SMOG grade is only valid if a text has at least 30 sentences, and thus is only implemented for the bodies.

Lexical features As lexical features we implement the type-token-ratio (TTR) and the measure of textual lexical diversity (MTLD) (McCarthy, 2005) for the document, and only type-token-ratio for the headline, since MTLD needs at least 50 tokens to be valid. Also, the baseline feature *word overlap* belongs to this group. It divides the cardinality of the intersection of unique words in headline and document by the cardinality of the union of unique words in headline and document.

POS features The POS features amongst others include counters for nouns, personal pronouns, verbs and verbs in past tense, adverbs, nouns and proper nouns, cardinal numbers, punctuations, the ratio of quoted words, and also the frequency of the three least common words in the text. The headline feature also contains a value for the percentage of stop words and the number of verb phrases, which showed good results in the work of Horne and Adali (2017). For the *word-similarity* feature, which are mainly based on Ferreira and Vlachos (2016) we calculated average word embeddings (pre-trained word2vec model[1]) for all verbs (retrieved with Stanford Core NLP toolkit[2]) of headline/document separately. The cosine similarity between the averaged embeddings of headline and document is taken as a feature, as well as the hungarian distance between verbs of headline and document based on the paraphrase database[3]. The same computation is done for all nouns of headline and document. Additionally the average sentiment of the headline and the average sentiment of

[1] https://code.google.com/archive/p/word2vec/

[2] https://stanfordnlp.github.io/CoreNLP/

[3] http://www.cis.upenn.edu/ ccb/ppdb/

the document is used as a feature. A count of negating words of the headline and the document is added to the feature vector as well as the distance from the negated word to the root of the sentence. The number of average words per sentence of headline and document is another feature. The aforementioned features are improved by only selecting a predefined number of sentences of document and headline. Therefore the sentences are ordered by TF-IDF score.

Structural features The structural features contain the average word length of the headline and document, and the number of paragraphs and average paragraph length of the document.

A.2 Snopes annotation interface and annotation guidelines

A.2.1 Annotation guidelines for the annotation of the stance of the ETSs

Task: *You are given claims and articles (ETSs). Your task is to determine whether the articles are* ***supporting*** *or* ***refuting*** *the claims. If an article is off-topic or does not explicitly express a stance towards the claim, the third option must be selected:* ***No explicit reference to the claim***

Further remarks: *The stance must be determined only on the basis of the given article (ETS), that is, don't make the decision on the basis of your own knowledge about the claim. Also note, many articles are about the topic of the claim but do not express a stance. Thus, select the option* ***No explicit reference to the claim*** *in these cases.*

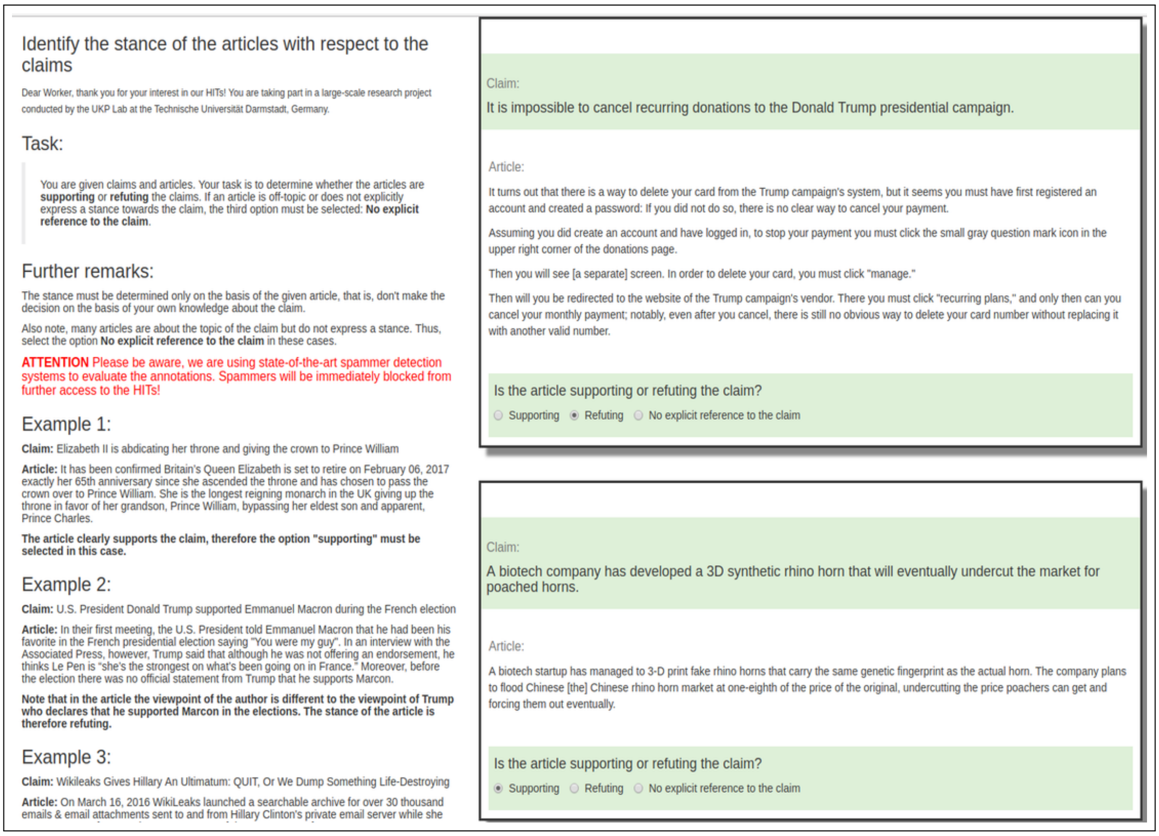
Pictures of the annotation interface are given in Figures A.1 and A.2. Whereas the first picture illustrates the entire annotation interface, the second displays a magnified instance for the annotation. As the figures show, in addition to the annotation guidelines, we provide a number of examples in order to help annotators to understand the task.

