Challenges of Computational Social Science Analysis with NLP Methods

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung.

Vorgelegt von

Erenay Dayanik

aus Karabuk, Turkei

Hauptberichter	Prof. Dr. Sebastian Padó
Mitberichter	Prof. Dr. Barbara Plank

Tag der mündlichen Prüfung: 14.10.2022
Institut für Maschinelle Sprachverarbeitung
der Universität Stuttgart

2022

Erklärung (Statement of Authorship)
Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet.

Abstract

Computational Social Science (CSS) is an emerging research area at the intersection of social science and computer science, where problems of societal relevance can be addressed by novel computational methods. With the recent advances in machine learning and natural language processing as well as the availability of textual data, CSS has opened up to new possibilities, but also methodological challenges. In this thesis, we present a line of work on developing methods and addressing challenges in terms of data annotation and modeling for computational political science and social media analysis, two highly popular and active research areas within CSS.

In the first part of the thesis, we focus on a use case from computational political science, namely Discourse Network Analysis (DNA), a framework that aims at analyzing the structures behind complex societal discussions. We investigate how this style of analysis, which is traditionally performed manually, can be automated. We start by providing a requirement analysis outlining a roadmap to decompose the complex DNA task into several conceptually simpler subtasks. Then, we introduce NLP models with various configurations to automate two of the sub-tasks given by the requirement analysis, namely claim detection and classification, based on different neural network architectures ranging from unidirectional LSTMs to

Transformer based architectures.

In the second part of the thesis, we shift our focus to fairness, a central concern in CSS. Our goal in this part of the thesis is to analyze and improve the performances of NLP models used in CSS in terms of fairness and robustness while maintaining their overall performance. With that in mind, we first analyze the above-mentioned claim detection and classification models and propose techniques to improve model fairness and overall performance. After that, we broaden our focus to social media analysis, another highly active subdomain of CSS. Here, we study text classification of the correlated attributes, which pose an important but often overlooked challenge to model fairness. Our last contribution is to discuss the limitations of the current statistical methods applied for bias identification; to propose a multivariate regression based approach; and to show that, through experiments conducted on social media data, it can be used as a complementary method for bias identification and analysis tasks.

Overall, our work takes a step towards increasing the understanding of challenges of computational social science. We hope that both political scientists and NLP scholars can make use of the insights from this thesis in their research.

Zusammenfassung

Die computergestützte Sozialwissenschaft (CSS) ist ein aufstrebendes Forschungsgebiet an der Schnittstelle zwischen Sozialwissenschaft und Informatik, in dem Probleme von gesellschaftlicher Relevanz mit neuartigen computergestützten Methoden angegangen werden können. Mit den jüngsten Fortschritten im Bereich des maschinellen Lernens und der Verarbeitung natürlicher Sprache sowie der Verfügbarkeit grosser Datenmengen haben sich der CSS neue Möglichkeiten, aber auch methodische Herausforderungen eröffnet. In dieser Dissertation stellen wir eine Reihe von Arbeiten vor, die sich mit der Entwicklung von Methoden und der Bewältigung von Herausforderungen in Bezug auf die Datenannotation und -modellierung für die computergestützte Politikwissenschaft und die Analyse sozialer Medien befassen, zwei sehr aktive Forschungsbereiche innerhalb von CSS.

Im ersten Teil der Arbeit konzentrieren wir uns auf einen Anwendungsfall aus der computergestützten Politikwissenschaft, nämlich der Diskursnetzwerkanalyse (DNA), deren Ziel ist, die Strukturen hinter komplexen gesellschaftlichen Debatten herauszuarbeiten. Wir untersuchen, wie Analysen dieser Art, die traditionell manuell durchgeführt werden, automatisiert werden können. Wir beginnen mit einer Anforderungsanalyse und skizzieren eine Roadmap zur Zerle-

gung der komplexen DNA-Aufgabe in mehrere konzeptionell einfachere Teilaufgaben. Dann stellen wir NLP-Modelle mit verschiedenen Konfigurationen vor, um zwei der durch die Anforderungsanalyse vorgegebenen Teilaufgaben zu automatisieren, nämlich die Erkennung und Klassifizierung von Behauptungen, basierend auf verschiedenen neuronalen Netzwerkarchitekturen, die von unidirektionalen LSTMs bis zu transformer-basierten Architekturen reichen.

Im zweiten Teil der Arbeit konzentrieren wir uns auf Fairness, ein zentrales Anliegen von CSS. Unser Ziel in diesem Teil der Arbeit ist es, die Leistung von NLP-Modellen, die in CSS verwendet werden, im Hinblick auf Fairness und Robustheit zu analysieren und zu verbessern, ohne dabei ihre Gesamtleistung zu beeinträchtigen. Zu diesem Zweck analysieren wir zunächst die oben erwähnten Modelle zur Erkennung und Klassifizierung von Behauptungen und schlagen Techniken zur Verbesserung der Fairness und der Gesamtleistung der Modelle vor. Danach weiten wir unseren Fokus auf die Analyse sozialer Medien aus, einem weiteren sehr aktiven Teilbereich von CSS. Hier untersuchen wir zunächst die Textklassifizierung der korrelierten Attribute, die eine wichtige, aber oft übersehene Herausforderung für die Modellgerechtigkeit darstellen. Im letzten experimentellen Kapitel erörtern wir dann die Grenzen der derzeitigen statistischen Methoden zur Identifikation von bias (inhärenter Ungleichbehandlung), schlagen einen auf multivariater Regression basierenden Ansatz vor und zeigen anhand von Experimenten mit Social-Media-Daten, dass dieser als ergänzende Methode zur Identifizierung von Verzerrungen und für Analyseaufgaben eingesetzt werden kann.

Zusammengefaßt stellt unsere Arbeit einen Schritt in Richtung eines besseren Verständnisses der Herausforderungen der computergestützten Sozialwissenschaften dar. Wir hoffen, dass wir es sowohl Politikwissenschaftlern als auch NLP-Forschern ermöglichen, die Erkenntnisse dieser Arbeit für ihre eigene Forschung nutzbar zu machen.

Acknowledgements

I would like to express my utmost gratitude to my advisor Prof. Dr. Sebastian Padó for spending countless hours in the last couple of years to listen to my research problems (and sometimes personal problems as well) and to guide me. I'd like to thank him also for having belief in me even when I have doubts on my abilities. I have learnt a lot from him both academically as well as personally.

I would like to also express my gratitude to the members of my thesis examination committee, Prof. Dr. Barbara Plank, Prof. Dr. Michael Sedlmair, Dr. Roman Klinger and Prof. Dr. Marco Aiello for dedicating precious time in reviewing this thesis as well as for challenging me with insightful questions and providing valuable suggestions.

I would next like to thank the members of MARDY project, Prof. Dr. Jonas Kuhn, Prof. Dr. Sebastian Haunss, Dr. André Blessing, Dr. Nico Blokker, Dr. Gabriella Lapesa and Tanise Ceron for the discussions, inspiration and cooperation.

I would like to give another hearty thank you to all of my friends and colleagues at the TCL group with whom I shared office space, many cups of coffee, biscuits and insightful discussions.

Next, I would like to thank my mom and my dad for their unbreakable support, trust and love, and to my brother who has opened up

my vision and made me believe he is always there whenever I need him.

And finally, I would like to thank my dear wife, Secil Unsal Dayanik, who has been by my side throughout this PhD, living every single minute of it, lifting me up from all my falls, and without whom, I am sure that this thesis would not have become a reality.

Contents

List of Figures

List of Tables

Part I

Introduction and Background

1 Introduction

1.1 Locating Computational Political Text Analysis

We begin by providing an overview of the main fields of research that this thesis overlaps with, in order to localize it.

Political Science is a social science subject aiming to describe, analyse and explain the workings of government and relationship between political and non-political institutions and processes (Heywood, 2015). It consists of three major subfields, which are comparative politics, international relations, and political theory, as well as several minor subfields such as political economy, political methodology and political communication. Research in political science covers a wide range of topics including public opinion (Burstein, 2010), politics and the gender gap (Mendelberg et al., 2014), human rights oppression (Milner et al., 1999), law enforcement and its effects (Møller and Skaaning, 2013; Oztig and Donduran, 2020), and the political control of bureaucracy (Dahlström and Holmgren, 2019; Bach et al., 2020).

Political Communication, as one of the main subfields of political science, concerns with how information spreads and influences politics, policy makers, the news media, and citizens (Smelser et al., 2001). Political science researchers have long recognized the value of textual sources such as newspapers, social media, party manifestos for political communication: They are one of the primary mediums for political actors to interact with public (and vice versa), run political campaigns, to achieve certain political goals and to shape public opinion in their own way, and hence analysis of such political texts can provide an understanding of the current political process and events (Subramanian, 2020).

Political Text Analysis has traditionally been performed manually due to the lack of automated tools. However, recent advances in Machine Learning (ML) and Natural Language Processing (NLP) have opened up the possibility of being able to automate this analysis, leading to the rise of **Computational Social Science (CSS)** (Lazer et al., 2009). CSS is an emerging interdisciplinary field which is driven by new sources of data from the Internet, sensor networks, government databases, and aims to empirically study topics related to the social sciences such as economics, sociology and political science with the help of computational methods.

Early research in **Computational Political Text Analysis** has focused on relatively less complicated document-level tasks such as political text scaling (Slapin and Proksch, 2008), political orientation classification (Cohen and Ruths, 2013) and topic classification (Hopkins and King, 2010; Karan et al., 2016) using conventional statistical methods such as Naïve Bayes classifier, SVMs and static word embeddings, which require relatively little annotated data and expertise to train effectively. However, just like in many other domains, the recent advances such as development of dedicated resources and significantly better performing NLP models based on LSTMs- and Transformers-based architectures are changing the level of analysis performed in political discourse, enabling researchers to perform deeper semantic and structured analysis such as political event extraction (Hürriyetoğlu et al., 2019, 2020), aspect-controlled argument generation (Schiller et al., 2020), fact checking (Ostrowski et al., 2021) and entity level sentiment analysis (Yang et al., 2021), to name just a few applications.

Political debates with their complex structures, both semantically and syntactically, are one of the main text types in political discourse (Kersting, 2005). In democratic countries, most political decisions which are bound to affect a substantial part of the community attract public attention and thus accompanied by public debates (De Wilde, 2011; Zürn, 2014; Haunss et al., 2020). To better understand democratic decision-making, one needs a fine-grained representation of such debates and their dynamics that captures specific aspects of the debate topic and represents how the structure of agreement (or disagreement) develops around such aspects. Hence, understanding the structure and evolution of political debates has always been a popular research topic in political science domain and various analysis techniques have been developed over the years, such as advocacy coalition framework (Sabatier and Jenkins-Smith, 1993) and Discourse Network Analysis (Leifeld, 2009).

Discourse Network Analysis (DNA) is a framework for the structural representation and the analysis of policy debates where a policy debate is a political discourse centered on a given policy such as immigration or climate change. It builds on the assumption that policy debates can be modeled as a affiliation network as

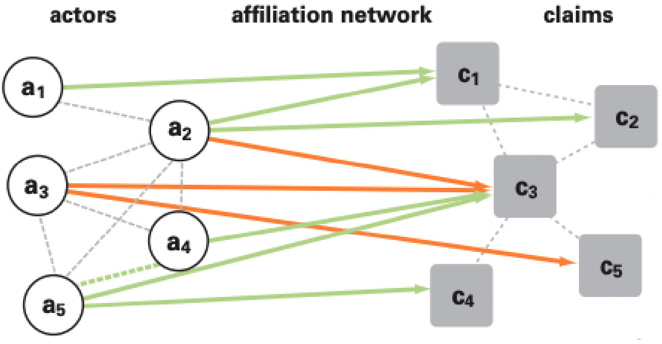


Figure 1: Example of an affiliation network

the one depicted in Figure 1.1. The network contains two types of nodes: actors(circles) and claim categories (squares). Edges between the claims and the actors encode the fact that the actor made a statement regarding the specific claim categories, along with the polarity of this statement. Use of DNA on affiliation networks allows to empirically track the evolution of discourse coalitions, which are groups of actors who are bound together according to shared ideas, over time and to identify conditions for their success in terms of dominating political debates and influencing policy-making (Leifeld and Haunss, 2011; Haunss et al., 2013).

Traditionally, DNA studies heavily relies on expensive human effort both for coding of data and creation of the networks, limiting their scope to the amount of data that can be considered. However, as we are going to discuss in more detail in the upcoming chapters of this thesis, this process can be (semi-)automated thanks to the recent advances in natural language processing and machine learning.

This automatization can substantially broaden the empirical basis and applicability of discourse network studies, but at the same time it introduces new challenges, as we discussed in Section 1.2.

1.2 Challenges of Computational Analysis of Political Discourse

It is true that recent developments in machine learning enable political scientists to work with corpus sizes that are infeasible for manual analysis and let them to explore new research questions or to study old research questions by new means, hence create remarkable new possibilities. However, there are still challenges for the community. We group these challenges into two main categories: Data Annotation and Modeling.

Data Annotation. Data is essential for all researchers, regardless of the domain of interest. A central challenge of working with data of any sort is that it must be organized and classified so that the researcher can use it for the task at hand. This process of labeling and organizing data for further analysis is called data annotation and it is considered as one of the first steps of conducting research in many domains. In that regard, political science is not an exception. Regardless of the way the analysis performed, much of the work on computational analysis of political texts has been enabled by the development of dedicated datasets through annotation (*coding* in the political science terminology) using carefully prepared annotation guidelines (*codebook* in the political science terminology).

While in a manual analysis the goal is to fully annotate the data, in a computational analysis the annotations are often used to train computational models to automatically recognize patterns that are associated with the labels of task at hand (Cardie and Wilkerson, 2008). We observe two data annotation related issues as shown below. Note that these issues are not specific to the political science domain, on the contrary, they are well known issues in (computational) linguistics.

- Data annotation is extremely expensive in both annotator-hours and financial cost (Snow et al., 2008). The process often starts with creation of codebook which is a document containing the coding instructions. Usually codebook generation requires several iterations, in which the instructions are updated and examples are added. For example, political scientists using deductive-inductive strategy, a popular codebook generation method (Saldaña, 2009), start with a set of deductive coding rules without examining examples first. These are then tested and adjusted based on a sample of the data. The final annotations can be gathered by a group of experts, a crowd-sourcing tool or a smaller number of well-trained annotators. However, training is required in all cases. While human annotators become more efficient with practice, the marginal cost of coding each document does not substantially decline and with more and more data coming online, and relying on human annotators becomes prohibitively expensive. As a result, there is an increasing tendency between researchers to question the value of such labor-intensive approaches (Hillard et al., 2008). With the introduction of well-performing computational models (which can ideally be trained without need for a large amount of annotated training data), this particular challenge can be eased.

- Ensuring high annotation consistency is another data annotation related challenge. If the quality of annotation is not good enough, the computational model does not get trained well, resulting in poor performance. This may arise as a result of various reasons with different level of importance. For example, one reason that can be solved relatively easily would be that the annotators are not interested or focused about the annotation task (Hovy and Prabhumoye, 2021). As another example to this category, annotation reliability issues may also arise when there is a mismatch between social and linguistic norms of annotators and authors of the data (Sap et al., 2019). While these issues related to annotation can easily be fixed by, for example, hiring new annotators, some other reasons may be harder to handle. For instance, Plank et al. (2014) report that in some annotation tasks, such as part-of-speech tagging, there might be more than one possible correct label[1], which can cause systematic differences in annotations between annotators. Unlike the previous cases, this problem can't be solved by hiring a new annotator team and requires more sophisticated solutions, such as having varying model training dynamics based on the agreement level between annotators.

[1] Drawing an example from Plank et al. (2014): *'social media'* can be treated as a noun phrase with an adjective and a noun, as well as a compound noun with two nouns.

Modeling. Besides the data annotation challenges discussed above, we also identify some modeling-related challenges associated with computational analysis of political text:

- As a result of the gap between limited linguistic variations in the training data and the diversity in real-world languages, models trained on a specific dataset are likely to rely on statistical irregularities or spurious correlations between target label and an attribute grounding inside (i.e. language internal) or outside of the language (i.e. language-external) to achieve better overall performance (Hovy and Prabhumoye, 2021). Such models are usually susceptible to poor generalization because models learn to identify the correlations between given examples and their labels rather than true intrinsic factors of the task. Besides poor generalization, such shortcuts also give rise to biased predictions. For example, Sap et al. (2019) has recently shown that there is high correlation between the existence of certain markers associated with a certain ethnic dialect of English and toxic labels in some of the widely used hate speech detection datasets and that models trained on these corpora propagate race bias such that tweets by self-identified African Americans are two times more likely to be labelled as offensive than others. Therefore, it is good practice to assume that every statistical model will involve some form of unintended bias and to perform additional evaluations in this regard so that potential biases can be detected before the models are released.

- The codebooks used in political science domain typically consist of fine-grained categories usually organized in a hierarchical structure, as a reflection of the complexity of the problems addressed. Although such a setup is required for political scientists to gain more insights and understanding on the subject, for computational models, it presents a challenging situation with increased number of classes (attested with widely different frequencies), decreased number of samples per class, and decreased semantic differences between classes, causing standard computational methods to perform badly. Ways to improve performance in such cases involve incorporating implicit or explicit sources of domain-knowledge into the models.

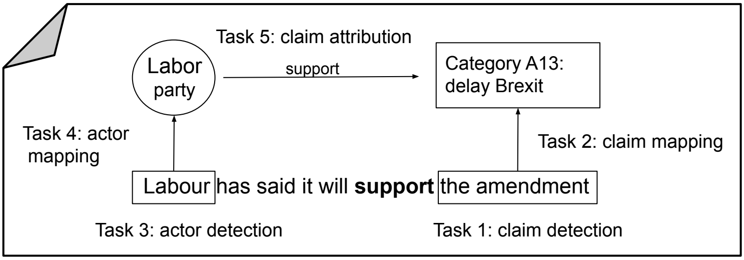


Figure 2: Workflow for Computational Construction of Affiliation Networks

1.3 Contributions

This work makes contributions in two directions:

1. **System-wise.** Our contributions on this aspect are threefold. We first outline the road towards using computational

methods from natural language processing for the construction of discourse networks as shown in Figure 2. After sketching a workflow for semi-automatic analysis of public debates, we continue with developing semantic NLP models using state-of-the-art Neural Network based NLP techniques such as LSTMs and Transformers for Political Claim Identification and Claim Classification tasks. Finally, We perform a case study on a manually annotated corpus of the German migration debate and show that using NLP models for the aforementioned tasks, it is possible to partially automate the annotation process as well as improve annotation quality and consistency.