As the above annotation guidelines indicate, the stance of the ETS (article) towards the claim needs to be labeled as *agree*, *disagree*, and in case the ETS is off-topic or does not explicitly express a stance towards the claim, a third label is given (*No explicit reference to the claim*). For the sake of simplicity, we refer to this label as *no stance* henceforth.

A.2.2 Annotation guidelines for the annotation of FGE in the ETSs

Find supporting or refuting sentences for a claim:

- *You are given a claim and an article (ETS), and you are asked to annotate* ***supporting*** *or* ***refuting*** *sentences (FGE) in the article (ETS).*



Identify the stance of the articles with respect to the claims

Dear Worker, thank you for your interest in our HITs! You are taking part in a large-scale research project conducted by the UKP Lab at the Technische Universität Darmstadt, Germany.

Task:

> You are given claims and articles. Your task is to determine whether the articles are **supporting** or **refuting** the claims. If an article is off-topic or does not explicitly express a stance towards the claim, the third option must be selected: **No explicit reference to the claim**.

Further remarks:

The stance must be determined only on the basis of the given article, that is, don't make the decision on the basis of your own knowledge about the claim.

Also note, many articles are about the topic of the claim but do not express a stance. Thus, select the option **No explicit reference to the claim** in these cases.

ATTENTION Please be aware, we are using state-of-the-art spammer detection systems to evaluate the annotations. Spammers will be immediately blocked from further access to the HITs!

Example 1:

Claim: Elizabeth II is abdicating her throne and giving the crown to Prince William

Article: It has been confirmed Britain's Queen Elizabeth is set to retire on February 06, 2017 exactly her 65th anniversary since she ascended the throne and has chosen to pass the crown over to Prince William. She is the longest reigning monarch in the UK giving up the throne in favor of her grandson, Prince William, bypassing her eldest son and apparent, Prince Charles.

The article clearly supports the claim, therefore the option "supporting" must be selected in this case.

Example 2:

Claim: U.S. President Donald Trump supported Emmanuel Macron during the French election

Article: In their first meeting, the U.S. President told Emmanuel Macron that he had been his favorite in the French presidential election saying "You were my guy". In an interview with the Associated Press, however, Trump said that although he was not offering an endorsement, he thinks Le Pen is "she's the strongest on what's been going on in France." Moreover, before the election there was no official statement from Trump that he supports Marcon.

Note that in the article the viewpoint of the author is different to the viewpoint of Trump who declares that he supported Marcon in the elections. The stance of the article is therefore refuting.

Example 3:

Claim: Wikileaks Gives Hillary An Ultimatum: QUIT, Or We Dump Something Life-Destroying

Article: On March 16, 2016 WikiLeaks launched a searchable archive for over 30 thousand emails & email attachments sent to and from Hillary Clinton's private email server while she

Claim:

It is impossible to cancel recurring donations to the Donald Trump presidential campaign.

Article:

It turns out that there is a way to delete your card from the Trump campaign's system, but it seems you must have first registered an account and created a password: If you did not do so, there is no clear way to cancel your payment.

Assuming you did create an account and have logged in, to stop your payment you must click the small gray question mark icon in the upper right corner of the donations page.

Then you will see [a separate] screen. In order to delete your card, you must click "manage."

Then will you be redirected to the website of the Trump campaign's vendor. There you must click "recurring plans," and only then can you cancel your monthly payment; notably, even after you cancel, there is still no obvious way to delete your card number without replacing it with another valid number.

Is the article supporting or refuting the claim?

○ Supporting ● Refuting ○ No explicit reference to the claim

Claim:

A biotech company has developed a 3D synthetic rhino horn that will eventually undercut the market for poached horns.

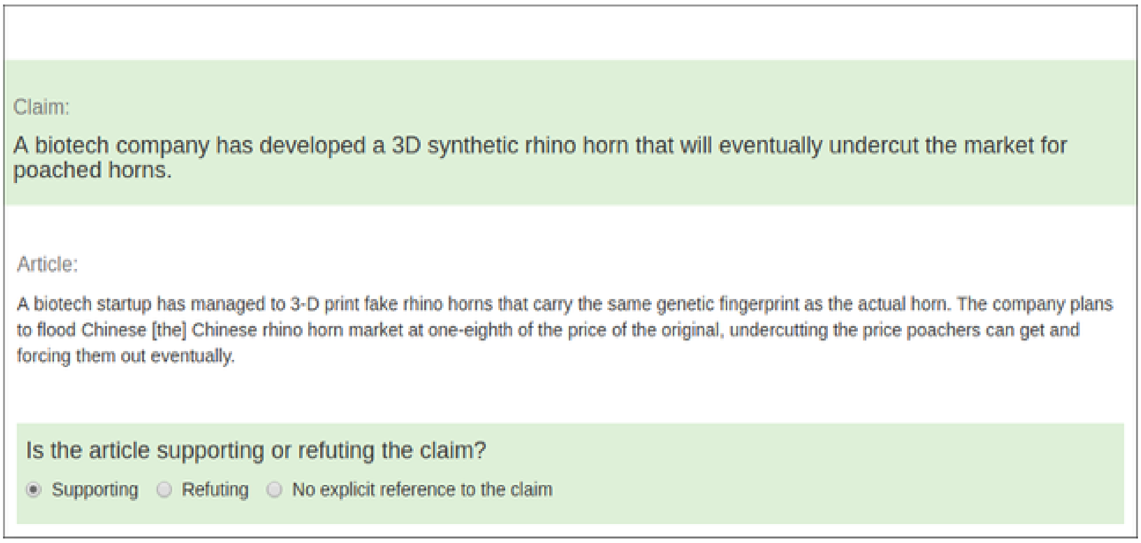
Article:

A biotech startup has managed to 3-D print fake rhino horns that carry the same genetic fingerprint as the actual horn. The company plans to flood Chinese [the] Chinese rhino horn market at one-eighth of the price of the original, undercutting the price poachers can get and forcing them out eventually.

Is the article supporting or refuting the claim?

● Supporting ○ Refuting ○ No explicit reference to the claim

Figure A.1: Annotation interface for the annotation of the stance of the ETSs (entire annotation interface)



Claim:

A biotech company has developed a 3D synthetic rhino horn that will eventually undercut the market for poached horns.

Article:

A biotech startup has managed to 3-D print fake rhino horns that carry the same genetic fingerprint as the actual horn. The company plans to flood Chinese [the] Chinese rhino horn market at one-eighth of the price of the original, undercutting the price poachers can get and forcing them out eventually.

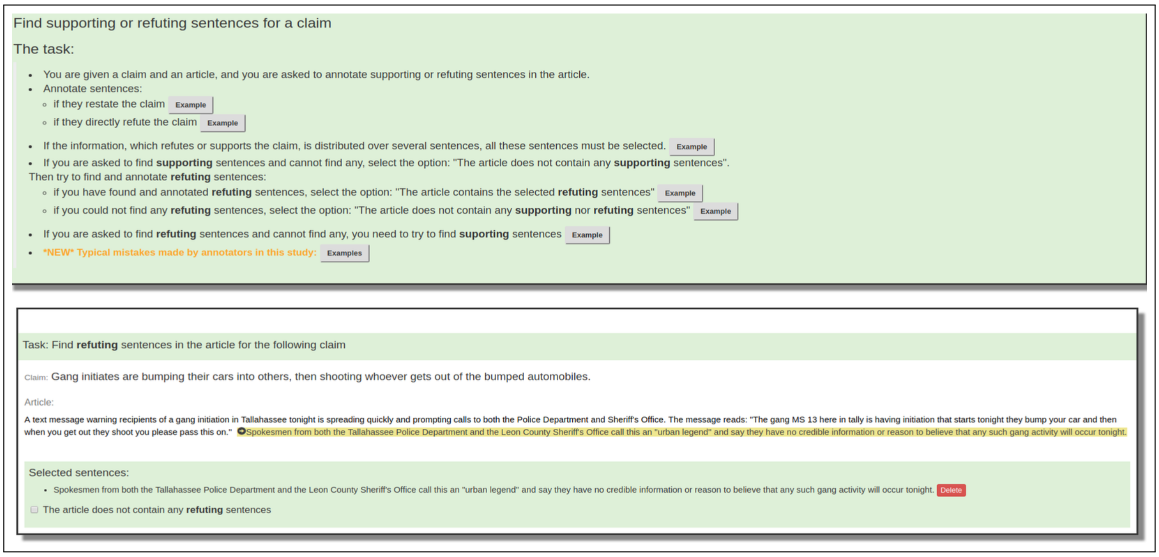
Is the article supporting or refuting the claim?

● Supporting ○ Refuting ○ No explicit reference to the claim

Figure A.2: Annotation interface for the annotation of the stance of the ETSs (a magnified instance)

- *Annotate sentences:*
  - *if they restate (support) the claim*
  - *if they directly refute the claim*
- *If the information, which refutes or supports the claim, is distributed over several sentences, all these sentences must be selected.*
- *If you are asked to find supporting sentences and cannot find any, select the option: "The article does not contain any supporting sentences."*
  Then try to find and annotate refuting sentences:
  - *if you have found and annotated refuting sentences, select the option: "The article contains the selected refuting sentences"*
  - *if you could not find any refuting sentences, select the option: "The article does not contain any supporting nor refuting sentences"*
- *If you are asked to find refuting sentences and cannot find any, you need to try to find supporting sentences.*

As the last two bullet points of the annotation guidelines describe, the annotation interface allows users to correct a wrong stance label, which was given in the first step of the annotation, and then select sentences with the corrected stance. Even though the annotation of the stance is of good quality and a wrong stance was assigned relatively rarely (see Section 3.2.2), we wanted to further improve the quality of the corpus by providing this option. The annotation interface is displayed in Figures A.3, A.4, and A.5. Figure A.3 gives an overview over the annotation interface, and the two other figures illustrate the two parts of the annotation interface: an instance to be annotated and the annotation guidelines. As illustrated in Figure A.4, the user can select and unselect sets of sentences in ETSs. Moreover, as shown in Figure A.5, the annotation interface provides an example for each bullet point, which can be accessed by pushing the `Example` button.



Find supporting or refuting sentences for a claim

The task:

- You are given a claim and an article, and you are asked to annotate supporting or refuting sentences in the article.
- Annotate sentences:
  - if they restate the claim
  - if they directly refute the claim
- If the information, which refutes or supports the claim, is distributed over several sentences, all these sentences must be selected.
- If you are asked to find **supporting** sentences and cannot find any, select the option: "The article does not contain any **supporting** sentences".
Then try to find and annotate **refuting** sentences:
  - if you have found and annotated **refuting** sentences, select the option: "The article contains the selected **refuting** sentences"
  - if you could not find any **refuting** sentences, select the option: "The article does not contain any **supporting** nor **refuting** sentences"
- If you are asked to find **refuting** sentences and cannot find any, you need to try to find **suporting** sentences
- ***NEW* Typical mistakes made by annotators in this study:**

Task: Find **refuting** sentences in the article for the following claim

Claim: Gang initiates are bumping their cars into others, then shooting whoever gets out of the bumped automobiles.