2. **Fairness-wise.** Our contributions on fairness aspect are also threefold. First, we start with investigating presence of frequency bias, a language internal attribute, in our claim identifier and classifier models. By comparing models performances on evaluation sets with varying frequencies, we find that our models exhibit frequency bias and propose lightweight methods to improve fairness of models. Second, we continue with investigating presence of language external bias types in the CSS models. For this study, we pick gender as the bias attribute and focus on the text classification task on the social media analysis domain, another subfield of CSS. Our third and last contribution is to propose a regression analysis based bias evaluation framework which allows to, unlike current evaluation methods, quantify the contribution of multiple attributes (language internal or external) to the observed bias with measures of effect size. We demonstrate the practical application

and usefulness of this workflow by reanalyzing the predictions of a range of emotion intensity prediction models trained on social media data.

1.4 Structure of Thesis & Previous Publications

Structure of the thesis. This thesis is structured into four parts. Chapter 1 of Part I (the current part) introduces the themes of this thesis and Chapter 2 provides the background in Natural Language Processing as well as Computational Social Science needed to understand the thesis. Then, Chapter 3 introduces DebateNet which is a dataset for German which covers the public debate on immigration in 2015, and has been used through this thesis.

Part II presents our system-wise contributions. We first provide a requirement analysis for computational construction of the discourse networks using methods from natural language processing (Chapter 4). Next, in Chapter 5 we introduce our neural network based models for detecting political claims in a text and present our results on DebateNet corpus. Then, in Chapter 6, we move to claim classification task, which is another step in our requirement analysis. Similar to previous chapter, we present claim classification models based on different neural network architectures ranging from unidirectional LSTM to Transformer based architectures and evaluate them on DebateNet.

In Part III, we shift our focus to fairness. Our aim is to analyze and improve the performances of NLP models in terms of fairness and robustness while maintaining (or improving) their overall performance. With that in mind, we first analyse the claim detection and classification models introduced in Part II and propose lightweight methods to improve fairness and overall performance in Chapter 7 and Chapter 8 respectively. In Chapter 9 we broaden our focus to social media analysis and present a case study to focus on text classification for social media analysis in the context of correlated attributes. Finally, in Chapter 10 we introduce our multivariate regression based bias analysis approach.

Lastly, Part IV presents the conclusions drawn from this research work and the possible future work that is required to further answer some open ended questions which are presently outside the scope of this work (Chapter 11).

Previous Publications. A portion of the work presented here has previously been published.

- Dayanık, E., Vu, T. & Padó, S. (2022). Analysis of Bias in NLP Models With Regression and Effect Sizes. Northern European Journal of Language Technology 8.1. **(Chapter 10)**

- Dayanık, E., Blessing, A., Blokker, N., Haunss, S., Kuhn, J., Lapesa, G., & Padó, S. (2022). Improving Neural Political Statement Classification with Class Hierarchical Information. In Findings of the Association for Computational Linguistics (pp. 2367-2382). **(Chapter 8)**

- Dayanık, E., Blessing, A., Blokker, N., Haunss, S., Kuhn, J., Lapesa, G., & Padó, S. (2021). Using Hierarchical Class Structure to Improve Fine-Grained Claim Classification. In Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021) (pp. 53-60). **(Chapter 6, 8)**

- Dayanık, E., & Padó, S. (2021). Disentangling document topic and author gender in multiple languages: Lessons for adversarial debiasing. In Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 50-61). **(Chapter 9)**
- Dayanık, E., & Padó, S. (2020). Masking actor information leads to fairer political claims detection. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 4385-4391). **(Chapter 7)**
- Padó, S., Blessing, A., Blokker, N., Dayanık, E., Haunss, S., & Kuhn, J. (2019). Who sides with whom? towards computational construction of discourse networks for political debates. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2841-2847). **(Chapter 4, 5)**

2 Background

2.1 Natural Language Processing

2.1.1 Fundamentals of Machine Learning

The work in this thesis concerns a family of machine learning algorithms known as *Supervised Learning*, one of the main paradigms in machine learning. In supervised learning, the goal is to create models that can look at examples and produce generalizations (Goodfellow et al., 2016). More precisely, for a given annotated dataset with input-output pairs $((x_1,y_1),\ldots,(x_n,y_n))$, and a function $y = f(x;\Theta)$ where x is the input and Θ denotes the function parameters, the goal is to search for well-behaved Θ values so that unseen inputs $(x_1^{'}, \ldots, x_n^{'})$ can be accurately mapped to $(y_1^{'}, \ldots, y_n^{'})$. This is a guided search operated by a function called *loss function*. Generally speaking, a loss function, $L(y, y')$, can be any function (with a lower bound not equal to negative infinity) that takes a predicted output y' and the corresponding gold output y as input and produces a scalar value indicating the discrepancy between the prediction and the ground truth. However, in most cases loss functions that do not require complex gradient calculations are preferable for practical reasons. The general form of the corpus-wise loss function can be defined as follows:

$$\frac{1}{n}\sum_{i=1}^{n} L\left(f\left(\boldsymbol{x}_i;\Theta\right),\boldsymbol{y}_i\right) \tag{2.1}$$

where n denotes the number of training examples and L() is the loss on ith instance. Depending on the type of the task (see below) that the neural network is performing, the implementation of loss function may vary. For example, while cross entropy loss is the most common loss function for classification problems, for regression, mean squared error is preferred[1]. The $y = f(x;\Theta)$ function's Θ parameters are then adjusted, as shown below in Eq. 2.2, to minimize the loss over the training instances.

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\frac{1}{n}\sum_{i=1}^{n} L\left(f\left(\boldsymbol{x}_i;\Theta\right),\boldsymbol{y}_i\right) \tag{2.2}$$

One shortfall of Eq. 2.2 is the lack of a constraint that prevent model from capturing the noise in the dataset which will help to further reduce the loss, but at the same time, also make the model fail to generalize on unseen data. One of the most common ways to avoid this phenomenon – which is also called *overfitting* – is to modify the equation Eq. 2.2 above by adding a second term called *regularization term*, $R(\Theta)$:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\frac{1}{n}\sum_{i=1}^{n} L\left(f\left(\boldsymbol{x}_i;\Theta\right),\boldsymbol{y}_i\right) + \lambda R(\Theta) \tag{2.3}$$

Over the years, many different regularization methods have been

[1] For more detailed discussion on individual loss functions in the context of neural networks, see Goodfellow et al. (2016)

developed, such as weight decay (Hanson and Pratt, 1988), Dropout (Srivastava et al., 2014) or stochastic depth (Huang et al., 2016) that are developed to constraint the optimization problem above. The most apparent common feature of these methods is the fact that they take the parameters θ as input and return a scalar that reflect the complexity of the learned function which needs to remain low to avoid overfitting. While constraining the effective capacity of a given model is one of the most popular approaches to reduce overfitting, there are also other ways, such as data augmentation (Zhang et al., 2017) which is a data-driven regularization strategy that fight against overfitting by artificially increasing the number of training samples. For more details on individual regularization methods, we refer the reader to Peng et al. (2015) and Moradi et al. (2020).

There are two main types of supervised learning, namely *regression* and *classification*. Regression problems are concerned with mapping inputs to outputs where the output is a continuous real number, i.e. $y_i \in \mathbb{R}$. In classification problems, on the other hand, the output is one of a finite set of discrete labels, i.e. $y_i \in (1, ..., C)$ where C denotes the number of possible classes. Classification problems can be further categorized: Classification tasks with two class labels (C=2) are referred as *binary classification* and more than two classes are called *multi-class classification*. In the latter one, if an instance is associated with multiple labels (i.e. classes are not mutually exclusive), then it is called *multi-label classification*.

2.1.2 Typology of NLP Classification Problems

Classification tasks account for a significant part of NLP research. One way of grouping these tasks into categories, as shown in Goldberg (2017), is based on the properties of the subject being classified:

- **Text Classification** is a general classification problem type, where the goal is, for given a single input text which can be a phrase, sentence or even a document, to predict a label for it.[2] The important thing to keep in mind is that in text classification tasks, the label is assigned to the complete input, not the parts of it. The tasks that fall under this category can further be grouped into two. The first group deal with classification of words in isolation. They involve addressing a number of issues, such as identifying the language the word written in (Hammarström, 2007; Gottron and Lipka, 2010); checking whether a given word is correctly spelled (Farra et al., 2014); deciding whether the input word is a complex or a simple word (Malmasi and Zampieri, 2016), and so on. Note that, as words seldom appear in isolation in NLP (Goldberg, 2017), this group covers only a small part of all text classification problems. The tasks in the second subgroup of this category deal with classification of text that is longer than a word. While there are many different tasks in this subgroup, some of the most noticeable ones are sarcasm detection (Zhang et al., 2016); sentiment classification (Liu, 2012); hate speech classification (Schmidt and Wiegand, 2017), authorship attribution (Mekala et al., 2018), and topic classification (Xia et al., 2019).

[2] In case of multi-label classification y might contain multiple labels.

- **Sequence Labeling** covers, similar to text classification, a variety of classification tasks. The main difference between this and the first category is that here, for a given input sequence, the goal is to assign a class or label to each token in a given input sequence, instead of assigning a label to the complete sequence. These tasks include, but not limited to, part-of-speech tagging (Kanakaraddi and Nandyal, 2018); morphological tagging (Cotterell and Heigold, 2017); named-entity recognition (Yadav and Bethard, 2018) and text chunking(Liu et al., 2018).

2.1.3 Neural Architectures for Classification in NLP

Over the years, a range of classification algorithms has been developed, including geometric models (e.g., nearest-neighbour classifiers, SVMs), logical models (e.g., decision trees) and probabilistic models (e.g., naive bayes classifiers, neural networks) (Flach, 2012). With the recent advances in computational power, as well as greater availability of big data in the last couple of years, neural networks has led to a revolution in our capabilities to process and analyze large sets of complex data and yield new state-of-the-art results in a very wide range of NLP tasks (Graves, 2013; Hinton et al., 2015; Jozefowicz et al., 2016)

At the high level, most of the neural architectures that have been used to tackle the classification tasks composed of three components: a) an embedding module which maps words into their distributed representations, b) a context encoder module which extracts contextual features, and c) an inference module which predicts labels or generate optimal label sequence as output of the model.

Embedding module. In all models, tokens that together form the textual input must be represented in a form that can be understood by the model. One-Hot Encoding is one simple approach for encoding textual data in a way which is amenable for use by classifiers. In this approach, each word is represented by a sparse vector in the size of the vocabulary, with 1 in the entry representing the word and 0 in all other entries. While it is very simple to construct, there are two main disadvantages of this approach : 1) It results in high-dimensional, sparse (mostly zero) data which doesn't work well with neural models. 2) It does not convey any similarities between words. The word "cat" is as dissimilar to word "lion" than it is to word "plane". That means the model cannot reuse information it already learned about cats for the much rarer word lion. Word embeddings are an alternative to one-hot encoding by representing each word (or token in general) with a real-valued vector. Unlike one-hot encoding where all token vectors are equidistant, tokens that are close in the embedding space are expected to be similar in meaning (Mikolov et al., 2013b).

Word embedding models are quite closely intertwined with language models which generally try to compute the probability of a word w_t given its n-1 previous words, $p\left(w_t \mid w_{t-1}, \cdots w_{t-n+1}\right)$. By applying the chain rule together with the Markov assumption, we can approximate the probability of a whole sentence or document by the product of the probabilities of each word given its n previous words:

$$p\left(w_1, \cdots, w_T\right)=\prod_i p\left(w_i \mid w_{i-1}, \cdots, w_{i-n+1}\right) \tag{2.4}$$

The common types of language modeling techniques involve n-gram and neural langauge models. Bengio et al. (2000) is the first to propose learning word embeddings within a neural network language model (NNLM). The goal of the NNLM model of Bengio et al. (2000) is to predict the next word based on a sequence of preceding words using a simple feedforward neural network.

In 2013, Tomas Mikolov released word2vec (Mikolov et al., 2013a), which led to word embeddings becoming very popular in NLP. As opposed to the NNLM model of Bengio et al. (2000), in which word embeddings are produced as a by-product, the word2vec algorithms are specifically designed to generate word embeddings. There are two different training methods to obtain Word2Vec embeddings: Continuous bag-of-words (CBOW) and Skip-gram. With CBOW, the goal is to predict a middle word based on its surrounding words. The skip-gram approach aims to do the exact opposite: It tries to predict the surrounding words given a current word. Another highly popular word embedding method is Glove (Pennington et al., 2014). The key differences between Glove and Word2Vec are 1) while Word2Vec is a predictive model, GloVe is a count-based model and 2) as opposed to Word2Vec which only uses local information, GloVe does take into account the global information (via a global co-occurance matrix) besides the local information to obtain the embedding vectors.

One common problem that occurs with many word embedding, including Glove and Word2Vec, is handling out-of-vocabulary words (words that do not appear in training vocabulary or document). To solve this issue, Bojanowski et al. (2017a) proposed a novel method based on the skip-gram architecture. In this method, called FastText, Each word is represented as a bag of character n-gram and each word embedding of each word is obtained by a sum of vectors, with each vector representing an n-gram.

The above word embedding methods are some of the most popular *static* word embedding methods. Static word embeddings generate, regardless of the context the word being used, only a single vector for each word meaning that these methods do not model polysemy (phenomenon where a word form can have multiple senses). Recently, however, many approaches for learning contextual word representations have been proposed. As opposed to static word embeddings, these *contextualized embedding* techniques can compute an embedding vector for a word by taking the context of the word into account, leading to a representation that better match the specific use of the word in a sentence.

Peters et al. (2018) propose one of the first deep neural network based contextualized embedding method, called ELMo. In this model, contextualized embeddings are extracted from a bidirectional language model. Two multi layer LSTMs are applied to the sentence in both directions to encode the left and right context independently. Then, at each layer, the hidden states of left-to-right and right-to-left LSTMs are concatenated, obtaining N hidden representations for a sequence of length N. The idea of ELMo was later extended by GPT and BERT in which LSTMs are replaced by uni-directional and bi-directional transformer-based (Vaswani et al., 2017) models respectively. BERT and GPT were just the starting point for the development of many variations and many others followed this approach (i.e. using transformers) to obtain contextualized embeddings, for which we refer the reader to Liu et al. (2020).

Encoder module. Recurrent Neural Networks, Convolutional Neural Networks and Transformers are the three common architectures used as encoder module in NLP tasks. *Recurrent Neural Networks* (RNNs) have been one of the most widely used architectures as encoder module for classification tasks in NLP. The model takes the current time step's input and the previous time step's hidden state and creates a hidden state and optionally an output. Depending on the nature of the classification task, the hidden state from the last time-stamp or all of the hidden states from each step can be used as the representation of the input sequence.

RNNs cannot capture long-term dependencies of very long sequences, which appear in many real applications, due to the gradient vanishing and explosion issue. LSTM is a variation of RNNs designed to better capture long-term dependencies. LSTM layer consists of a memory cell, which remembers values over arbitrary time intervals, and three gates (input gate, output gate, forget gate) that regulate the flow of information in and out the cell.

There have been different attempts to improve RNN-based models for various classification tasks by capturing richer information. Some of the prominent variants are: 1) Tree-LSTM model (Tai et al., 2015), an extension of simple LSTM to tree-structured network typologies which leads to richer syntactic representations of the input; 2) TopicRNN proposed by Dieng et al. (2016) integrates the capabilities of latent topic models so that it can capture long-range dependencies more accurately; 3) The Disconnected Recurrent Neural Network (DRNN) (Wang, 2018) which incorporates position-invariance into RNN by limiting the maximal transmission step length in RNN to a fixed value. The list above is not exhaustive and a reasonably detailed survey on RNN-based model variants can be found in the works of Kowsari et al. (2019).

Convolutional Neural Networks(CNNs) are - although it is originally built for computer vision tasks - another popular neural deep learning architecture in NLP, especially for document classification. A standard CNN architecture is composed of three different layers: 1) Convolutional layers which are used to obtain local features; 2) Pooling layers where local features are aggregated; and 3) Fully connected layers which form the last few layers in the network and drive the final classification decision.

While many CNN-based NLP models have been proposed over the years, TextCNN (Kim, 2014) is one of the first and most popular CNN-based classification approach in NLP. It is a relatively small model which has one convolutional layer with kernels of different sizes, followed by max pooling, and a fully-connected layer. Kim et al reported that TextCNN improves upon the state of the art on sentiment classification. Later, due to its high performance, small number of parameters, and fast training speed, it has also been tested by other researchers on various classification tasks such disease detection (Yang et al., 2018), malicious user detection (Hong et al., 2018) and fake news detection (Bsoul et al., 2022).

Compared to RNNs in which there is a dependency between subsequent steps, CNN based methods are faster since the computations in CNN can (mostly) happen in parallel. Though relatively high efficiency, a major disadvantage of CNNs is that due to the convolutions and pooling operations in the model, it has difficulties in capturing word order information which makes pure CNN-based methods to less suitable for sequence labeling tasks such as POS tagging. (He et al., 2020)

The Transformer is a novel encoder–decoder based architecture introduced by Vaswani et al. (2017). As previously mentioned above, one of the main disadvantages of RNNs is that they process the input sequentially. While CNNs suffer less from this – compared to RNNs – the computational cost of capturing relationships between words still grows as the input sequence length increases, similar to RNNs. Transformers overcome this limitation by processing the entire input at the same time, without any notion of order. To capture sequential information, these models add a special vector called positional encoding to each input embedding whose purpose is injecting information about the relative positioning of words.