Article:

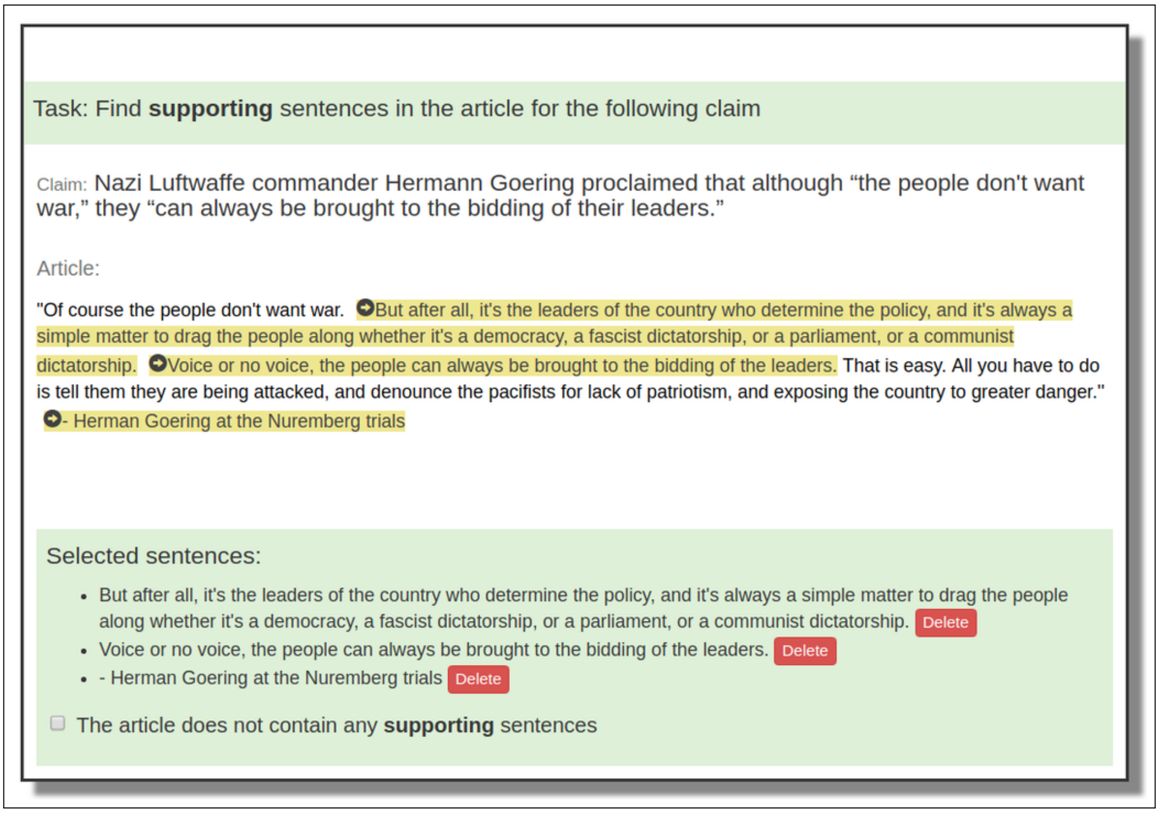
A text message warning recipients of a gang initiation in Tallahassee tonight is spreading quickly and prompting calls to both the Police Department and Sheriff's Office. The message reads: "The gang MS 13 here in tally is having initiation that starts tonight they bump your car and then when you get out they shoot you please pass this on." Spokesmen from both the Tallahassee Police Department and the Leon County Sheriff's Office call this an "urban legend" and say they have no credible information or reason to believe that any such gang activity will occur tonight.

Selected sentences:

- Spokesmen from both the Tallahassee Police Department and the Leon County Sheriff's Office call this an "urban legend" and say they have no credible information or reason to believe that any such gang activity will occur tonight.

□ The article does not contain any **refuting** sentences

Figure A.3: Annotation interface for the annotation of FGE in ETSs (entire annotation interface)



Task: Find **supporting** sentences in the article for the following claim

Claim: Nazi Luftwaffe commander Hermann Goering proclaimed that although “the people don't want war,” they “can always be brought to the bidding of their leaders.”

Article:

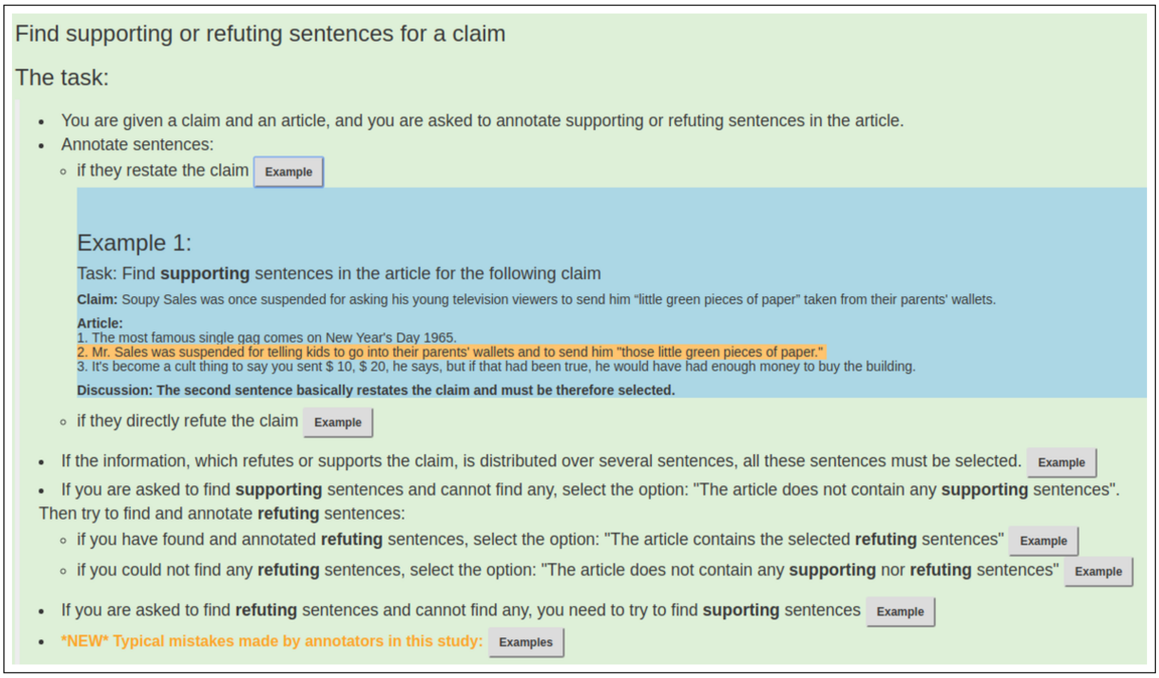
"Of course the people don't want war. But after all, it's the leaders of the country who determine the policy, and it's always a simple matter to drag the people along whether it's a democracy, a fascist dictatorship, or a parliament, or a communist dictatorship. Voice or no voice, the people can always be brought to the bidding of the leaders. That is easy. All you have to do is tell them they are being attacked, and denounce the pacifists for lack of patriotism, and exposing the country to greater danger." - Herman Goering at the Nuremberg trials