The layers in the Transformer model consist of two main components; namely, a multi-head attention layer followed by a feed forward neural network. The multi-head attention sub-layer is the main component which lets the encoder look at other words in the input sentence (without relying on recurrence), to capture interdependencies between words, while encoding a specific word. As the first step in multi-head attention sub-layer in the encoder side, the embeddings of each input token are multiplied by three weight matrices (which are learnt during the training process) to create three word representation vectors called Query (Q), Key (K) and Value (V), as shown in Eq. 2.5:

$$Q = X \cdot W_Q \quad K = X \cdot W_K \quad V = X \cdot W_V \tag{2.5}$$

where W_Q, W_K, $\mathrm{W}_V \in \mathbb{R}^{\mathrm{d\ x\ k}}$; $\mathrm{X} \in \mathbb{R}^{\mathrm{N\ x\ d}}$; N is the number of tokens in the input; *d* and *k* are hyperparameters defining embedding lengths. After generating the Q, K and V representations, a self-attention score for each input token is calculated by multiplying the query vector of the current token with the key vectors from other input tokens. These attention scores can be interpreted as the alignment score between each token and the other tokens in the input and indicates the relative importance between them. The scores are then scaled (for stability reasons) and passed through a softmax function, so that they are all positive and add to 1. Eventually, the representation of each word is obtained by multiplying the scaled term with the Value vector, as shown in Eq. 2.6.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{k}}\right) \cdot V \tag{2.6}$$

Multi-head attention sub-layers at the decoder side operate similarly. The key differences are 1) In the decoder, these sub-layers are only allowed to attend to earlier positions in the output sequence. 2) There is an additional multi-head attention layer called "Encoder-Decoder Attention" which works just like regular multi-head attention, except it takes the Keys (K) and Values (V) matrices from the output of the encoder stack.

The fact that transformers can be trained on enormous amounts of data in far less time than other sequential architectures such as RNNs led to the development of large-scale Transformer-based Pretrained Language Models (PLMs), which operate under a pretrain-finetune paradigm: First, models are pretrained over a large text corpus to learn contextual text representations. As PLMs are very large models, pretraining of PLMs are highly computationally expensive[3].

[3] e.g., The training of a GPT-3 is estimated to cost $4m (Wang et al., 2021b)

Therefore, this step is generally performed by large industry research labs, many of which release their models in order to save others the costs of retraining them. Second, pretrained models are fine-tuned on labeled task-specific datasets, whose size are significantly smaller than pre-training datasets, to optimize for task-specific accuracy on a downstream task. This paradigm become the new state-of-the-art in a variety of tasks in NLP. Consequently, many PLM variants have been proposed over the years. Below, we mention only the most popular ones and refer the readers to work by Qiu et al. (2020) for a more detailed survey on PLMs.

PLMs can be grouped into two categories, autoregressive and autoencoding PLMs. OpenGPT (Radford et al., 2018) is one of the earliest autoregressive PLMs which is made up of unidirectional decoder stacks from the Transformer architecture. Similarly, BERT (Devlin et al., 2019) is the one of the first approaches on the autoencoding PLMs side. Unlike OpenGPT which predicts words based on previous predictions, BERT is trained using the masked language modeling task. There have been numerous works on improving BERT. Some of the most notables are: RoBERTa (Liu et al., 2019) which is more robust than BERT because it is trained with larger mini-batches on significantly more training data; ALBERT (Lan et al., 2019) reduces the memory usage and training time of BERT; Distill-BERT (Sanh et al., 2019) a smaller, faster and cheaper BERT thanks to the knowledge distillation technique used during pre-training; and SpanBERT (Joshi et al., 2020) which extends BERT to better predict text spans.

Inference module. The inference module is the outermost module which takes the representations from encoder module as input, and performs prediction. Sigmoid, Softmax and Conditional Random Fields (CRF) are three most popular methods used in neural classification models in NLP.

The sigmoid function[4] is a continuous, monotonically increasing function defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

It accepts any real value as input and produces values between 0 to 1. The greater the input, the closer the output to 1, and the smaller the input, the closer the output to 0. It is especially useful for binary classification tasks (where the number of classes is equal to two) or when the categories are not mutually exclusive (i.e. multi-label classification tasks), as the sigmoid function predicts independent probabilities for each class.

Softmax is a generalization of sigmoid function which turns a N-dimensional vector with real values into another N-dimensional vector in which each element is between 0 and 1 and the sum of all elements equals 1. As a result, output of softmax function can be interpreted as a probability distribution over classes. It is defined as:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{2.8}$$

and has been widely used in neural networks for multi-class clas-

[4] We acknowledge that the sigmoid function can also be used in the earlier layers of the network to introduce nonlinearity. In this discussion we focus its role as an output unit.

sification tasks.

Conditional Random Fields (CRF) are, in general, Markov Random Fields that are globally conditioned on observations. Over the years, many versions of CRFs have been developed for use in NLP and other branches of AI. Here, we focus on linear-chain CRFs (Lafferty et al., 2001), but the generic idea described here can be extended to CRFs of any structure. Unlike sigmoid and softmax, which can be used for all the classification task categories discussed in Section 2.1.2, CRFs are typically used in NLP models when the output is a sequence of classifications (i.e. the sequence labelling category). In sequence labeling tasks, such as POS Tagging or NER, the correct label to each word often depends on the neighboring labels. For instance, the correct POS tag of a word can sometimes be deduced from the POS tag of the adjacent words. As opposed to above methods in which predictions are made independently for each position, CRF models have been proven to be powerful in learning the strong dependencies across output labels, thus most of the neural network-based models for sequence labeling employ CRF as the inference module. Specifically, let $\mathbf{Z} = [\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \ldots, \hat{\mathbf{z}}_n]$ be the output of context encoder of the given sequence $\hat{\mathbf{x}}$), the probability $\Pr(\hat{\mathbf{y}} \mid \hat{\mathbf{x}})$ of generating the whole label sequence

$$\Pr(\hat{\mathbf{y}} \mid \hat{\mathbf{x}}) = \frac{\prod_{j=1}^{n} \phi\left(y_{j-1}, y_j, \hat{\mathbf{z}}_j\right)}{\sum_{y' \in \mathbf{Y}(\mathbf{Z})} \prod_{j=1}^{n} \phi\left(y'_{j-1}, y'_j, \hat{\mathbf{z}}_j\right)} \tag{2.9}$$

where $Y(Z)$ is the set of possible label sequences for Z, and ϕ is s a potential function defined as the summation of transition (scores representing how likely is y_{j-1} followed by y_j) and emission (scores representing how likely is y_j given the input z_j) features at each time step.

2.1.4 Evaluation Metrics

Evaluation of NLP models can be classified into intrinsic and extrinsic methods. While extrinsic evaluation is aimed at evaluating models outputs based on their impact on the performance of other NLP models, in an intrinsic evaluation, quality of outputs of NLP models is evaluated against predefined ground truths (Clark et al., 2012).

The evaluation metric to use depends on the type of NLP task. However, for most of the classification, a matrix called the confusion matrix can be constructed which shows each combination of the true and predicted classes: A true positive (TP) is an outcome where the model correctly predicts the positive class. Similarly, a true negative (TN) is an outcome where the model correctly predicts the negative class. A false positive (FP) is an outcome where the model incorrectly predicts the positive class and a false negative (FN) is an outcome where the model incorrectly predicts the negative class. From the confusion matrix, different evaluation metrics can be calculated. Some of the most common intrinsic metrics to evaluate classification models in NLP are as follows:

- Accuracy: Accuracy is the simplest metric and can be defined as the number of test cases correctly classified divided by the total number of test cases:

$$\text{Acc.} = (\text{TP}+\text{TN})/(\text{TP}+\text{FN}+\text{TN}+\text{FP}) \tag{2.10}$$

It can be applied to most generic problems but is not very useful when it comes to unbalanced datasets.

- Precision: The ratio between the number of positive samples correctly classified to the total number of samples classified as positive:
  $$\text{Pre.} = \text{TP}/(\text{TP}{+}\text{FP}) \tag{2.11}$$
  where a perfect score of 1 indicates that every sample the model identified as belonging to a specific class did in fact belong to that specific class.

- Recall: The ratio between the number of positive samples correctly classified as positive to the total number of positive samples:
  $$\text{Rec.} = \text{TP}/(\text{TP}{+}\text{FN}) \tag{2.12}$$
  where a perfect score of 1 would indicate that the model correctly classified all the samples belonging to that class.

- F-Score: The harmonic mean of precision and recall. F-score combines the precision and recall of a classifier into a single metric:
  $$F_\alpha = \left(1+\alpha^2\right)\frac{\text{Pre.} * \text{Rec.}}{(\alpha^2 * \text{Pre.}) + \text{Rec.}} \tag{2.13}$$
  By varying α in the equation, one can put more emphasis on recall.

2.1.5 Bias and Fairness in NLP

In this section, we provide a brief overview of research on bias in NLP systems, where a system is defined as biased if it systematically discriminate against certain individuals or groups of individuals in favor of others (Mehrabi et al., 2021). We first provide a brief overview of research on detecting various unintended bias types in word embeddings. Next, we continue with an overview of bias analysis for downstream tasks. Finally, we discuss the main strategies for mitigating bias. Note that there is an enormous body of literature on bias and fairness in NLP and we can only touch on the main directions in this field. We refer readers to Mehrabi et al. (2021) and Bansal (2022) for more detailed reviews.

Bias in embeddings. As pointed out above, almost all state-of-the-art NLP systems use corpus-derived embeddings. These embeddings were the starting point for a lot of work on bias in NLP. Bias in embeddings is generally shown by comparing embeddings for two sets of previously established, e.g., gendered (male and female) words (e.g. *man, woman*).

Bolukbasi et al. (2016) propose an approach to investigate gender bias present in word2vec embeddings by constructing a gender subspace using word embedding vectors of a set of binary gender pairs (e.g. she/he, woman/man, etc.). Then, gender bias of a word embedding is defined by the size of the component of the word embeddings that project onto the above-mentioned gender subspace. The larger a word's projection is on gender subspace, the more biased it is. Others built on this approach later on, for example Manzini et al. (2019) extended it to find non-binary gender bias in pretrained Word2Vec embeddings. Recently, however, this method was found to be an imperfect metric of bias by Gonen and Goldberg (2019). Specifically, they showed that word embeddings representing words with similar biases still cluster together even after the projections of word embeddings representing gender-neutral words onto the gender subspace have been removed.

As an alternative, Caliskan et al. (2017) introduce the WEAT benchmark, which is an adaptation of the Implicit Association Test (IAT) in which response times were recorded when subjects were asked to match two concepts to quantify societal bias in sociological research (Greenwald et al., 1998). WEAT uses word similarities between targets and attributes instead of the response times. It computes the difference in relative cosine similarity between two sets of attribute words A and B (e.g. male names and female names) and two sets of target words A and B (e.g. science and art). Caliskan et al. (2017) applied WEAT to the Glove and Word2Vec pre-trained word embeddings and found that both exhibit gender bias. Later, WEAT was extended by introducing a multilingual (Lauscher and Glavaš, 2019) and cross-lingual (Lauscher et al., 2020) versions of it, too.

While much of the attention has been dedicated to identify and mitigate gender bias in word embeddings, there are also various works which go beyond gender and investigate other bias types in word embeddings. Some of the noticeable examples include Garg et al. (2018) who analyzed ethnic biases in historical embeddings covering 100 years of language use; Swinger et al. (2019) who showed that word embeddings of names reflect broad societal biases that are associated with those names, including race, gender, and age biases; and Rozado (2020) who showed that most of the famous pre-trained word embeddings also display biases based on age and religion.

Bias in NLP systems. At the system level, bias has been investigated in applications including (but not limited to) named entity recognition (NER), Machine Translation (MT), Sentiment Analysis, and Coreference Resolution. Kiritchenko and Mohammad (2018) examined 219 sentiment analysis systems and found that a majority exhibits gender and race biases. Mehrabi et al. (2019) reported that NER models recognize male names with higher recall compared to female names. Rudinger et al. (2018) and Zhao et al. (2018a) showed that coreference resolution systems perform unequally across gender groups by associating occupations (such as doctor and engineer) more with men and others (like nurse) more with women. Similarly, Stanovsky et al. (2019) found that both commercial and academic MT models are at risk of generating translations based on gender stereotypes rather than the actual source content.

Bias in systems is usually measured by using benchmarks datasets for specific tasks with a one-factor design which are created to be as balanced as possible while varying the levels of the bias variable. Examples include WinoBias (Zhao et al., 2018a) and WinoGender (Rudinger et al., 2018), two benchmarks for gender bias in coreference resolution which contrast "pro-stereotype" cases (the correct antecedent of a pronoun is conventionally associated with the pronoun's gender) and "anti-stereotype" cases (opposite situation); GAP (Webster et al., 2018), another dataset for coreference resolution task, which is larger than existing Winograd schema datasets, and consists of examples from naturally occurring Wikipedia text; and the Equity Evaluation Corpus (EEC, Kiritchenko and Mohammad (2018)), developed to analyze gender and race bias in sentiment analysis and consists of 11 sentence templates instantiated into 8,640 English sentences for four emotions: Anger, joy, fear and sadness .

Bias is then quantified by measuring the differences in performance between these levels. Sometimes, but not always, the differences are subsequently tested for statistical significance, (e.g. t-test is used in Kiritchenko and Mohammad (2018)). To our knowledge, almost no studies on system-level bias have considered covariates, nor computed effect sizes, which makes them vulnerable to the criticisms outlined in Chapter 10.

Bias Mitigation. There are two main families of methods to mitigate bias at the representation level. Approaches from the first family perform modifications on the training sets, on which embeddings are obtained. One of the most common training set modification way is to add examples that balance with respect to an attribute (e.g., gender or race). Thus seeking to make the data represent that given attribute in a less biased way (Zhao et al., 2019; Hall Maudslay et al., 2019). Another popular modification technique is to remove textual features which might be an indicator of a demographic feature of a person from the data, such as pronouns or named entities so that the model cannot learn differences associated with them (De-Arteaga et al., 2019; Heindorf et al., 2019).

Approaches from the second family mitigate bias by transforming learned embeddings according to some balancing objective. One of the first examples of this approach is Zhao et al. (2018b), who proposed to debias GloVe vectors by adding a constraint to its objective function such that the gender-related information is confined to a sub-vector. During optimisation, the distance between sub-vectors are maximised, while simultaneously minimising the GloVe objective. This idea of extending the objective function with a regularization term in order to mitigate bias in embeddings has been later used by other researchers to reduce gender bias in different static word embedding algorithms (Bordia and Bowman, 2019; James and Alvarez-Melis, 2019) as well as contextualised word embeddings (Kaneko and Bollegala, 2021).

At the system level, Zhao et al. (2017) proposed to constrain model predictions to follow a distribution from a training corpus. Rather than constraining the output, some of the previous work such as Elazar and Goldberg (2018) and Kumar et al. (2019) used a technique called adversarial debiasing (Zhang et al., 2018) to remove unintended bias from latent representations. In adverserial debiasing, two models (main and adversary) with a common encoder component are trained simultaneously. While the main and adversarial classifiers are trained to predict the main target and bias attribute respectively, the encoder is trained to make adversarial classifier fail. Adjusting the loss function is another popular system level approach for bias mitigation. For instance, Qian et al. (2019) introduces a new term to the loss function to equalise the probabilities of male and female words in the output, and Jin et al. (2021) introduce a regularization term which reduces the importance placed on surface patterns.

2.2 Computational Social Science

Computational Social Science (CSS) is a very active, rapidly growing inter-disciplinary field that aims to develop computational tools to study long–standing questions in the social sciences. While CSS covers a wide range of domains, including but not limited to political science, public health, social media, economics, psychology, sociology, sociolinguistics, we restrict ourselves in the upcoming sections to discuss work directly adjacent to our own research.

2.2.1 Political Claims Analysis And Discourse Network Analysis

Political discourses evolve through political claims made by various actors within the public sphere, and by the specific interests, policy ideas, and values they propagate (Münnich 2011). For this reason, analysis of claims which describe the main patterns of public debates has always been a central issue in the context of political and social sciences. Previous work in political science has performed political claims analysis on a wide range of topics related to politics such as energy (Haunss et al., 2013), education (Fairclough, 2013) and economy (Temple et al., 2016) using variety of data genres such as manifestos (Helbling and Tresch, 2011), newspapers (Zamponi and Bosi, 2016) and social media data (Bilbao-Jayo and Almeida, 2021).

There is a recent innovation called Discourse Network Analysis (DNA) (Leifeld, 2009) which is an approach that combines political claims analysis with network science, allowing to investigate claims through a network perspective and visualize the structure of policy debates. DNA has been used by numerous researchers, such as Leifeld and Haunss (2012) to analyse discourse coalitions in the European conflict over software patents; Rinscheid et al. (2020) to trace political discourses around the 2011 Fukushima nuclear accident in Canada, Germany, and Japan; and Blokker et al. (2021) to capture the temporal dynamics of the political debate on immigration in Germany in 2015.

The first step in DNA is the qualitative or semiautomatic coding of statements in a text corpus. Then using these identified elements various DNA network types such as affiliation network, actor congruence network or concept congruence network can be generated. Affiliation network is the main entry point for DNA. It is, as depicted in Figure 3(b), a bipartite graph with two types of nodes: 1) Actor nodes: $A = [a_1, a_2, \dots a_m]$ and 2) Concept nodes: $C = [c_1, c_2, \dots c_n]$. Actor and Concept nodes are connected via edges which indicate some sort of relationship between the two entities, either in the form of agreement or disagreement: $R = [r_1, r_2]$. Lastly, since affiliation networks can be repeatedly observed, they also have temporal attributes $T = [t_1, t_2 \dots t_k]$, where t_k denotes the kth time step. Overall, an affiliation network can be formally defined as follows:

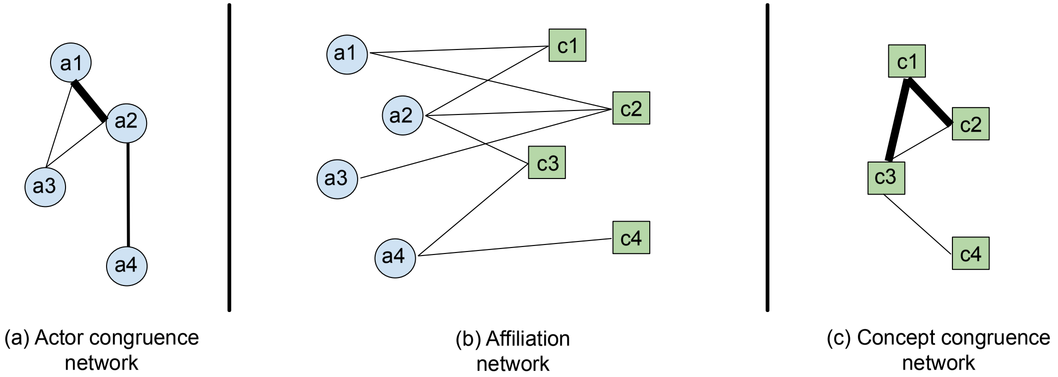
$$\begin{gathered}\left(G^{aff}_{r,t}\right) = \left(A, C, E^{aff}_{r,t}\right)\\ \left\{a, a'\right\} \notin E^{aff}_{r,t} \wedge \left\{c, c'\right\} \notin E^{aff}_{r,t}\end{gathered} \tag{2.14}$$

where a and a' are two different actors; c and c' are two different concepts and $E^{aff}_{r,t}$ refers to the set of edges in the affiliation graph G^{aff} at time t and for relation r. Alternatively, an affiliation network can also be represented by an mxn matrix $X_{r,t}$ for each relation and time point where actors are placed to rows and concepts are placed to columns.

An actor congruence network is a network that only consists of actor nodes which can be created from the affiliation network by multiplying the affiliation matrix by its transpose (i.e., $X_{r,t} * X_{r,t}^T$). Edges in the resulting network connect actors employing same concepts. Thus, an actor congruence network can be useful for detecting actor coalitions, groups with similar policy preferences. Concept congruence network is another network type that is frequently used in DNA, and it can be constructed from the affiliation network by multiplying the transpose of affiliation network with the matrix itself (i.e., $X_{r,t}^T * X_{r,t}$). Similar to the actor congruence network, concept congruence network also contains one type of node, which is the concept node. The two nodes in this type of network are connected when the concepts represented by those nodes are used by the same actor in the same way, and the edge weight between two nodes is equal to the number of actors that refer to the two concepts. Concept congruence networks are useful for observing concepts clusters which can be considered as coherent storylines (Leifeld, 2017). Actor and concept congruence network examples are shown in Figure 3(a)-(c).

2.2.2 Social Media Analysis

Social media is a general term which describes internet services that allow users to interconnect and share information with each other (Kaplan and Haenlein, 2010). There are many different social media channels, such as Twitter for short messages, Facebook for social connections, Instagram for primarily at sharing pictures and TikTok for video sharing. With tens of millions people posting and sharing



(a) Actor congruence network

(b) Affiliation network

(c) Concept congruence network

Figure 3: Example actor (left), affiliation (center), and concept (right) networks. Circles indicate actors and squares indicate concepts. In affiliation network, the in-degree of a concept node indicates the number of times a concept is mentioned and the out-degree of an actor node indicates the number of concepts made by that actor. The edge width in an actor and concept congruence networks resemble the edge weight which is proportional to number of concepts referred by the actors and number of actors referring to both concepts respectively.

information about their lives, a huge quantity of knowledge is generated on social media platforms. Consecutively, social media analysis has become a very active line of research in recent years with many different applications related to various tasks in NLP and Computer Vision. While we can't acknowledge all here due to the large number of studies, and we refer readers to Natural Language Processing for Social Media book by Farzindar and Inkpen (2017) and a survey article on ML algorithms for social media analysis by Balaji et al. (2021) for that, we provide an overview of some of the most well known and widely used NLP tasks in social media analysis.

One of the most popular tasks in social media analysis is author profiling, a task of predicting authors' attributes based on the text that they have written. The most prominent author profiling task is gender classification (Kucukyilmaz et al., 2006; Li and Dickinson, 2017; Sezerer et al., 2018), other tasks include the prediction of age (Morgan-Lopez et al., 2017), race and region of origin(Pennacchiotti and Popescu, 2011; Chen et al., 2015). Besides demographic features, there is also work on other attributes such as personality types (Plank and Hovy, 2015) or mental health (Parapar et al., 2021) of the author. Emotion analysis is another well-studied task in the social media domain. It entails the process of identifying the underlying emotions expressed in textual data. Previous studies proposed methods to determine who is feeling what emotion (Islam et al., 2019) and towards whom (Mohammad et al., 2014; Campagnano et al., 2022) on social media data. Apart from these, there has been also considerable work on identifying abusive language that is specifically directed towards a particular group or person (Schmidt and Wiegand, 2017; Garibo i Orts, 2019), and detecting and analysing disinformation, hoaxes and fake news on social media (Majithia et al., 2019; Atanasova et al., 2020).

A topic of general importance that we will return to later in this thesis is demographic characteristics. It has been shown by previous work that authors' demographic factors such, such as age, gender and nationality affects their language use significantly. Consecutively, these factors have always been an important aspect in NLP research, both in the context of social media analysis and in general. There are two main stance towards demographic attributes. The first group of work, as we have already discussed above in Section 2.1.5, concerns removal of demographic attributes from text and models in order to avoid unintended biases.

There is a second line of work, especially in social media domain where authors' attributes can be derived from publicly available information such as written profile bio, that studies how the demographic factors can be used to improve performance of the classifiers. For instance, Hovy (2015) reported that age and gender information of authors can be used to improve performance of review classification model. Similarly, Volkova et al. (2013) and Lynn et al. (2017) showed that incorporating demographic factors into the models yields better sentiment classification and POS tagging results on Twitter data respectively. Machine translation is another task where incorporating gender and age traits has been proven beneficial (Rabinovich et al., 2017). Apart from improving performance on fundamental NLP tasks as outlined above, recent research has demonstrated that including demographic factors can also improve performance of classifiers on more challenging tasks such as Suicide Risk Assessment (Matero et al., 2019).

3 DebateNet

In this chapter, we describe the DebateNet dataset (Lapesa et al., 2020) a high-quality policy debate dataset on the topic of immigration in Germany in 2015. The year of 2015 was special as the topic of immigration has increasingly moved into the center of public debates in Germany as a result of dramatic increase in the number of refugees attempting to enter Europe from Africa and the Middle East mostly due to the increase in internal conflicts in these countries. Politicians and public figures responded to the growing numbers of refugees with rapidly changing policy proposals, which were reported and discussed in the media outlets. Among those outlets, newspapers are extremely valuable for political scientists who are interested in studying dynamics of policy debates to understand decision-making processes as they can provide a fine-grained representation of debate, both at the level of content (extensive reports of the positions of politicians and parties) and at the level of time (multiple articles per day). DebateNet has been created exactly for this purpose in mind: It provides a fine-grained picture of the public discourse concerning the domestic debate on immigration in Germany in 2015.

DebateNet is a large annotation project conducted under the scope of project MARDY[1], a collaborative project between University of Stuttgart and University of Bremen. The annotation studies took place at the University of Bremen and took roughly three years, involving six political science students and two domain experts as annotators, and targets the German side of the debate on the refugee crisis. The development of the annotated corpus can be seen in the size of the different releases over time: The first version has been released in Padó et al. (2019) and it contains 982 Claims in 764 different text spans. Second version of DebateNet (Lapesa et al., 2020) includes 1815 textual spans corresponding to 2274 distinct claims and the latest version, as of March 2022, released in Blokker et al. (2021) contains 3442 text spans translating into 4417 individual claims.

Many of the experiments in this thesis have been conducted on DebateNet; therefore, we provide in depth description of the latest version of DebateNet, i.e. (Blokker et al., 2021), in the rest of this chapter. Details of the specific version of the dataset used in a particular experiment is provided in the corresponding chapter.

3.1 Source Corpus & Article Selection

The source corpus consists of newspaper articles from the German quality newspaper *die Tageszeitung (taz)* which is a major national German daily newspaper, founded in 1978 as an alternative, self-governing newspaper project[2]. It is perceived as the most left-oriented major German newspaper, but can still be assumed to por-

[1] The work presented in this thesis is also developed as part of MARDY project.
[2] `https://en.wikipedia.org/wiki/Die_Tageszeitung`

tray both sides of the relevant discussion. As there are around 38000 articles published in TAZ in 2015 (Blokker et al., 2021), and it was infeasible for the annotators to read all of them, the articles related to the migration topic needs to be selected. The selection process has been done in three steps as follows:

- First, a keyword-based approach has been used to select articles relevant to the topic at hand.
- Next, in order to find the articles missed by the keyword-based approach, a binary classifier has been trained on the articles found in the first step. As the goal is to find as many migration related articles as possible, a binary classifier has been optimized on recall during training.
- Finally, annotators flagged false positive articles for being off topic.

After completion of article selection process, the annotators start working on the selected articles.

3.2 Annotation

In this section, we describe annotation process of newspaper articles. Note that although the dataset and codebook provided at the time of annotated were in German, we use English translations provided by domain experts for the examples and claim category names in this thesis[3].

[3] The original German codebook can be found at `https://clarin09.ims.uni-stuttgart.de/debatenet/MARDY_Codebook_Mig_german.pdf`

Annotation follows a procedure successfully used by Haunss et al. (2013) in the analysis of the German nuclear phase-out debate: It is carried out in double, independent annotation by trained student research assistants, and it targets multiple levels, with different degrees of abstraction and complexity for the annotator. First, the annotators have to identify the textual spans containing claims which are demands, proposals, or criticisms that are supported or rejected by actors and can be categorized with regard to its contribution to the debate at hand. Recall that not all statements concerning the topic are to be considered a claim, but only those which refer to a specific action to be taken (cf. Section 2). Textual spans containing claims do not necessarily coincide with a sentence; they can be a subpart of a sentence, or span beyond the sentence boundary. After identifying relevant text spans, they are assigned to one or more claim categories from the codebook. Note that the initial codebook contains relevant categories found by domain experts on an initial sample of articles from dataset. However, throughout the annotation process, it has evolved, displaying the "hermeneutic cycle" which is typical of Digital Humanities projects (Blokker et al., 2021). The aptitude of individual claim-categories is constantly reviewed, new categories are adopted, outdated categories revised, overlapping categories merged.

Categories in the codebook are arranged hierarchically, with fine-grained subcategories being grouped together into supercategories which represent the separation of policy fields. Hierarchical schemes help researchers both with annotation as it is often easier when the annotation decision is first based on a supercategory and then on fine-grained subcategories. Table 1 lists the 8 high-level categories

ID	**Label**	**freq.**	**number of subcats.**	**percentage**
1xx	Controlling Migration	992	16	22
2xx	Residency	630	18	14
3xx	Integration	386	15	9
4xx	Domestic Security	154	9	3
5xx	Foreign Policy	711	9	16
6xx	Economy	153	12	3
7xx	Society	740	19	17
8xx	Procedures	651	20	15
	Overall	4417	118	

Table 1: High-level categories: Code; Label; Frequency; number of subcategories; the percentage over the total number of claims in the dataset

(supercategories) along with their 3 digit codes[4], frequencies and number of subcategories exist within each supercategory, available in the latest version of the dataset. The most frequent category is 'controlling migration', which contains demands and proposals concerned with regulating immigration (border controls, upper limit, asylum law, etc.). Related and also prominent is 'foreign policy' (e.g. EU-wide quota, international solutions). Other categories are 'society' that deals with humanitarian and cultural aspects (human rights, Christian values) and 'residency', mostly concerned with the accommodation of migrants. Least frequent are 'domestic security' and 'economy + labour market' which are further downstream of the acute (perceived) crisis situation. A special, less topical category is 'procedures' that often appears in combination with other categories (additional funding, transparency, etc.). Besides fine-grained cate-

[4] The first digit of the code denotes the supercateogry and the last two digits denote the subcategory.

gories, the codebook also contains descriptions and defining examples providing guidance to the annotators. Corresponding guidelines as well as the English version of the full codebook can be found in Appendix A.

Actors	
name	frequency
Angela Merkel	247
Thomas de Maizière	162
Bundesregierung	152
CSU	86
Horst Seehofer	79
SPD	78
EU	77
Sigmar Gabriel	68
Grüne	60
Jean-Claude Juncker	57

Table 2: The most frequent 10 actors

Along with assigning claim categories to the claims, annotators also perform actor identification and mapping. This involves identification of the strings corresponding to actor mentions (e.g., ‘Angela Merkel’, ‘Die Kanzlerin’, ‘Frau Merkel’) as well as mapping of the actor mention to a canonical name which serves as a unique identifier of the actor in the dataset (e.g., ‘Angela Merkel’ for ‘Die Kanzlerin’). Note that a single claim can be attributed to more than one actor, and actors can be mentioned inside or outside the textual span. Table 2 displays the 10 most frequent actors in the entire year of 2015 after actor mapping. Unsurprisingly, the most prominent actor of the migration crisis in Germany is chancellor Angela Merkel, followed by minister of the interior Thomas de Maizière, and, as institutional actor, the federal government ('Bundesregierung'). Other relevant actors include Merkel's political antagonist in the migration debate Horst Seehofer and chairman of the Christian Democratic Union (CSU), the Foreign Minister Sigmar Gabriel, and also the President of the European Commission Jean-Claude Juncker.

Once the claims are linked to the relevant actor(s) and claim categories, the next step is to perform polarity assignment to the claim, for which annotators need to figure out whether the actor attributed to the claim support or reject the categorized claim. Table 3 lists the most frequent positive and negative subcategories of 2015. EU-Solution (501), such as a Europe-wide quota for refugees, is the most often used claim with a positive polarity. Followed by calls for more funding (805) and an upper limit (102). Oppositely, frequent claim categories with negative polarity are calls to oppose xenophobia (703), right wing radicalism (709), and the current immigration policies (190). An EU solution is also high on the list, indicating that this is a polarizing and contested claim category.

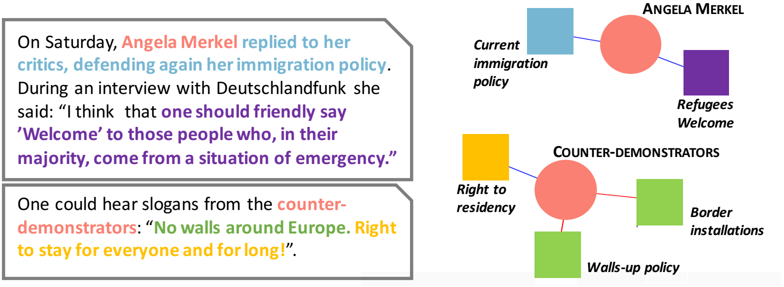
Figure 4 illustrates an example of the annotation and the discourse network that can be created from it. It is a real example from the dataset and it is based on the annotated documents for October 3rd, 2015. In one article, Angela Merkel is reported to have replied to those who criticised her immigration policy, and a direct quotation from her speech is reported, stating the need for a welcoming attitude towards refugees. The second set of claims are attributed to a group of counter-demonstrators, who showed up during an official ceremony in Saxony: claiming the right of residency for refugees, the

Positive Claims			
Code	Claim Category	Freq.	Total (incl. neg. Claims)
501	EU solution (quotas for refugees)	237	310
805	additional financing	147	155
102	ceiling/upper limit	129	152
812	fast / accelerated procedure	124	132
207	deportations	112	140
504	safe country of origin	112	153
105	border controls	103	126
705	refugees welcome	94	107
309	care (medical, financial, ...)	87	119
104	isolation/immigration stop	86	128

Negative Claims			
Code	Claim Category	Freq.	Total (incl. pos. Claims)
703	xenophobia	129	161
709	right-wing radicalism	86	98
190	current migration policy	73	95
501	EU solution (quotas for refugees)	73	310
202	refugee accommodation	56	79
110	asylum law	49	98
711	islam	47	63
104	isolation/immigration stop	42	128
504	safe country of origin	41	153
401	violence against migrants	36	44

Table 3: The most frequent positive and negative sub-categories

demonstrators also made two claims against the isolation of Europe and the construction of border installations as a solution to the immigration problem. The claims are highlighted in colors in the text, and give rise to the corresponding parts of the network representa-



On Saturday, Angela Merkel replied to her critics, defending again her immigration policy. During an interview with Deutschlandfunk she said: “I think that one should friendly say ’Welcome’ to those people who, in their majority, come from a situation of emergency.”

One could hear slogans from the counter-demonstrators: “No walls around Europe. Right to stay for everyone and for long!”.

Figure 4: Annotation and corresponding network representation for the immigration debate, October 3rd 2015

tion to the right. The actors are represented by red squares in the discourse network. Blue edges indicate support towards a claim category (Merkel supports the "Refugees Welcome" claim), red edges indicate opposition to it (the demonstrators stand against the claim "Controlling migration with border installations").

3.3 Conclusion

In this chapter, we described the DebateNet, an annotated dataset for the analysis of political debates, targeting the public discourse during the domestic debate on immigration in Germany in 2015. We showed that corpus annotation for the purposes of political claims analysis targets multiple levels with different degrees of abstraction and complexity for the annotator. We should note that while DebateNet targets a specific topic, the structures annotated in the dataset (e.g. actors, claims) are independent from the topic of discussion and applicable to any other political claims analysis datasets.

Part II

Towards Automatic Construction of Discourse Networks for Political Science

4 Decomposing Construction of Affiliation Networks into Subtasks

As described in Section-2, (manual-)construction of Affiliation Networks(ANs) from raw text requires identification of different syntactic and semantic properties as well as complex relations between them in raw text. Given the complexity of this task and the fact that today's NLP systems fall short of human-level general understanding, even though language models trained on huge datasets of raw text have pushed the limits of NLP (Rogers et al., 2020; Merrill et al., 2021), we see that it is unrealistic to attempt to automate the construction of ANs in an end-to-end fashion with a single model at a quality that makes it useful for political scientists. For this reason, in order to reduce the complexity of the task, we propose to decompose it into several sub-tasks which are conceptually simpler to learn and present a step-wise workflow for this purpose.