Selected sentences:

- But after all, it's the leaders of the country who determine the policy, and it's always a simple matter to drag the people along whether it's a democracy, a fascist dictatorship, or a parliament, or a communist dictatorship.
- Voice or no voice, the people can always be brought to the bidding of the leaders.
- - Herman Goering at the Nuremberg trials

□ The article does not contain any **supporting** sentences

Figure A.4: Annotation interface for the annotation of FGE in ETSs (annotation of sentence sets)



Find supporting or refuting sentences for a claim

The task:

- You are given a claim and an article, and you are asked to annotate supporting or refuting sentences in the article.
- Annotate sentences:
  - if they restate the claim Example

    Example 1:

    Task: Find **supporting** sentences in the article for the following claim

    Claim: Soupy Sales was once suspended for asking his young television viewers to send him “little green pieces of paper” taken from their parents' wallets.

    Article:
    1. The most famous single gag comes on New Year's Day 1965.
    2. Mr. Sales was suspended for telling kids to go into their parents' wallets and to send him "those little green pieces of paper."
    3. It's become a cult thing to say you sent $ 10, $ 20, he says, but if that had been true, he would have had enough money to buy the building.

    Discussion: The second sentence basically restates the claim and must be therefore selected.
  - if they directly refute the claim Example
- If the information, which refutes or supports the claim, is distributed over several sentences, all these sentences must be selected. Example
- If you are asked to find **supporting** sentences and cannot find any, select the option: "The article does not contain any **supporting** sentences". Then try to find and annotate **refuting** sentences:
  - if you have found and annotated **refuting** sentences, select the option: "The article contains the selected **refuting** sentences" Example
  - if you could not find any **refuting** sentences, select the option: "The article does not contain any **supporting** nor **refuting** sentences" Example
- If you are asked to find **refuting** sentences and cannot find any, you need to try to find **suporting** sentences Example
- ***NEW* Typical mistakes made by annotators in this study:** Examples

Figure A.5: Annotation interface for the annotation of FGE in ETSs (an opened example presented for a particular case, i.e. the **Example** button was activated for this case)

List of Figures

List of Tables

Bibliography

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68.

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims, task 1: Check-worthiness. In *CLEF 2018 Working Notes. Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, Avignon, France. CEUR-WS.org.

Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Simari, Matthias Thimm, and Serena Villata. 2017. Towards artificial argumentation. *AI magazine*, 38(3):25–36.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 876–885, Austin, TX, USA.

Ricardo Baeza-Yates, Berthier de Araújo Neto Ribeiro, et al. 2011. *Modern information retrieval.* New York: ACM Press; Harlow, England: Addison-Wesley,.

Eytan Bakshy, Solomon Messing, and Lada A Adamic. 2015. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132.

Ramy Baly, Mitra Mohtarami, James Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2018. Integrating stance detection and fact checking in a unified corpus. In *Proceedings of NAACL-HLT*, pages 21–27.

Hannah Bast, Björn Buchhold, Elmar Haussmann, et al. 2016. Semantic search on text and knowledge bases. *Foundations and Trends® in Information Retrieval*, 10(2-3):119–271.

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Christian Bizer, Tom Heath, and Tim Berners-Lee. 2011. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent dirichlet allocation. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 601–608, Vancouver, BC, Canada.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Peter Bourgonje, Julian Moreno Schneider, and Georg Rehm. 2017. From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles. In *Proceedings of the EMNLP 2017 Workshop 'Natural Language Processing meets Journalism'*, pages 84–89, Copenhagen, Denmark.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.*

Andrea Braithwaite. 2016. It's about ethics in games journalism? Gamergaters and geek masculinity. *Social Media+ Society*, 2(4):2056305116672484.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence.*

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations).*

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051.*

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Yimin Chen, Niall J Conroy, and Victoria L Rubin. 2015. Misleading online content: Recognizing clickbait as false news. In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, pages 15–19. ACM.

Anton Chernyavskiy and Dmitry Ilvovsky. 2019. Extract and aggregate: A novel domain-independent approach to factual data verification. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 69–78.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one*, 10(6).

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Costanza Conforti, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Towards automatic fake news detection: Cross-level stance detection in news articles. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 40–49.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067.

Niall J Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Sanjiv R. Das and Mike Y. Chen. 2007. Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. *Management Science*, 53(9):1375–1388.

Richard Davis and Chris Proctor. 2017. Fake News, Real Consequences: Recruiting Neural Networks for the Fight Against Fake News. Online: `http://web.stanford.edu/class/cs224n/reports/2761239.pdf`. Accessed: 2018-03-16.

Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. What is the essence of a claim? Cross-domain claim identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2066.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017).*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT).*

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3988–3994, Melbourne, Australia.

Sebastian Dungs, Ahmet Aker, Norbert Fuhr, and Kalina Bontcheva. 2018. Can rumour stance alone predict veracity? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3360–3370.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22.

Matthias Eickhoff and Nicole Neuss. 2017. Topic modelling methodology: its use in information systems and other managerial disciplines. In *Proceedings of the 25th European Conference on Information Systems.*

Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. 2017. Fake news mitigation via point process based intervention. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1097–1106. JMLR. org.

Adam Faulkner. 2014. Automated classification of stance in student essays: An approach using stance target information and the Wikipedia link-based measure. In *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014, Pensacola Beach, Florida, May 21-23, 2014*.

Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 813–820. IEEE.

William Ferreira and Andreas Vlachos. 2016. Emergent: A novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT)*, pages 1163–1168, San Diego, CA, USA.

Seth Flaxman, Sharad Goel, and Justin M Rao. 2013. Ideological segregation and the effects of social media on news consumption. *Available at SSRN*, 2363701.

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

Valery I Frants, Jacob Shapiro, Isak Taksa, and Vladimir G Voiskunskii. 1999. Boolean search: Current state and perspectives. *Journal of the American Society for Information Science*, 50(1):86–95.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. 2015. Defacto—temporal and multilingual deep fact validation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 35:85–101.

Jennifer Golbeck. 2008. *Computing with social trust*. Springer Science & Business Media.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine*. O'Reilly Media, Inc.

Jeffrey Gottfried and Elisa Shearer. 2016. *News Use Across Social Medial Platforms 2016*. Pew Research Center.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Anthony G Greenwald and Mahzarin R Banaji. 1995. Implicit social cognition: attitudes, self-esteem, and stereotypes. *Psychological review*, 102(1):4.

Ramanathan Guha, Rob McCool, and Eric Miller. 2003. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709. ACM.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64. ACM.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? A study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030.

Iryna Gurevych, Eduard H Hovy, Noam Slonim, and Benno Stein. 2016. Debating technologies (dagstuhl seminar 15512). In *Dagstuhl reports*, volume 5. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Ivan Habernal, Patrick Pauli, and Iryna Gurevych. 2018a. Adapting serious game for fallacious argumentation to german: Pitfalls, insights, and best practices. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018b. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT)*, pages 1930–1940, New Orleans, LA, USA.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018c. Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 386–396, New Orleans, Louisiana. Association for Computational Linguistics.

Andreas Hanselowski, PVS Avinesh, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018a. A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, and Felix Caspelherr. 2017. Description of the system developed by team Athene in the FNC-1, 2017. Online: `https://github.com/hanselowski/athene_system/blob/master/system_description_athene.pdf`. Accessed: 2018-03-13.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In

Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 493–503.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018b. UKP-Athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1348–1356, Nagoya, Japan.

Naeemul Hassan, Bill Adair, James T Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The quest to automate fact-checking. *world*.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, et al. 2017. Claimbuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.

Michiel Hermans and Benjamin Schrauwen. 2013. Training and analysing deep recurrent neural networks. In *Advances in neural information processing systems 26 (NIPS)*, pages 190–198, Stateline, NV, USA.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Benjamin D. Horne and Sibel Adali. 2017. This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. In *Proceedings of the ICWSM 2017 Workshop on News and Public Opinion*, pages 759–766, Montréal, QC, Canada.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning Whom to Trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT)*, pages 1120–1130, Atlanta, GA, USA.

Lee Howell et al. 2013. Digital wildfires in a hyperconnected world. *WEF Report*, 3:15–94.

Xinyu Hua and Lu Wang. 2017. Understanding and detecting supporting arguments of diverse types. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 203–208.

Jiang Huiping. 2010. Information retrieval and the semantic web. In *2010 International Conference on Educational and Information Technology*, volume 3, pages V3–461. IEEE.

Anderson Jonathan. 1983. Lix and Rix: Variations on a Little-known Readability Index. *Journal of Reading*, 26(6):490–496.

Garth Jowett and Victoria O' Donnell. 2006. What is propaganda, and how does it differ from persuasion? *Propaganda and Misinformation, Chapter 1.*

Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. 2019. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (BERT). *Applied Sciences*, 9(19):4062.

J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Research Branch Report 8-75, Naval Technical Training Command, Millington, TN, USA.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

David Klein and Joshua Wueller. 2017. Fake news: a legal perspective. *Journal of Internet Law (Apr. 2017).*

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2018. Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. In *Proceedings of the 1st Workshop on Fact Extraction and VERification (FEVER).*

Bill Kovach and Tom Rosenstiel. 2014. *The elements of journalism: What newspeople should know and the public should expect.* Three Rivers Press (CA).

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Ch. Aswani Kumar, M Radvansky, and J Annapurna. 2012. Analysis of a vector space model, latent semantic indexing and formal concept analysis for information retrieval. *Cybernetics and Information Technologies*, 12(1):34–48.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics.*

Neil Levy. 2017a. The bad news about fake news. *Social Epistemology Review and Reply Collective*, 6(8):20–36.

Neil Levy. 2017b. Nudges in a post-truth world. *Journal of medical ethics*, 43(8):495–500.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500.

Quanzhi Li, Qiong Zhang, and Luo Si. 2019. Rumor detection by exploiting user credibility information, attention and multi-task learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1173–1179, Florence, Italy. Association for Computational Linguistics.

Zile Li. 2018. Claim validation for the FEVER shared task. Master's thesis, Technische Universität Darmstadt.

Chih-Jen Lin. 2007. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779.

Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80.

Marco Lippi and Paolo Torroni. 2016a. Argument mining from speech: Detecting claims in political debates. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Marco Lippi and Paolo Torroni. 2016b. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2):10.

Marco Lippi and Paolo Torroni. 2016c. Margot: A web server for argumentation mining. *Expert Systems with Applications*, 65:292–303.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Zhenghao Liu, Chenyan Xiong, and Maosong Sun. 2019b. Kernel graph attention network for fact verification. *arXiv preprint arXiv:1910.09796*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Christopher Malon. 2018. Team papelo: Transformer networks at fever. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Ana Marasovic. 2018. NLP's generalization problem, and how researchers are tackling it. *The Gradient.*

Gary Marcus. 2018. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631.*

Coleman Mari and Liau Ta Lin. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Alice Marwick and Rebecca Lewis. 2017. Media manipulation and disinformation online. *New York: Data Society Research Institute.*

G. Harry Mc Laughlin. 1969. SMOG grading—a new readability formula. *Journal of reading*, 12(8):639–646.

Rachel McAlpine. 1997. *Global English for global business.* Longman.