4.1 Workflow

Seen as an end-to-end task, the computational construction of affiliation networks from raw text can be decomposed conceptually as shown in Figure 5. Before going through the sub-tasks and describing their role in the pipeline, we should note that aside from automatic construction of ANs, such a decomposition can also be useful for the data annotation step. When the NLP tools designed for the individual sub-tasks are available, they can be integrated into the annotation environment as "pseudo-annotators" which can assist human annotators on corresponding tasks and can thus shorten the time and improve annotation reliability. We do not cover this "semi-automatic annotation approach" in this thesis; however, we refer interested readers to (Haunss et al., 2020) which compare political claim annotation quality and speed of annotators with and without machine learning based annotation support.

1. Claim Detection: The task of claim detection is related to Argumentation Mining (AM) (Peldszus and Stede, 2013). AM is a field encompassing varying tasks that deal with the automatic extraction of arguments in natural language texts. This involves separation of argumentative text units from non-argumentative units, parsing argument structures, and recognizing argumentative discourse units(ADUs). Claim detection can be defined as a task which is responsible for identification of a specific ADU type called claim which is defined as any kind of assertion that deserves our attention (Toulmin, 2003). Although it is mostly considered as a sub-component of AM as described above, we treat claim detection here as a stand-

alone task that aims to detect special kind of claims called "political claims" in running text which only covers a subset of what is often considered a claim in argument mining (cf. Section 3). Claim detection can computationally be framed as a sentence-level classification (Does the input sentence contain a claim?) or as token-level classification(Which tokens make up the claim?), both of which are popular tasks in NLP.

2. Claim Classification: As described in Section 3, the second step of annotation is to assign claim categories to claims. Under the assumption that the codebook is static, we can consider claim categories known a-priori, and hence model this step as a classification task. Specifically, we define claim classification as a text classification task on relatively short texts that deals with assigning claim categories from domain-specific codebooks to the claim spans detected by claim identification models. Depending on the properties of the annotation, the task is either single-label or multi-label text classification.

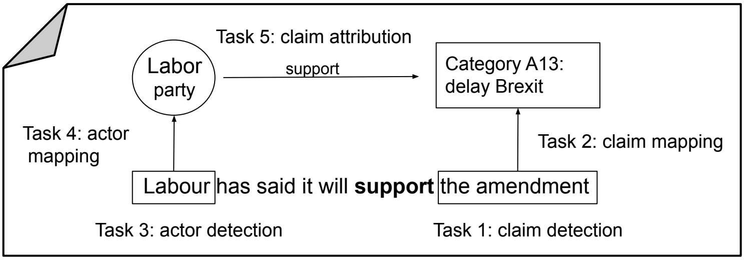


Figure 5: Workflow for Computational Construction of Affiliation Networks

3. Actor Detection: This task can be seen as a special Named Entity Recognition (NER) task[1]: The named entities such as a person, location, or organization denoting potential actors in the text need to be identified. The resulting candidate list can be narrowed down even further by Coreference Resolvers which can recognize textual spans referring to the same entity. Both NER and Coreference Resolution are fundamental tasks in NLP; nonetheless, there are still challenges in their use, especially with the data which is different from the type of data they are trained on: 1) Unlike regular NLP datasets, political texts often include definite descriptions referring to political actors that are challenging for both NER and Coreference Resolution models, such as "The minister of the interior". It is necessary for an NER model to recognize such phrases as named entities (as PERSON in this particular case), besides recognizing regular Person or Organization entities. For coreference resolvers the resolution of such references (e.g., The prime minister says) are even more difficult as it may require knowledge on various levels, from morphology up to semantics and pragmatics (Zhang et al., 2019). 2) Another major challenge with coreference resolvers is that state-of-the-art coreference resolution models use span representations and antecedent prediction mechanisms that are expensive in terms of both memory requirements and compute time, and are not particularly suitable for cases such as ours where input is generally a long newspaper article (Thirukovalluru et al., 2021).

[1] To some extent because there can be also actors that are not named entities

4. Actor Mapping: In order for actors to be represented in discourse networks, different textual strings referring to the same actor need to be mapped to the same canonical name which serves as a unique identifier of the actor in the dataset. The computational counterpart of this process is Entity Linking (EL), the task of establishing a link between an entity mention in the (unstructured) text and the corresponding entity in the knowledge bases (KB) such as Wikidata or Freebase. Although KBs contain rich and precise information about entities of all kinds, such as persons, locations, organizations, movies, and scientific theories; some entities, or entity relations which are only important for a specific topic might not be available in the KBs. This problem can be addressed by building a custom KB on the target corpus and using it in parallel with the standard KBs.

5. Claim Attribution: Claim Attribution is a relation identification task which deals with linking the previously identified claims (Task-1) with relevant actor(s) chosen among the candidates proposed by actor detection and entity linking models (Task-3,4). While pairing claims and corresponding actors, the identification of relation type (i.e. support vs opposition) also need to be determined. The arrow on the top of Figure 5 labeled with "support" pictures output of Claim Attribution model: It makes a supportive connection between "Labor party" with a claim on delaying Brexit.

Discussion. We acknowledge that the workflow proposed above is not the only way, and perhaps for some, not even the best way of decomposing the computational construction of affiliation networks into smaller steps. Differently from our proposal, one may think that the various tasks are clearly not independent of one another, and joint models could have been developed for a subset of the tasks. For instance, claim detection (Task 1) and claim classification (Task 2) would be a good example for tasks that can be modelled jointly. A joint model for these two tasks can recover from the errors that propagate between stages of a pipeline solution. Nevertheless, such a joint model also has some shortcomings: We believe this design choice would reduce the applicability of the claim detection model to different policy debate topics as claim classification is a substantially more domain-specific task than claim detection. Claim detection and actor detection (Task 3) are another possible combination for joint modelling. While on the one hand such a joint model has the potential to perform better as it in theory allows information sharing between tasks (Chen et al., 2018), on the other hand, using a single representation for the two tasks would sometimes also lead to feature confusion i.e., features extracted for one task may conflict with those for the other, thus confusing the learning model (Wang and Lu, 2020). The latter problem circumvented by, for example, using distinct encoders to capture such two different types of information in the learning process, but this solution would require working with even larger annotated datasets, which is not preferable for CSS.

In the rest of this part (Part II), we discuss the two components of the proposed workflow, namely claim detection (Chapter 5) and claim classification (Chapter 6), which are responsible for construction of the claim nodes that are one of the two node types in the Affiliation Networks (cf. Chapter 2). This can be motivated by practical considerations: 1) As stated above, beyond the computational construction of discourse networks, these methods can also be integrated into the annotation environment to speed up the manual annotation process, in which case automating these two tasks (especially claim detection) would arguably lead to highest speed up in the annotation as they prevent the annotators from reading the entire newspaper articles. 2) Modeling and annotation works have been carried out in parallel and some of the annotations were not available at the beginning, which also influenced which task to focus on initially.

5 Claim Detection

In the preceding chapters, we explained that claims are conceptual building block of the DNA framework and form one of the two node types in a bipartite affiliation network. Consequently, developing reasonable models for claim detection is one of the main prerequisites for automatic construction of affiliation networks from raw text. In this chapter, we discuss exactly this issue, namely, automatic claim detection task. We describe neural network based NLP methods for claim detection task and evaluate on a policy debate on the controversial topic of immigration.

5.1 Introduction

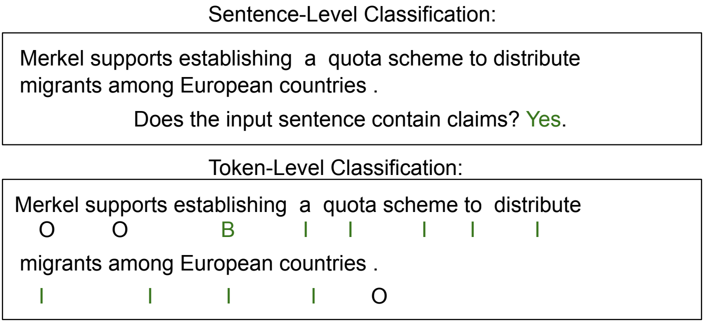
Claim detection has become a popular research area both in NLP and Political Science. Among NLP scholars, claim detection is considered as a fundamental task in Argumentation Mining(AM) (Daxenberger et al., 2017). Although there is no argument model that most researchers agree upon and the chosen argument model often depends on the tasks and the application domain, most of the recent research agrees that the building block of the argument is claim (Liebeck et al., 2016). Outside the realm of AM, the ability to analyze claims is crucial for tasks such as fine-grained opinion analysis (Yang and Cardie, 2013) and stance classification (Anand et al., 2011; Lai et al., 2019). The common point of these and many other studies dealing with claim detection in NLP is to define claim as "an assertion put forward publicly for general acceptance" which is first proposed in Toulmin (2003).

Political Science scholars, on the other hand, mostly rely on another definition of claim proposed by Koopmans and Statham (1999) for their studies on manual claim detection. According to Koopmans and Statham, the (political) claims should be understood as utterances, actions or other statements made in public and can be defined as "the strategic demands made by collective actors within a specific contested issue field". Koopmans and Statham (1999)'s claim definition covers only a subset of what is often considered a claim according to Toulmin (2003)'s definition.

In this chapter, we propose neural network based models for the task of claim detection. Our contribution is to study the automatic claim detection task using Koopmans and Statham's claim definition. It is more challenging for computational models to detect claims based on Koopmans and Statham's definition than Toulmin's definition because the former definition requires, not only detecting statements that are in dispute but also distinguishing the ones that are relevant to the discussion. In the rest of this chapter, we review the political claim detection task; compare the two different ways the claim detection can be formulated as; describe the automatic claim detection models; and discuss our results on DebateNet.

5.2 Task

As described in Chapter 4, claim detection aims at identifying claims in a given text. Following Koopmans and Statham (1999), we define claim as statements concerning specific actions to be taken with respect to a specific aspect of a domain of interest. Computationally, claim detection can be framed in various ways:



Sentence-Level Classification:

Merkel supports establishing a quota scheme to distribute migrants among European countries .

Does the input sentence contain claims? Yes.

Token-Level Classification:

Merkel supports establishing a quota scheme to distribute migrants among European countries .

Figure 6: Different formulations for claim detection.

Settings for Automatic Claim Detection. The task of automatic claim detection can be framed either as a token-level or as a sentence-level classification task. In the former case, the goal is to answer to the question of "Which tokens make up the claim?" by generating an output label for every token in an input sequence: For a given text $x_{1:N} = [w_1, w_2, \dots w_N]$, a detection model is trained to predict a sequence of labels $y_{1:N} = [y_1, y_2, \dots y_N]$, where y_i indicates whether w_i is in a claim or not. Sentence-level classification setup on the other hand addresses the question of whether or not the input sentence $x_{1:N} = [w_1, w_2, \dots w_N]$ overlaps, partially or completely, with a claim span which requires a single binary decision per input. Figure 6 shows the input/output pair for both formulations.

Both settings have their advantages and disadvantages. Sentence-level classification setting is more coarse-grained and, therefore, more likely to perform well than the token-level classification. A potential drawback of this setting is that it may lead to missing of the exact starting and endpoints of claim spans if the claim boundaries are not aligned with sentence boundaries in the dataset at hand, which can be problematic depending on the use case. Token-level classification, on the other hand, has the potential to output more exactly which tokens belong to the claim. However, this setup has its own disadvantages: Token-level classification is a structured prediction task where systems need to assign the correct label to every token in the input sequence. Due to the exponential size of the output space, structured learning problems tend to be more challenging than the conventional sentence-level classification problems (Nguyen and Guo, 2007; Papay et al., 2020a). Besides increasing modeling complexity, this setup also incurs annotation complexity as optimization of these models generally requires appropriate training data where annotation is performed on the token level. Hence, as we mention above, both approaches offer advantages and disadvantages, depending on the use case. In the next section, we investigate NLP models from both setups.

5.3 Models

For both configurations, we adopt the approach that has become standard for semantic tasks in NLP, namely neural network models.

5.3.1 Claim Detection as Sentence-Level Classification

Our sentence-level claim detection model is a fairly standard model consisting of a pre-trained BERT as an embedding layer, and a linear classifier with sigmoid activation function as an output layer. As described in Chapter 2, BERT is a stack of bidirectional Transformer encoder layers (Vaswani et al., 2017) that consist of multiple self-attention heads and a feed-forward network. Compared to the traditional FastText- or GloVe based embedding layer which only provides a single context-independent representation for each token, the BERT embedding layer takes the sentence as input and calculates the token-level representations using the information from the entire sentence.

Figure 7 illustrates the architecture of our network in detail. First, we use WordPiece (Wu et al., 2016) to split the input text into word pieces and add a special token, [CLS], to the beginning of the tokenized sequence. After tokenization, we construct input representation e_i for each token w_i in the input sequence S by summing the corresponding token, segment, and position embeddings:

$$e_i = \mathbf{p}_i + \mathbf{t}_i + \mathbf{s}_i \tag{5.1}$$

where $\mathbf{p}_i$, $\mathbf{t}_i$, and $\mathbf{s}_i$ are the position, token, and segment embeddings for w_i respectively. Such input representations are then fed into L successive Transformer encoder layers to generate deep, context-aware representations for the tokens in the input sequence:

$$\mathbf{h}_i^\ell = \text{Transformer}\left(\mathbf{h}_i^{\ell-1}\right), \quad \ell = 1, 2, \cdots, L \tag{5.2}$$

where $\mathbf{h}_i^0 = e_i$. Following Devlin et al. (2019), we regard the final hidden state corresponding to special token [CLS], H_0^L, as the aggregate sequence representation and feed it into the output layer with sigmoid activation function for binary classification.

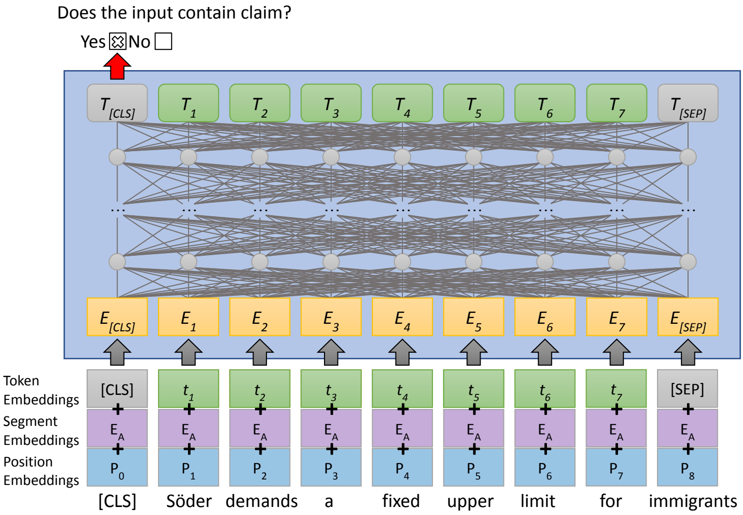

Does the input contain claim?
Yes ☒ No □
$T_{[CLS]}$ T_1 T_2 T_3 T_4 T_5 T_6 T_7 $T_{[SEP]}$
$E_{[CLS]}$ E_1 E_2 E_3 E_4 E_5 E_6 E_7 $E_{[SEP]}$
Token Embeddings
Segment Embeddings
Position Embeddings
[CLS] t_1 t_2 t_3 t_4 t_5 t_6 t_7 [SEP]
E_A
P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8
[CLS] Söder demands a fixed upper limit for immigrants

Figure 7: Visualization of the model tackling claim detection as sentence-level classification task.

5.3.2 Claim Detection as Token-Level Classification

For sequence labeling models, we adapt the IOB format (Ramshaw and Marcus, 1999), a common tagging format for tagging tokens. Specifically, we label each token in a sentence as B-Claim, I-Claim or Outside. Using Bidirectional LSTM(BiLSTM), Convolutional Neural Network(CNN), BERT and Conditional Random Field(CRF), we developed set of models where the input to the model consists of an N word sentence S $= [w_1, \cdots, w_N]$, w_i denoting the i^{th} word in the sentence:

BiLSTM. We use FastText embeddings (Bojanowski et al., 2017b) to represent words in the input. On separate experiments, we try using both publicly available fastText embeddings and in-domain FastText embeddings that we trained on our own corpus. These word embeddings $e_1, \cdots, e_N$ are fed into the BiLSTM which processes them in both directions and constructs a unique representation for each word in the input sequence. For a word w_i we define its corresponding representation c_i as the concatenation of the forward $\overrightarrow{c}_i$ and the backward $\overleftarrow{c}_i$ hidden states that are produced after the forward and backward LSTMs process the embedding e_i. This token representation c_i is then fed into the output layer with softmax function whose dimension is the number of tag types. It outputs the tag probability distribution of input word w$_i$.

BiLSTM+CNN. In this model, each word is represented as a combination of the word-based and character-based representations. We use in-domain FastText embeddings to obtain a word-level representation. Character-based representations for each word

$w_i = [w_{i1}, \cdots, w_{ij}]$ where w_{ij} is the jth character in the ith word, are obtained by applying a CNN with a max-pooling layer to the character sequence in the word. A dropout layer (Srivastava et al., 2014) is applied before character embeddings are input to CNN. The rest of this model is exactly identical to the BiLSTM model described above.

BiLSTM+CNN+CRF. Figure 8 illustrates BiLSTM+CNN+CRF architecture in detail. The difference between this model and BiLSTM+CNN is that we replace the softmax layer at the top of BiLSTM+CNN model with CRF layer in order to jointly model the label sequence. For a sequence with n words, we parameterize the distribution over all possible label sequences, $\mathcal{Y}$, as

$$p(\boldsymbol{y}|\mathbf{c};\mathbf{W}) = \frac{\prod\limits_{i=1}^{n} \phi_i \left(y_{i-1}, y_i, \mathbf{c}\right)}{\sum\limits_{y' \in \mathcal{Y}} \prod\limits_{i=1}^{n} \phi_i \left(y'_{i-1}, y'_i, \mathbf{c}\right)} \tag{5.3}$$

where $c = [c_1, c_2, \ldots c_n]$ is the set of representation produced by BiLSTM for each input word and $\phi_i \left(y_{i-1}, y_i, \mathbf{c}\right)$ is a function calculating emission and transition potentials between the tags y_{i-1} and y_i. During training, we maximize the log-likelihood function over the training set, and during inference, the sequence with highest conditional probability is predicted by a Viterbi decoder:

$$L(\mathbf{W}) = \sum_i \log p(\boldsymbol{y}|\mathbf{c};\mathbf{W}) \tag{5.4}$$

$$\underset{y \in \mathcal{Y}}{\operatorname{argmax}}\ p(\boldsymbol{y}|\mathbf{c};\mathbf{W}) \tag{5.5}$$

BERT Unlike previous token-level claim classification models, this architecture doesn't use FastText embeddings. Instead, we use pre-trained BERT model to obtain context-aware representations for the tokens in the input sequence. As described in Chapter 2, BERT separates tokens into subtokens using WordPiece tokenization(Wu et al., 2016), which means it may generate multiple representations for a single word. When a word consists of multiple tokens, we use the hidden state corresponding to the first sub-token as input to the classifier. Since the WordPiece tokenization boundaries are a known part of the input, this is done for both training and test time. Following Devlin et al. (2019), we do not use a CRF layer in the output. We use the representation of the first sub-token as the input to the token-level classifier which outputs the tag probability distribution of the corresponding input word.

5.4 Experimental Setup

5.4.1 Dataset

We conduct our claim detection experiments on DebateNet dataset. Specifically, we use the first release of DebateNet (Padó et al., 2019)

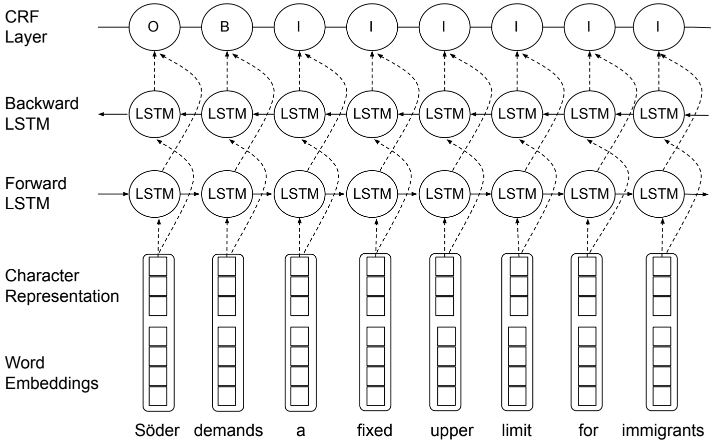


Figure 8: Visualization of the model that tackles claim detection as token-level classification problem

which comprises 423 annotated articles from the 2015 Tageszeitung, among which 179 articles contain at least one claim. As described in Chapter 3, the corpus contains 982 Claims in 764 different text passages. We discarded articles with no claims, and randomly sampled 90% of our dataset for training and evaluate on the other 10%.

5.4.2 Training Details

Token-Level Classification Models In the CNN which we use to obtained character-level embeddings, we set the kernel size to 3 and filter number to 30. The input characters of the CNN are represented as 25 dimensional vectors. Maximum word length is set to

20 characters. This configuration yields 750 dimensional ($= 30 * 25$) character-level embeddings. We set word embedding size to 100 when we use in-domain embeddings; and 300 when we use publicly available FastText embeddings. In the LSTM layer, we set the number of hidden units in each direction to 100. We use standard SGD as optimizer and set learning rate and decay rate to 0.01 and 0.05 respectively. We divide data-sets into batches of 10 sentences and train the model for 50 epochs. For the BERT model, we use the publicly available multilingual case-sensitive model that is pre-trained on the top 104 languages with the largest Wikipedia. We use the base model with default parameters: The number of attention heads, hidden layers, and the number of hidden units are 12, 12, and 768, respectively. During fine-tuning, we set the maximum sequence length to 75, batch size to 32 and norm of maximum gradient to 1.0. We use Adam optimizer with 0.003 learning rate. We did not freeze any part of the model and fine-tuned it for 20 epochs with this configuration.

Sentence-Level Classification Models For the sentence level claim identification model, we use publicly available BERT model[1] trained solely on German corpora; including German Wikipedia dump, the OpenLegalData dump, and news articles. We use the base model with default parameters which are the same as multilingual model's hyperparameters. During fine-tuning, we set the maximum sequence length to 128, batch size to 32 and norm of maximum gradient to 1.5. We use Adam optimizer with 3e-5 learning rate. We did not freeze any part of the model and fine-tuned it for 20 epochs.

[1] https://www.deepset.ai/german-bert

5.5 Results

Token-level Classification Results Table 4 shows the performance of token-level claim detection models described in Section 5.3.2. Our simplest model, BiLSTM only model with publicly available FastText word-level embeddings (1), achieves F_1 scores of 31.3 and 37.5 for classes B-C and I-C, as well as a macro average F1 score of 54.1. When replacing the Wikipedia FastText embeddings with our in-domain FastText embeddings trained on our complete TAZ newspaper corpus, we observe that the resulting model, (2), achieves 58.7 macro F_1, 4 points improvement over the first model. The next row in the table shows results of model (3) which combines in-domain word embeddings with character-based representations obtained by a CNN component for input representation. This results in 1.5 points increase in F_1 for the B-C class and 0.3 point improvement in the macro F_1 score. Replacing the softmax layer in the BiLSTM+CNN with a CRF layer results in model (4) which yields F_1 scores of 49.4, 53.8, 95.5 for B-C, I-C and O classes respectively, with a major 7 points increase in averaged F_1 over model (3). 11 points drop in F_1 score between model (4) and (5) shows that just like the BiLSTM model (cf. (1) vs. (2)), CNN+BiLSTM+CRF model also profits substantially from the in-domain embeddings. The last row in Table 4 shows the result of our BERT based claim detection model (6) which achieves a macro F_1 score of 65.5, comparable to model (4).

Performance differences among the models in Table 4 provide useful insights regarding the different components tested above. For instance, we see that although the use of CNN to include character-level information is useful, it improves the performance to some extent only. We suspect this is because FastText includes character information and probably already helps unseen and under-represented words. Unlike CNN, we observe that the use of CRF has a huge positive impact on the performance. Another component that we recommend to use is in-domain embeddings which are very effective regardless of the complexity of the model used: It leads to substantial improvements for both simple methods such as BiLSTM ((1) vs (2)) and composite models such as BiLSTM+CNN+CRF ((4) vs (5)). Use of in-domain embeddings makes this model to outperform, though slightly, the transformer based BERT model as well which we, similar to Lai et al. (2021), think is due to the limited amount of available training data.

ID	Configuration				Performance			
	Use of FastText	Use of CNN	Use of CRF	Use of BERT	B-C F_1	I-C F_1	O-C F_1	Macro F_1
(1)	global	-	-	-	31.3	37.5	93.5	54.1
(2)	in-domain	-	-	-	38.5	43.9	93.6	58.7
(3)	in-domain	+	-	-	40.0	44.1	93.1	59.1
(4)	in-domain	+	+	-	49.4	**53.8**	**95.5**	**66.3**
(5)	global	+	+	-	35.1	39.1	90.6	55.0
(6)	-	-	-	+	**49.5**	52.4	94.7	65.5

Table 4: Claim identification scores of token-level models on the evaluation set.

Sentence-Level Classification Results Table 5 shows Precision, Recall and F_1 Score of our BERT based model which tackles claim detection as a sentence-level binary classification task. Our model achieves F_1 score of 52.2 with 40.1 points Precision and 74.7 points Recall. Besides sentence-level classification performance, we also evaluate token-level classification performance of our model for comparison to models reported in Table 4. For this, we converted sentence level labels to token level as follows: All tokens in a sentence predicted as "claim-free" by the model are labeled "O". For the other sentences, the first token is labeled "B-C" and the rest labeled "I-C". We should remind that BERT uses WordPiece subword tokenization algorithm rather than whole-word tokenization which may lead some words to be broken up into smaller subtokens. In such cases, we followed the original study, Devlin et al. (2019), and use the prediction on the first subtoken of the word. In this way, we keep token-level classification results of BERT model comperable with the other models. After converting sentence level labels to token level, we measure the token-level Macro F1-Score using the same evaluation on the same splits and report it at the right hand side of Table 5. We make the surprising observation that sentence level claim detection model achieves macro F_1 score of 67.6 when evaluated at token level, a moderate improvement (+1.3 Points) over the best model in Table 4. We believe this finding is the joint result of i) the fact that, as explained in Section 5.2, modeling the task on the sentence-level reduces complexity which generally makes learning easier and ii) characteristics of claims in our dataset: We find that, although claims can theoretically be a sub-part of a sentence, or span beyond the sentence boundary; their boundaries overlap with sentence boundaries very frequently in our case. Analysis on our dataset shows that 90% of the claim spans begins before the third word of the sentence, and 88.3% of the tokens in sentences that overlap with claim spans are marked as part of claim (i.e. either "B-Claim" or "I-Claim").

	Sentence Level Evaluation			Token Level Evaluation
	Precision	Recall	F1-Score	F1-Score
BERT	31.4	74.4	44.2	67.6

Table 5: Claim identification scores of sentence-level model. Precision, Recall, F_1 Scores are for sentence-level labels. The right most column indicates macro average of F_1 scores of token level BIO labels.

5.6 Conclusion

In this chapter, we discussed the task of automatic claim detection. First, we have shown that the task can be framed either as token-level or sentence-level classification task, and each configuration has its own advantages and disadvantages. Namely, the text classification setting is more coarse-grained with one less label than token-level classification setting. The disadvantage this approach is obvious; it might miss the exact starting and endpoints of a claim, if the start and end points of claims and sentences are not aligned in the dataset. Unlike sentence-level classification setting, the token-level classification setting can potentially output exactly which tokens belong to the claim. The problems with this setting, on the other hand, are the fact that it is a more challenging setup and it requires more fine-grained annotation dataset. Next, we described various claim detection models from both settings and evaluated them quantitatively on the manually annotated German public debate on immigration in 2015. We found that the best token-level classification model (cf. Model (4) in Table 4) achieves 66.3 F_1 score. On the same dataset, our BERT based sentence-level claim detection model achieves 44.2 F_1 score on sentence level which corresponds to 67.6 F_1 score on token level.

Our quantitative evaluation shows that we are able to detect automatically the claims in a debate that are relevant to the topic at hand with a reasonable performance. However, it does not provide insights into what kind of features does the model use while making predictions. With this in mind, we also performed a qualitative analysis and found that our sentence-level claim detection model predicts only the first of the following two statements as a claim, but both statements are claims and are identical except for the actor's name:

(1) S1:Angela Merkel meldet Zweifel an der Umverteilung von Flüchtlingen an
S2:Pro Asyl meldet Zweifel an der Umverteilung von Flüchtlingen an

"Angela Merkel/Pro Asyl expresses doubts about the redistribution of refugees."

where S1 is a real example from DebateNet and we created S2 by replacing the actor name with a less frequent one. We interpret these patterns as overfitting of the claim detection model: It relies too much on actor mentions (i.e., either proper names or pronouns) as indicators of claims. As described in Chapter 1, a central concern in CSS is fairness and the quality of computational tools whose role is to scale up text analysis to large corpora should be as independent as possible of textual properties that are irrelevant to the target. We will continue to discuss this topic in Chapter 7.

6 Claim Classification

In Chapter 4, we provided a step-wise description of a workflow that allows us to break down the task of automatic generation of ANs into several subtasks that are conceptually easier to learn, and in the last chapter we discussed automatic claim detection, the first step of the workflow. Through our experiments on DebateNet we showed that we are able to automatically detect the claims with a reasonable performance. In this chapter, we move to the next step of workflow and discuss the task of claim classification.

6.1 Introduction

The categorization of unstructured text into categories has become an increasingly important step in the Political Science, and in CSS in general as it enables researchers to perform more detailed analysis on the large amounts of unstructured data (Grimmer and Stewart, 2013).

In this chapter, we present various neural network based claim classification methods ranging from a simple Bidirectional LSTM model to more complex context-aware method, based on the BERT for automatic claim classification. Our goal is to replicate the manual coding task, but with a computational model. First human coders are used to classify a subset of claims into a predetermined categorization scheme. For instance, the following claim in DebateNet (cf. Chapter 3):

(1) Eine weitere massive Verfahrensbeschleunigung ist bei vorübergehenden Grenzkontrollen vor der Einreise vorgesehen

"A further massive acceleration of procedures is envisaged for temporary border controls prior to entry"

is assigned to Border Controls and Accelerated Procedure categories from the codebook. Then, this training set is used to train the automated methods, which then classify remaining claims. In the rest of this chapter, we first define the automatic claim classification task; then describe the neural models we develop; and finally discuss the results obtained on DebateNet dataset.

6.2 Task

Even though claim classification is fundamentally a text classification task, it has its own characteristics. As described in Chapter 2, text classification is an important task, and it can be used in a broad range of contexts including classifying very large documents such as customer reviews(Liu et al., 2014), news articles (Büyüköz et al., 2020) or legal contracts (Tuggener et al., 2020). Claim classification, on the other hand, specifically deals with classification of claims, a special kind of political statements that are shorter than paragraphs or documents, into categories based on a domain-specific codebook. Regarding the type of the input to be classified, this task is also highly related to frame classification (Boydstun et al., 2014; Naderi and Hirst, 2017; Heinisch and Cimiano, 2021) but they are not the same. The latter one aims at classifying the frames which are short statements that are used as justification of a claim, while the former one aims at classification of the claim itself.

As mentioned in Section 6.1, it is a common practice in computational political science or in text-based CSS in general to classify such short statements with the help of a codebook. Hemphill et al. (2021) train a classifier to label tweets to examine the differential attention that policy topics receive from Members of the US Congress using the Comparative Agenda Project's Policy (CAP) Codebook (Baumgartner et al., 2006). The same CAP codebook has also been used to label several other short text CSS corpora with more formal language such as newspaper headlines and US Congressional bills, on which Terechshenko et al. (2020) performed experiments with standard deep learning based classification methods. Party manifestos are another widely used information source in CSS. These manifestos include short statements declaring parties' positions over a range of topics (e.g., Foreign policies, Welfare, Economy) and political scientists have been topically coding manifestos from countries around the world within the Comparative Manifesto Project (CMP). The resulting corpora has been used by previous work to train classifiers that can categorize short statements in party manifestos based on the labels available in the codebook.(Glavaš et al., 2017; Subramanian et al., 2018).

In this chapter, we assume that claims have already been detected in the first step of the workflow (cf. Chapter 4) and define claim classification as follows: Given a claim and a set of possible classes (e.g., Safety, Economy, Security) to which the claim can belong, the goal is to predict the correct claim category label(s) assigned to the input. Depending on the characteristics of the policy debate dataset, the claim can be a segment, a sentence or a set of sentences. Furthermore, again, depending on the dataset characteristics, claims can be related to multiple policy issues (e.g., DebateNet), or single policy issue, which determines whether the claim classification task is configured as a multi-label classification (MLC) task, or as a single-label classification, a simplified version of MLC. In the rest of this chapter, we treat claim classification as a multi-label text classification task over the claim statements that can theoretically be sub-part of a sentence or span beyond the sentence boundary.

6.3 Models

Classification of short texts has been shown to be more challenging than document level classification as they contain less words and it might not be always obvious to find the corresponding category(Lee and Dernoncourt, 2016). To overcome this difficulty, some previous work, such as Yan and Guo (2019); Liu et al. (2022), take advantage of the context information to help classify the current sequence. The main idea of leveraging contextual information is to extend the input into a sequence covering the preceding and successive sentences. Although this might have helped to improve performance, to some extent, on some specific cases, we do not consider the subsequent sentences in our model because (1) claim spans often provide the necessary topical information alone, we do not consider the subsequent sentences and (2) this would have restricted general applicability of our models as context information is not always available (e.g. manifestos). Instead, we assume that the input to the model consists of an N word claim $S = [w_1, \ldots, w_N]$, where w_i is the i'th word and the model assigns a set of labels to the claim. We conduct experiments with four neural network models ranging from unidirectional LSTMs to state-of-the-art transformer based architectures:

LSTM. Our simplest model is a unidirectional LSTM. After embedding the input word sequence using FastText embeddings, this model passes the input through a single-layer LSTM. The final hidden state is used as input to a fully connected layer.

BiLSTM. This model is identical to the first one, except that the LSTM is replaced with Bidirectional LSTM (Graves et al., 2013) which consists of two LSTM components traversing the input sequence in opposite directions. The final hidden states in both directions are concatenated and fed to a fully connected layer.

BiLSTM+Attention Our third model combines BiLSTM with an attention mechanism which aims to encourage the model to focus on salient local information that is relevant for the classification decision. First, a bi-directional LSTM is applied for both the right and left context:

$$\overrightarrow{h}_i = \mathrm{LSTM}_f(e_i, \overrightarrow{h}_{i-1}) \tag{6.1}$$

$$\overleftarrow{h}_i = \mathrm{LSTM}_b(e_i, \overleftarrow{h}_{i+1}) \tag{6.2}$$

$$h_i = \left[\overrightarrow{h}_i; \overleftarrow{h}_i\right] \tag{6.3}$$

where e_i is the word embedding for word w_i and h_i is the concetenation of $\overrightarrow{h}_{i-1}$ and $\overleftarrow{h}_{i+1}$ which are the hidden states that are produced after the forward and backward LSTMs process the word embedding e_i. Next, for each h_i, a scalar value a_i is computed using a feed forward neural network with the hidden layer:

$$a_i = \exp\left(W_a * \tanh(W_e * h_i)\right) \tag{6.4}$$

After normalizing the scalar a_i values such that they sum to 1, we compute the sum of the output layers of the bidirectional LSTM, weighted by the attentions ai as the representation of the input text, v_c:

$$\tilde{a}_i = a_i / \sum_{j=1}^{N} a_j \tag{6.5}$$

$$v_c = \sum_{i=1}^{N} \tilde{a}_i h_i \tag{6.6}$$

Finally, v_c, weighted sum of the hidden states, is fed to a fully connected layer.