Philip M. McCarthy. 2005. An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD). *Dissertation Abstracts International*, 66:12.

Ryan McDonald, George Brokos, and Ion Androutsopoulos. 2018. Deep relevance ranking using enhanced document-query interactions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1849–1860.

Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509.*

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*, pages 31–41, San Diego, CA, USA.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval)*, pages 321–327, Atlanta, GA, USA.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In *Proceedings of the NAACL/HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA, USA.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic stance detection using end-to-end memory networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 767–776.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Moin Nadeem, Wei Fang, Brian Xu, Mitra Mohtarami, and James Glass. 2019. FAKTA: An automatic end-to-end fact checking system. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT)*.

Arpitha Nagaraja. 2017. Development of a system for automated fact-checking - corpus construction and evidence extraction. Master's thesis, Technische Universität Darmstadt.

Preslav Nakov, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Lluís Màrquez, Wajdi Zaghouani, Pepa Atanasova, Spas Kyuchukov, and Giovanni Da San Martino. 2018. Overview of the CLEF-2018 checkthat! Lab on automatic identification and verification of political claims. In *Proceedings of the Ninth International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Lecture Notes in Computer Science, Avignon, France. Springer.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Zhiyuan Ning, Yuanzhen Lin, and Ruichao Zhong. 2019. Team peter-parker at semeval-2019 task 4: Bert-based method in hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1037–1040.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you*. Penguin UK.

Danny Paskin. 2018. Real or fake news: Who knows? *The Journal of Social Media in Society*, 7(2):252–273.

Jeff Pasternack and Dan Roth. 2013. Latent credibility analysis. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1009–1020. ACM.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401.

Nathaniel Persily. 2017. The 2016 US election: Can democracy survive the internet? *Journal of democracy*, 28(2):63–76.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Dean Pomerleau and Delip Rao. 2017. The Fake News Challenge: Exploring how artificial intelligence technologies could be leveraged to combat fake news. `http://www.fakenewschallenge.org/`. Accessed: 2017-10-20.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2016. Credibility assessment of textual claims on the web. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2173–2178. ACM.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012. International World Wide Web Conferences Steering Committee.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 231–240.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Gordon Ramsay and Sam Robertshaw. 2019. Weaponizing news RT, Sputnik and targeted disinformation. *King's College London: The Policy Institute, Center for the Study of Media, Communication, and Power*.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937.

Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence-an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, pages 451–463, Denver, CO, USA.

Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second Workshop on Computational Approaches to Deception Detection*, pages 7–17.

Victoria L Rubin, Yimin Chen, and Niall J Conroy. 2015. Deception detection for news: three types of fakes. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, page 83. American Society for Information Science.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806. ACM.

Andreas Rücklé and Iryna Gurevych. 2017. Representation learning for answer selection with LSTM-based importance weighting. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers*.

Tuukka Ruotsalo. 2012. Domain specific data retrieval on the semantic web. In *Extended Semantic Web Conference*, pages 422–436. Springer.

SafeGuardCyber. 2019. Contactless actions against the enemy: How russia is deploying misinformation on social media to influence european parliamentary elections. https://www.safeguardcyber.com/resources/white-papers/eu-election-security. Accessed: 2019-10-15.

Eric SanJuan, Fidelia Ibekwe-SanJuan, Juan-Manuel Torres-Moreno, and Patricia Velázquez-Morales. 2007. Combining vector space model and multi word term extraction for semantic query expansion. In *International Conference on Application of Natural Language to Information Systems*, pages 252–263. Springer.

Benjamin Schiller. 2017. Development of machine learning methods for automated claim validation. Master's thesis, Technische Universität Darmstadt.

Jodi Schneider. 2014. Automated argumentation mining to the rescue? Envisioning argumentation and decision-making support for debates in open online collaboration communities. In *Proceedings of the First Workshop on Argumentation Mining*, pages 59–63.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Baird Sean, Sibley Doug, and Pan Yuxi. 2017. Talos Targets Disinformation with Fake News Challenge Victory. http://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html. Accessed: 2017-12-02.

R.J. Senter and Edgar A. Smith. 1967. Automated readability index. Technical Report AMRL-TR-66-220, University of Cincinnati.

Chengcheng Shao, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. 2016. Hoaxy: A platform for tracking online misinformation. In *Proceedings of the 25th international conference companion on world wide web*, pages 745–750. International World Wide Web Conferences Steering Committee.

Ivor Shapiro, Colette Brin, Isabelle Bédard-Brûlé, and Kasia Mychajlowycz. 2013. Verification as a strategic ritual: How journalists retrospectively describe processes for ensuring accuracy. *Journalism Practice*, 7(6):657–673.

Baoxu Shi and Tim Weninger. 2016a. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowledge-Based Systems*, 104:123–133.

Baoxu Shi and Tim Weninger. 2016b. Fact checking in heterogeneous information networks. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 101–102. International World Wide Web Conferences Steering Committee.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

Richard H Shultz and Roy Godson. 1984. *Dezinformatsia: Active measures in Soviet strategy*. Potomac Books.

Craig Silverman and Jeremy Singer-Vine. 2016. Most americans who see fake news believe it, new survey says. https://www.buzzfeednews.com/article/craigsilverman/fake-news-survey. Accessed: 2020-3-10.

Jacob Silverman. 2016. *Terms of service: social media and the price of constant connection*. Harper Perennial.

Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. https://www.blog.google/products/search/introducing-knowledge-graph-things-not/. Accessed: 2020-3-11.

Noam Slonim, Iryna Gurevych, Chris Reed, and Benno Stein. 2016. Nlp approaches to computational argumentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*.

Amir Soleimani, Christof Monz, and Marcel Worring. 2019. Bert for evidence retrieval and claim verification. *arXiv preprint arXiv:1910.02655*.

N. Watson Solomon. 2006. Strain index: A new readability formula. Master thesis, Madurai Kamaraj University, December.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL/HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA, USA.