BERT Our last model is a BERT-based model pretrained solely on German corpora [1]. Similar to sentence-level claim detection model described in Chapter 5, input text is tokenized using WordPiece; the [CLS] special token is added to the beginning of the token sequence. The resulting input representations are then fed into successive Transformer encoder layers to generate latent context-aware representations. We treat the final hidden state of BERT model corresponding to [CLS] token as the

[1] https://deepset.ai/german-bert

contextualized representation of the input sequence and feed it to a fully connected layer.

In all of our models, we use sigmoid activation function in the output layer as the output classes are not mutually exclusive and many of them can be choosen at the same time. We provide hyperparameters for all models in Section 6.4.2

6.4 Experimental Setup

6.4.1 Dataset

We conduct our claim classification experiments on the DebateNet dataset using the version released in Blokker et al. (2021). As described in Chapter 3, the corpus is annotated manually according to a two-level ontology for the migration domain, comprising 8 supercategories with 118 subcategories(cf. Table 1 in Chapter 3). There is a total of 3827 annotated textual spans. Similar to Chapter 5, we randomly sample 90% of our dataset for training and evaluate on the other 10%.

6.4.2 Training Details

As our preliminary experiments show that such extremely infrequent categories hinder convergence during training, we, similar to Shardlow et al. (2022), reduce the number of subcategories by combining subcategories with less than 20 instances under the preexisting subcategory x99, which exists for each supercategory as a ‘catch-all’ category for outlier cases. We acknowledge that that makes the catch-all subcategories are presumably challenging to learn but we believe that this strategy is reasonable, since no instances are discarded in this manner.

After filtering, there are 8 super- and 72 subcategories left in the dataset. Table 6 shows the number of categories in each subcategory after filtering operation. As described in Section 6.2, we model claim classification as multi-label text classification task. In particular, our models are designed for flat text classification where the goal is to predict correct claim labels among 72 classes for a given input claim.

Code	Supercategory Label	Freq.	Number of subcats.	Reduced number of subcats.
1xx	Controlling Migration	998	16	12
2xx	Residency	726	18	12
3xx	Integration	475	15	6
4xx	Domestic Security	230	9	6
5xx	Foreign Policy	689	9	8
6xx	Economy	194	12	5
7xx	Society	749	19	12
8xx	Procedures	676	20	11

Table 6: Claim distribution for each supercategory before and after removal of extremely infrequent subcategories. Freq.: Frequency, number of textual spans in each super-category. Total num. of sub-cats:118; Total reduced num. of subcats:72.

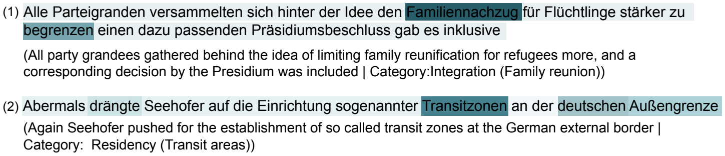
In all models except BERT, we set the number of hidden units to 500 in each direction and use 300-dimensional FastText word embeddings trained on immigration-related German articles published on TAZ. We use Adam(Kingma and Ba, 2015) with learning rate of 0.003 as optimizer; set batch size to 16 and train models for 20 epochs. For the BERT model, we use a cased BERT variant[2] that was trained specifically for German with default parameters for the number of attention heads, hidden layers, and the number of hidden units are 12, 12, and 768, respectively. During fine-tuning, we use the Adam optimizer with learning rates of 5e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and set the maximum sequence length to 200, batch size to 16 and norm of maximum gradient to 1.0 and trained for 20 epochs. All models are trained using cross entropy loss with the sigmoid activation function.

6.5 Results

Table 7 shows Precision, Recall and F_1 scores from all four claim classifiers. We observe that a unidirectional LSTM model achieves an F_1 score of 0.30 with 0.50 Precision and 0.24 Recall. Replacing a unidirectional LSTM with a bidirectional one leads to a slight improvement in performance, a 2 points increase in F_1 score. We find that there is a major effect of using an attention mechanism on performance. The resulting BiLSTM+ATTN model yields F_1 score of 0.46 with a 16 points increase in Precision and 13 points increase in Recall over the BiLSTM model. We believe this significant improvement is the result of the fact that attention mechanism helps to build more effective semantic representations by focusing on the words that are important for determining the categories the claim is related to. To support our hypothesis, we present visualization of attention weights for BiLSTM+ATTN model on two examples which are only correctly classified by BiLSTM+ATTN between the

[2] https://deepset.ai/german-bert

two models. Figure 9 shows that BiLSTM+ATTN model learns to attend words such as *Familiennachzug(Family reunification)* or *Transitzonen(i.e. transit zones)* that are relevant for the classification.



(1) Alle Parteigranden versammelten sich hinter der Idee den Familiennachzug für Flüchtlinge stärker zu begrenzen einen dazu passenden Präsidiumsbeschluss gab es inklusive

(All party grandees gathered behind the idea of limiting family reunification for refugees more, and a corresponding decision by the Presidium was included | Category:Integration (Family reunion))

(2) Abermals drängte Seehofer auf die Einrichtung sogenannter Transitzonen an der deutschen Außengrenze

(Again Seehofer pushed for the establishment of so called transit zones at the German external border | Category: Residency (Transit areas))

Figure 9: Attention weight visualization for BiLSTM+ATTN model on two claims from DebateNet. Words are highlighted according to attention scores.

In last row of the Table 7, we present results from BERT, another model with an attention mechanism. Our BERT based claim classifier achieves the best Recall (0.42) and F_1 (0.47) scores with an improvement of 3 and 1 points respectively over the BiLSTM+ATTN model. We attribute this improvement to two factors: Multiple self-attention modules located at different layers where each module learns features in different representation subspaces, leading to multi-representation that further improve performance, and knowledge gained during pre-training of BERT model where the orders of magnitude larger data was available. Overall, we find that these models, especially the BiLSTM+ATTN and BERT, do perform surprisingly well considering the large number of categories and limited amount of instances available in the data.

In addition to evaluating the overall performance of the models, we also analyze the performance of four models on individual claim

	Precision	Recall	F_1-Score
LSTM	0.50	0.24	0.30
BiLSTM	0.51	0.26	0.32
BiLSTM+ATTN	**0.67**	0.39	0.46
BERT	0.61	**0.42**	**0.47**

Table 7: Precision, Recall, F1 Scores for the four claim classification models

	LSTM	BiLSTM	BiLSTM+ATTN	BERT
p	0.46	0.42	0.54	0.60

Table 8: Spearman's rank correlation between F1 Scores of models and per-category training data size

categories. Figure 10 shows the per-claim category F1-Score of each model as well as the normalized number of claims available in training set for each category. First, we see that the LSTM and BiLSTM models performs poorly with F_1 scores less than 0.5 across categories in general. These models perform worse than the BiLSTM+ATTN and BERT models on 66 categories out of 72. Moreover, we observe that the LSTM model completely fails on more abstract categories such as Society (7xx), where it reports 0 F_1 scores for 9 labels out of 12 society related categories.

Second, we compare our two best performing models (BERT and BiLSTM+ATTN) and find that despite similar overall F_1 scores achieved, these models behave differently on different categories. As an example, BERT outperforms BiLSTM+ATTN with a high margin on eight "Controlling Migration" (1xx) related categories out of eleven. On the other hand, BiLSTM+ATTN model has a better performance on the "Security"(4xx) related categories. Further, we distinguish three regions in the Figure 10: We observe that on the left-hand side of the Figure in which the least amount of training data per category available, both models perform very bad with almost always 0 F1-Score. When we look at the middle region where categories have relatively more instances compared to the previous region, we realize BiLSTM+ATTN model outperforms BERT. Finally, when we move to the right side of the figure, we see the opposite trend: BERT yields better F1 Scores than the BiLSTM+ATTN model in general. We think these different patterns arise mainly because of the fact that the amount of available data varies among different categories in DebateNet and relatively small size of dataset restricted better performance in BERT because, as reported by previous work(Masala et al., 2021; Lai et al., 2021), transformer-based models such as BERT are more complicated and thus require larger dataset to train on. To further investigate this claim, we also calculate the Spearman's correlation coefficient between per-category F1 Scores of models and number of instances available in each category. As shown in Table 8, BERT model achieves the highest positive correlation coefficient with 0.60 which indicates that there is a very strong positive correlation between the amount of data exist for a category and how well BERT performs on that particular category.

6.6 Conclusion

In this chapter, we discussed the task of automatic claim classification, the second step of the proposed workflow for automatic construction of Affiliation Networks. Through our experiments on DebateNet, we evaluated various neural network based models from simple LSTM based models to state-of-the art architectures such as BERT and made following observations: 1) Our experiments showed that attention mechanism is one of the key factors for model the to produce reasonable predictions. It leads to better latent representations by focusing on the words that are important for determining the task. 2) We found that different models with similar overall F-measure can have significantly different performance on classifying particular claims. (BiLSTM+ATTN vs BERT). Our detailed analysis showed that there is no single winner model. Under the circumstances where there is enough data for finetuning, BERT model can be preferable against other models. In other cases, however, BERT might be overkill and BiLSTM+ATTN model can yield better performance. One clear shortcoming of the model architectures sketched in this chapter is that they make the standard assumption of class independence even though we know that two subcategories (e.g. 601:Labour market integration, 604:Guest workers) belong to the same supercategory are more related to each other than categories that belong to other supercategories (e.g. 508:military intervention). We will discuss this issue in detail in Chapter 8.

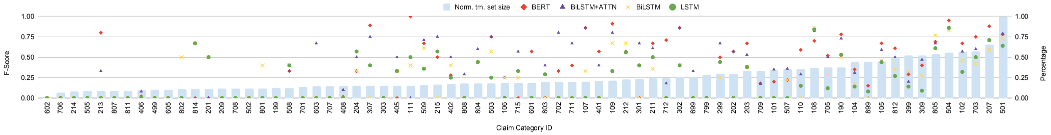


Figure 10: Per-label F_1 score of four claim classification models. Norm. trn. set: Normalized number of training instances per category obtained by dividing the training set size of each category by the largest training set size.

Part III

Evaluating and Improving Fairness of NLP Models for CSS

7 Improving Claim Detection by Addressing Frequency Bias

As we pointed out in Chapter 1, one of the main challenges of computational analysis of political text and statistical machine learning in general is to build fair and robust models since the models are data-driven and usually built on statistical correlations that are sometimes spurious. The spurious correlations are built in the model because those features happen to correlate with a specific class in the training data, and models are likely to rely on them to improve the performance on a specific dataset but it is not desirable that such features to carry predictive power in the model because that will make the model fail on different domains and also make the model biased towards specific groups.

In the upcoming chapters of this thesis, we focus on this challenge and aim to analyze and improve the performances of various NLP models used in CSS in terms of fairness and robustness while maintaining their overall performance. We start with claim detection. In Chapter 5, we described the task of claim detection and investigated neural architectures for automating the task. Following the discussion on the overall performance of our models, we went through an example and showed that providing the same sentence to the system but only changing the actor name in the sentence, the output of the claim detection model varies. We interpreted this as a sign of overfitting of the model to the actor mentions. In this chapter, we investigate this issue in a more systematic way.

7.1 Introduction

The general phenomenon of biased predictive models in NLP, as we discussed in the Background Chapter, is not recent. While neural network based NLP models have shown remarkable results, these achievements have been tempered by the observation that they often produce biased predictions, where we define bias, similar to Friedman and Nissenbaum (1996), as a systematic difference in system performance on one set of instances compared to another. Hovy and Prabhumoye (2021) identify five sources of bias in NLP: the data, the annotation process, the input representations, the models, and the research design; among them the data is considered as the first entry point for bias in the NLP pipeline.

There is a gap between limited variations in a training data and the diversity in real-world languages (Tu et al., 2020). When choosing a dataset to train a model for a particular task, we are also making implicit decisions about which real-world features to include and which features to exclude. Regardless of how the decision is made, this choice leads to the formation of spurious correlations between the chosen features of the data points and their labels, which statistical models will overfit rather than (or in addition to) actually solving the task to maximize their performance. As a result, the performance of the model will vary depending on whether such correlations hold in the test data or not. For example, Gururangan et al. (2018) shows that textual entailment models trained on one of the most widely used textual entailment detection dataset learn that particular keywords imply entailment, irrespective of context. Such a model may perform poorly on the non-entailment examples which include words from the above mentioned keyword as well as entailment examples which do not include any of the keywords.

In this chapter, we perform a similar analysis for our claim detection model introduced in Chapter 5 and find that our model detects claims made by frequently occurring actors with a higher recall than claims made by infrequently mentioned actors although actor frequency is an irrelevant feature for claim detection task. This is worrying, because it means that actors who repeat their claims often will now receive "preferential treatment" and be perceived as even more prominent than they are (Hovy and Spruit, 2016).

In the rest of this chapter, we first provide short description of the claim detection task and the model as reminder. Then, we continue by introducing our analysis method and various computational methods that we use to mitigate the frequency bias in the model. Finally, we present our results and conclude the chapter with a discussion.

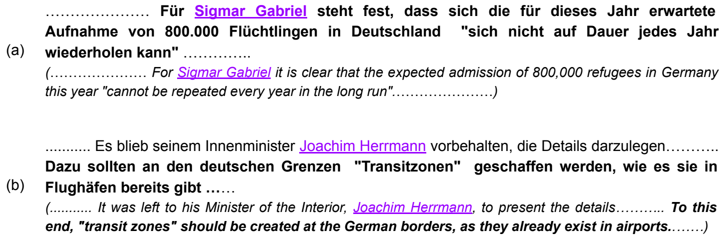
7.2 Claim Detection

Reminder of task. As described in Chapter 4, claim detection is the identification of claim spans in running text where a claim is a statement about certain future actions on the subject of debate, which the actor approves or denies. Note that identification and attribution of actor(s) to the claims is not part of claim detection task, this is taken care of by other sub-components. We frame claim detection as sentence-level classification task in which given an input sequence, the goal is to determine whether it contains a claim (partial or complete) or not which requires a single binary decision per input.

Reminder of model. We use our best performing sentence-level claim detection model described in Chapter 5. It consists of a BERT layer, pretrained solely on German corpora [1] and an output layer with a sigmoid activation function. The final hidden state corresponding to [CLS] token is used as the aggregate sequence representation and fed to the output layer to get the classification probabilities.

Dataset. We perform our experiments on DebateNet using the version released in Lapesa et al. (2020). As described in Chapter 3, this version consists of 960 fully annotated articles from the 2015 Tageszeitung, with a total of 1815 textual claim spans. For about half of the claims (879), the actor is local (i.e., inside the claim); for the rest, it is non-local (i.e., somewhere in the document context). Figure 11 shows example claims with local and non-local actors.

[1] https://deepset.ai/german-bert



(a) ………………… **Für Sigmar Gabriel steht fest, dass sich die für dieses Jahr erwartete Aufnahme von 800.000 Flüchtlingen in Deutschland "sich nicht auf Dauer jedes Jahr wiederholen kann"** …………..

(………………… For Sigmar Gabriel it is clear that the expected admission of 800,000 refugees in Germany this year "cannot be repeated every year in the long run"…………………)

(b) .... Es blieb seinem Innenminister Joachim Herrmann vorbehalten, die Details darzulegen……….. **Dazu sollten an den deutschen Grenzen "Transitzonen" geschaffen werden, wie es sie in Flughäfen bereits gibt** ……

*(.......... It was left to his Minister of the Interior, Joachim Herrmann, to present the details……….. **To this end, "transit zones" should be created at the German borders, as they already exist in airports.**…….)*

Figure 11: Example claims from DebateNet with (a) local and (b) non-local actor. The bold text in the passage indicates the claim. Actors are underlined and highlighted with purple color. Dots indicate the further parts of the news article.

7.3 Frequency Bias in Claim Detection

In order to investigate whether there is a relationship between the claim detection model's performance and occurrence frequency of actors, we first split the actors into three frequency bands using the gold standard actor annotation as shown in Table 9: Actors that appear once in the training data are placed in the Low band, actors that appear 2 or 3 times are placed in the Mid band, and other actors are placed in the High band. Such a configuration allows us to make sure that bands do not contain actors with the same frequency, while at the same time leads to have a good amount of data in each band. Then, we evaluate the performance of our model per actor frequency band which enable us to analyze claim detection performance depending on frequency. During our evaluation, we only analyze recall as there is no gold actor annotation for the false positive claim predictions of the classifier. We also restrict ourselves to the 879 claims with local actors, considering that our classifiers do not work on the document level.

As the last row of Table 9 shows, the prediction quality differs substantially across actor frequency bands: in particular claims made by very infrequent actors show a worse recall (74.5%) than frequent actors (78%). Its sensitivity to actor frequency indicates that the presence of a previously seen actor name is a strong indicator for the presence of a claim. This is surprising given that the claim classifier does not use any explicit actor signal during training. We believe that this is an undesirable situation, since it means that the model extracts a systematically biased set of claims from the corpus: claims made by frequently mentioned actors (such as office holders or spokespersons as shown in cf. Table 2 in Chapter 3) are reinforced, while claims made by infrequently mentioned actors are disregarded. This type of bias can lead to "echo chambers" (Del Vicario et al., 2016) and confers overly high visibility onto frequent actors (Hovy and Spruit, 2016).

Actor frequency band	All	Low	Mid	High
Frequency range	1–48	1	2–3	>3
Number of unique actors	186	85	70	31
Number of claims	879	122	226	531
Model (Standard) recall	77.1	74.5	77.0	78.0

Table 9: Properties of claims with local actors in DebateNet (all and by frequency band) as well as recall of the Standard claim detector

7.4 Debiasing Methods

Next, we present several computational methods based on different techniques for debiasing our claim detection model. As described in Chapter 2, there are two main families of methods to mitigate bias: (1) debiasing methods using data manipulation, and (2) debiasing by adjusting algorithms. In this section, we will describe methods from both families.