Daniil Sorokin and Iryna Gurevych. 2018. Mixing context granularities for improved entity linking on question answering data across entity categories. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 65–75.

Dhanya Sridhar, James R. Foulds, Bert Huang, Lise Getoor, and Marilyn A. Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL/IJCNLP)*, pages 116–125, Beijing, China.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018a. Argumentext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25.

Christian Stab and Iryna Gurevych. 2014. 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.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Christian Stab and Ivan Habernal. 2016. Existing resources for debating technologies. In *Report of Dagstuhl Seminar 15512, Debating Technologies.*

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018b. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674.

Chris Stahlhut. 2019. Interactive evidence detection: train state-of-the-art model out-of-domain or simple model interactively? In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 79–89.

Chris Stahlhut, Christian Stab, and Iryna Gurevych. 2018. Pilot experiments of hypothesis validation through evidence detection for historians. In *Proceedings of the First Biennial Conference on Design of Experimental Search & Information Retrieval Systems*, volume 2167 of *CEUR Workshop Proceedings*, pages 83–89.

Sanja Štajner, Richard Evans, Constantin Orăsan, and Ruslan Mitkov. 2012. What can readability measures really tell us about text complexity. In *Proceedings of the LREC 2012 Workshop on Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, pages 14–21, Istanbul, Turkey.

Stanford. 2017. Cs224n: Natural language processing with deep learning, course project reports for 2017. `http://web.stanford.edu/class/cs224n/reports.html`. Accessed: 2017-12-13.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *6th International Conference on Learning Representations.*

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7464–7473.

Cass R Sunstein. 2014. *On rumors: How falsehoods spread, why we believe them, and what can be done.* Princeton University Press.

Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some like it hoax: Automated fake news detection in social networks. In *2nd Workshop on Data Science for Social Good, SoGood 2017*, pages 1–15. CEUR-WS.

Christopher Tauchmann, Thomas Arnold, Andreas Hanselowski, Christian M Meyer, and Margot Mieskes. 2018. Beyond generic summarization: A multi-faceted hierarchical summarization corpus of large heterogeneous data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018).*

Mariona Taulé, Maria Antònia Martí, Francisco M. Rangel Pardo, Paolo Rosso, Cristina Bosco, and Viviana Patti. 2017. Overview of the Task on Stance and Gender Detection in Tweets on Catalan Independence. In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval) co-located with 33th Conference of the Spanish Society for Natural Language Processing (SEPLN)*, pages 157–177, Murcia, Spain.

Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Leif Thomas and Bertram Weiss. 2010. *Fact-Checking und redaktionelles Qualitätsmanagement. In: netzwerk recherche e. V. (Hrsg.): Fact-Checking: Fakten finden, Fehler vermeiden.* Geschäftsstelle, Stubbenhuk 10, 20459 Hamburg.

James Thorne, Mingjie Chen, Giorgos Myrianthous, Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. 2017. Fake news stance detection using stacked ensemble of classifiers. In *Proceedings of the EMNLP 2017 Workshop 'Natural Language Processing meets Journalism'*, pages 80–83, Copenhagen, Denmark.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359.

James Thorne and Andreas Vlachos. 2019. Adversarial attacks against fact extraction and verification. *arXiv preprint arXiv:1903.05543.*

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: A large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22.

Andreas Vlachos and Sebastian Riedel. 2015. Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2596–2601. Association for Computational Linguistics.

Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85.

Ari Ezra Waldman. 2017. The marketplace of fake news. *University of Pennsylvania Journal of Constitutional Law*, 20:845.

Marilyn A. Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT)*, pages 592–596, Montréal, QC, Canada.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2017. Evidence aggregation for answer re-ranking in open-domain question answering. *arXiv preprint arXiv:1711.05116*.

William Yang Wang. 2017. "Liar, Liar Pants on Fire": A new benchmark dataset for fake news detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Helena Webb, Marina Jirotka, Bernd Carsten Stahl, William Housley, Adam Edwards, Matthew Williams, Rob Procter, Omer Rana, and Pete Burnap. 2016. Digital wildfires: Hyper-connectivity, havoc and a global ethos to govern social media. *ACM SIGCAS Computers and Society*, 45(3):193–201.

Jen Weedon, William Nuland, and Alex Stamos. 2017. Information operations and Facebook. *Facebook*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 347–354, Vancouver, BC, Canada.

Ruoyao Yang, Wanying Xie, Chunhua Liu, and Dong Yu. 2019a. BLCU_NLP at SemEval-2019 task 7: An inference chain-based GPT model for rumour evaluation. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1090–1096, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Wenpeng Yin and Dan Roth. 2018. TwoWingOS: A two-wing optimization strategy for evidential claim verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Ucl machine reading group: Four factor framework for fact finding (hexaf). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102.

Sauleh Eetemadi Zarharan, Samane Ahangar, Fateme Sadat Rezvaninejad, Mahdi Lotfi Bidhendi, Mohammad Taher Pilevar, and Behrouz Minaei. 2019. Persian stance classification data set. In *Proceedings of the conference for Truth and Trust Online.*

Guido Zarrella and Amy Marsh. 2016. MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 458–463, San Diego, CA, USA.

Klaus Zechner. 2002. Automatic Summarization of Open-Domain Multiparty Dialogues in Diverse Genres. *Computational Linguistics*, 28(4):447–485.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pages 9051–9062.

Hao Zhang. 2018. Stance detection and evidence extraction for automated fact-checking. Master's thesis, Technische Universität Darmstadt.

Qiang Zhang, Emine Yilmaz, and Shangsong Liang. 2018. Ranking-based method for news stance detection. In *Companion Proceedings of the The Web Conference 2018*, pages 41–42. International World Wide Web Conferences Steering Committee.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. Gear: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901.

Xinyi Zhou and Reza Zafarani. 2018. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315*.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif M. Mohammad. 2014. NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*, pages 443–447, Dublin, Ireland.