7.4.1 Actor Masking

Actor Masking is a data modification method where we mask all referential expressions referring to political actors by replacing the referential expressions with placeholders. In this way we hope to make the model to focus the training signal on the actual task instead of relying on actor frequency which can't be deduced form text anymore. We consider two variants:

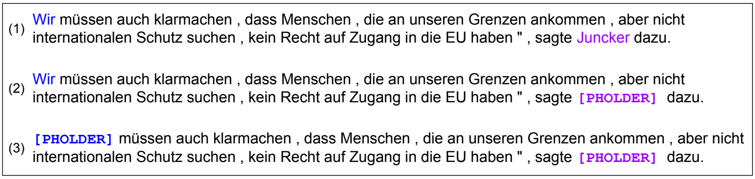
- MASKNAME This model masks the most frequent realization option of political actors, namely proper names of persons using one of the UNUSED tokens of the BERT (BERT reserves ∼ 100 tokens for future usages). We operationalize 'person name' as all phrases marked as `PER` by the SpaCy German Named Entity Recognizer with F-Score of 83.0 on WikiNER.[2] As person names are detected automatically and the NER model used for this purpose is not perfectly accurate, there might still be some information about actors for some cases. However, we believe

[2] Source for model and evaluation figures: `https://spacy.io/models/de#de_core_news_sm`.

this is unproblematic given that the remaining information will be too unsystematic for the model to learn to use.

- MaskNamePron This model masks persons names as above. In addition, it masks all personal pronouns in DebateNet, which can also provide actor information, even though in a more indirect and thus less informative way. It uses the same placeholder. We mask the nine German personal pronuns (ich, du, er, sie, es, wir, ihr, sie and Sie) and all inflected forms of them.

Figure 12 shows an example from DebateNet dataset as well as its modified versions by the two actor masking methods. These masking procedures make it impossible for the claim detector to use information about the actor identity. The motivation is similar to using denoising autoencoders for text representation, which introduce perturbations in the input to encourage models to discover stable latent rather than surface text properties (Glorot et al., 2011).



(1) Wir müssen auch klarmachen , dass Menschen , die an unseren Grenzen ankommen , aber nicht internationalen Schutz suchen , kein Recht auf Zugang in die EU haben " , sagte Juncker dazu.

(2) Wir müssen auch klarmachen , dass Menschen , die an unseren Grenzen ankommen , aber nicht internationalen Schutz suchen , kein Recht auf Zugang in die EU haben " , sagte [PHOLDER] dazu.

(3) [PHOLDER] müssen auch klarmachen , dass Menschen , die an unseren Grenzen ankommen , aber nicht internationalen Schutz suchen , kein Recht auf Zugang in die EU haben " , sagte [PHOLDER] dazu.

Figure 12: Data manipulation example. (1): Original claim from DebateNet (EN: *We* also have to make it clear that people who arrive at our borders but are not seeking international protection have no right of access to the EU," *Juncker* said.) ; (2): Output of MaskName (3): output of MaskNamePron

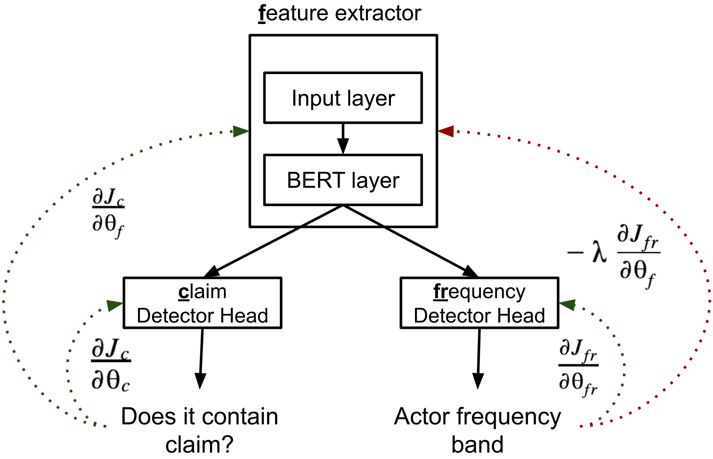

feature extractor
Input layer
BERT layer
$\frac{\partial J_c}{\partial \theta_f}$
$-\lambda \frac{\partial J_{fr}}{\partial \theta_f}$
claim Detector Head
frequency Detector Head
$\frac{\partial J_c}{\partial \theta_c}$
$\frac{\partial J_{fr}}{\partial \theta_{fr}}$
Does it contain claim?
Actor frequency band

Figure 13: Visualization of adversarial debiasing.

7.4.2 Adversarial Debiasing

Adversarial debiasing belongs to the second main debiasing method family, namely algorithm adjustment methods family. As described in Chapter 2, it is first introduced by Zhang et al. (2018) built upon the work of Goodfellow et al. (2014) on Generative Adversarial Networks and aims to have the model learn representations of the input that do not exhibit biases. The same idea has also used by McHardy et al. (2019) to prevent satire detection model from learning publication source characteristics.

We use adversarial debiasing to mitigate frequency bias in our claim detection model. Concretely, we train our model simultaneously to predict whether the given text contains any claim and to prevent the adversarial component from predicting how frequently the claim actor occurs: The adversarial and main components share the feature extractor whose parameters (θ_f) are therefore updated by the gradients coming through the objective functions of both model parts. Formally, let J_c and J_{fr} be the cross-entropy loss functions of the main (claim detector) and adversarial (frequency detector) components, let λ be the meta-parameter for the trade-off between the two losses, and let η be the learning rate. Then the updates are defined as:

$$\theta_c := \theta_c - \eta \frac{\partial J_c}{\partial \theta_c} \tag{7.1}$$

$$\theta_{fr} := \theta_{fr} - \eta \frac{\partial J_{fr}}{\partial \theta_{fr}} \tag{7.2}$$

$$\theta_f := \theta_f - \eta \left(\frac{\partial J_c}{\partial \theta_f} - \lambda \frac{\partial J_{fr}}{\partial \theta_f} \right) \tag{7.3}$$

Eq. 7.1 is used to update the weights of the claim detection model head using the claim detection task labels, and Eq. 7.2 updates the weights of the head of frequency detector. Eq. 7.3 causes the feature extractor to receive the opposite gradients from the two model components, maximizing the loss of the frequency detector. The dotted arrows in Figure 13 indicate the gradient flows and the red color indicates that the feature extractor receives the opposite gradients from the frequency detector during backpropagation.

7.4.3 Sample Weighting

Sample Weighting is an another debiasing method belonging to algorithm adjustment family. Unlike Adversarial debiasing which introduces an additional layer and a loss function to an existing training process, the Sample Weighting approach modifies the model's original loss function: It aims to mitigate frequency bias by punishing model more for false negative predictions on claims by infrequent actors. Each training example is assigned to a weight which reflects the importance of the instance when computing the loss function. Concretely, we introduce three weights (γ_{low}, γ_{mid}, γ_{high}) for the three actor frequency bands from Table 9 and γ_{neg} for negative instances. Parameter updates (i.e.,back-propagation) are performed using scaled loss values:

$$J(x, y) \quad = \quad \sum_{i=1}^{N} g(x_i) * J(x_i, y_i)/N \qquad (7.4)$$

where N is the number of instances in the training set; $g(x_i)$ is set to one of γ_{low}, γ_{mid}, γ_{high}, γ_{neg} depending on x_i.

7.5 Training Details

Similar to Chapter 5, we set the maximum sequence length to 128, batch size to 32 and norm of maximum gradient to 1.5. We use the Adam optimizer with a 3e-5 learning rate. The number of attention heads, hidden layers, and the number of hidden units are set to 12, 12, and 768, respectively. We did not freeze any part of the model and fine-tuned it for 20 epochs with this configuration. Following hyperparameter search, we set the λ parameter of Adversarial to 1.0. Similarly, we set $\gamma_{low} = 0.5$, $\gamma_{mid} = 0.3$ and $\gamma_{high} = 0.2$, and assign $\gamma = 0.1$ to negative instances (i.e. non-claims) in the SampleWeighting.

7.6 Results

In this section, we present our results for four debiasing methods described in Section 7.4 as well as our standard model.

Overall Results. We first investigate the effect of frequency debasing in overall performance. The upper part of Table 10 shows Precision, Recall and F1-Score for five models on the test set of DebateNet. Comparing the two actor masking models (i.e. MaskName and MaskNamePron) with the Standard, we find that the two actor masking models lead to approximately 1 point increase in recall and similar drop in precision. As a result, three models show similar F-Scores without statistically significant differences, indicating that the debiased models perform as well as Standard despite the loss of information in the dataset. Adversarial and SampleWeighting, the two methods belonging to the other family of debiasing methods, have a completely different impact on the claim detector: Both Adversarial and SampleWeighting improve the precision significantly, but suffer a decrease in recall. Nevertheless, we observe that these debiasing models do not lead to significant decrease in terms of overall performance, on the contrary, they can achieve improvements in F-Score, up to 2.7 points.

Frequency Band Analysis. Next, in order to investigate whether these methods help to mitigate frequency bias by improving the

Model Name	Performance for all claims		
	Precision	Recall	F1-Score
Standard	40.1	74.7	52.2
MaskName	39.3	**75.6**	51.7
MaskNamePron	39.8	**75.6**	52.1
Adversarial	**45.5**	69.1	**54.9**
SampleWeighting	42.3	73.5	53.7
Model Name	**Recall on claims with local actors**		
Standard	74.5	77.0	**78.0**
MaskName	80.3	80.1	77.4
MaskNamePron	**81.4**	**82.7**	77.2
Adversarial	77.1	73.5	74.5
SampleWeighting	72.1	79.2	76.3

Table 10: Results for four debiasing methods and standard claim detection model. Left: Precision, Recall and F-Score for all claims. Right: Recall on claims with local actors break down by actor frequency band.

model's performance on the claims made by infrequent actors, we analyze the results by frequency band on the set of local claims for all five methods as we did in Section 7.3. The bottom part of Table 10 shows the recall values. We observe that adversarial debiasing, to some extent, leads to fairer claim detector: It yields better Recall than Standard on low-frequency band. However, this method also causes a significant decrease (around 3.5 points) in recall on high-frequency band. For SampleWeighting, we see a different pattern. This model leads to even worse performance on low band than Standard and increases the performance gap between low and high bands. We think this outcome could be due to the sub-optimal initialization of three γ parameters in the SampleWeighting.

When we look at the data manipulation methods, the improvements in MaskNamePron surpass those of MaskName, which indicates that a more consistent treatment of referring expressions by replacing both proper names and pronouns is advantageous, maybe due to the fact that there is often a relatively free choice between pronouns and proper names (as in Figure 9). We find that both actor masking methods lead to a slight decrease in recall (under 1 point) for actors from the High band. We believe that this is unproblematic, given the redundancy of newspaper reporting: The core claims of debates tend to be mentioned multiple times in an article, and thus not every occurrence must be identified (Blokker et al., 2020). Moreover, we observe that actor masking methods bring about substantial improvements in recall on other two frequency bands: MaskNamePron improves the mid-frequency band Recall by approximately 5 points and the low-frequency band Recall by around 7 points. This is particularly important because it gives claims advanced by infrequent actors a substantially better chance of being recognized by the system. For example, the following claim was recognized by both data manipulation based debiased models but not Standard:

(1) **Der Dresdner Superintendent Christian Behr** ruft zu Nächstenliebe und Dialogbereitschaft auf.

"**Dresden superintendent Christian Behr** calls for charity and readiness for dialog."

We also see improvements for actors realized as general noun phrases which are almost guaranteed to occur infrequently:

(2) **Anwohner und NPD-Politiker** protestierten gegen die geplante Unterkunft.

"**Local residents and NPD politicians** protested against the planned accommodation facilities."

Model Name	Performance for all claims		
	Precision	Recall	F1-Score
STANDARD	19.8	40.4	26.6
MASKNAME	**21.3**	**43.2**	**28.5**
MASKNAMEPRON	20.5	42.2	27.6

Table 11: Cross-domain results for Standard model and data modification based debiasing methods.

Out-Of-Domain Generalization. As both quantitatively and qualitatively shown above, our debiasing methods, especially the data modification based ones, can significantly decrease frequency bias by improving performance on low-frequency band. We believe that this is possibly because, as we stated in previous section, these masking procedures make it extremely difficult for the claim detector to use information about the actor identity, forcing them to rely on more general features of the task. If this is the case, then we expect these methods to also improve, as a side benefit, out-of-domain generalization. To see this, we performed an additional evaluation of our models trained on DebateNet using Akw (Haunss et al., 2013) as the out of domain dataset. Akw is another German corpus for the task of political claims identification, which covers the debate on the future of nuclear energy use in Germany in the four months after the nuclear disaster of Fukushima, Japan in March 2011. The dataset contains 828 articles and 934 claims. Table 11 shows cross-domain results on Akw for our Standard model as well as two data-manipulation based debiasing approaches. The significant decrease in F-scores compared to Table 10 shows that similar to Standard, debiased models also perform badly on the out-of-domain data. Nevertheless, we observe that both MaskName and MaskNamePron outperform Standard in terms of all the evaluation metrics (Precision, Recall, F1-Score). Considering these two results together, we conclude that while the debiasing models used in this section are not sufficient alone for lifting out-of-domain performance of the claim detection models to usable level, they improve generalizability by forcing the model to rely on more general features of the task.

7.7 Conclusion

In this chapter, we systematically analyzed sensitivity of our claim detection model to actor frequency. To do so, we split the actors into three frequency bands using the gold standard actor annotation and evaluated the performance of our model per actor frequency band. We found that our claim detector recognizes claims made by infrequent actors with much worse recall. We then proposed various methods based on data manipulation and algorithmic debiasing to mitigate frequency bias. We compared approaches to mitigating frequency bias in political claims detection and found that a simple data modification strategy does as good as or better than algorithmic debiasing techniques in our case. We found that besides improving recall for infrequent actors without affecting overall performance, actor masking also improves out-of-domain generalization.

Frequency is known to be strongly correlated with performance in machine learning-based NLP, and while we only evaluated the strategy on one task, we believe its benefits carry over to similar tasks. For example, Wang et al. (2021a) show that a sentiment classification model predicts “The film directed by Spielberg is incredibly interesting” as positive. While the prediction is correct, their detailed analysis shows that the main reason for the positive sentiment prediction is the existence of the word "Spielberg", which is a spurious correlation since an annotator won’t label a review as positive just because it mentions "Spielberg". We think that one can adapt the actor masking strategy that we discussed in this chapter easily to mitigate frequency bias in the above-mentioned sentiment classifier model. Also, actor frequency is only one of a large number of potential frequency-related biases. In the next chapters, we will keep discussing frequency-related issues in other CSS tasks.

8 Improving Political Statement Classification with Class Hierarchical Information

In Chapter 6 we discussed the task of automatic claim classification. Our evaluation of various classifiers revealed that claim classification is a challenging task even for state-of-the-art transformer-based classifiers due to the existence of the large number of fine-grained subcategories most of which are infrequently attested and also because they make the standard assumption of class independence even though we know that categories in the codebook are not completely independent as they are arranged hierarchically, with fine-grained subcategories being grouped together into supercategories. For example, ‘border controls’ and ‘quota for refugees’ are subcategories of the supercategory ‘migration control’. In this chapter, we define an ontology of lightweight methods to exploit the hierarchical nature of codebooks by jointly predicting supercategories and subcategories. We use these relations available in the codebooks as prior knowledge to establish additional constraints on the learned model, thus improving performance of the model.

8.1 Introduction

As described in Chapter 3, codebooks, i.e., guidelines that map actual statements or textual passages to the abstract concepts relevant for the respective research, cover a broad variety of research interests as well as text types and they are at the heart of political text analysis. Yet, regardless of whether they have been created to analyze political party manifestos (Volkens et al., 2020), political statements in the European public sphere (Koopmans, 2002), legitimation discourses about political and economic regimes (Nullmeier et al., 2015), or the migration debate in Germany (Padó et al., 2019), they all group their categories of interest into a limited number of supercategories. For example, the codebook of the Comparative Manifesto Project (CMP), which analyzes party manifestos across several countries, includes 7 supercategories (such as *external relations* or *economy*) with 56 subcategories: for *economy*, among others, *free market*, *market regulation*, etc. etc.

Fine-grained, hierarchical schemes help researchers both with data annotation and with analysis. Annotation is often easier when the annotation decision is first based on a supercategory and then on fine-grained subcategories. For analysis, supercategories structure the annotated material according to different levels of abstraction, thereby supporting interpretation and modeling. However, the situation is different when we move to automatic analysis in NLP: due to the large number of fine-grained subcategories, the available data is distributed among many categories. In addition, most categories are infrequently attested, since categories typically show a skewed distribution. As pointed out in Chapter 6, both the limited data issue and skewed distribution of the labels make the task highly challenging and hinder models at achieving strong performances on the fine-grained setup. Hence, many of the existing prediction studies address the task only on the more coarse-grained supercategory level (Glavaš et al., 2017; Subramanian et al., 2018).

In this chapter, we define an ontology of lightweight methods to use the hierarchical structure of political science codebooks to our advantage: knowing that two subcategories (as *free market* and *market regulation*) belong to the same supercategory (*economy*) could lead us to expect that the representations learned for these categories should be more similar to one another than to categories that belong to other supercategories. In this manner, the representations learned for smaller categories can be biased in the right direction by their larger neighbor categories.

In the rest of this chapter, we first define the set of methods to exploit the hierarchical nature of codebooks by jointly predicting supercategories and subcategories (Section 8.2). Crucially, these methods introduce almost no additional parameters, thereby addressing the issues related to the limited amounts of annotated data typically available in CSS studies. Then, we evaluate the proposed methods by carrying out experiments on two datasets with hierarchical codebooks covering single label as well as multi label classification. Our first experiment (Section 8.3) adopts a monolingual multi-label statement classification task. For this, we integrate the lightweight methods into the claim classification models presented in Chapter 6 and evaluate them on DebateNet in terms of both fairness and overall performance. Next, in our second experiment (Section 8.4), we extend our scope in multiple dimensions. At the phenomenon level, we broaden the focus from forward-looking political claims to general political statements by investigating effectiveness of above-mentioned hierarchy encoding methods for classification models on Manifesto dataset. At the experimental level, we now work with single-label classification task involving five different languages. Finally, we conclude this study in Section 8.5.

8.2 Methods for Encoding Hierarchical Structure

As mentioned in Section 8.1, we focus on lightweight methods that introduce a minimal number of additional parameters and are therefore compatible with almost any type of classification model. Hence, through this section, we assume that the base classifier to which we would like to incorporate these methods, uses standard cross entropy loss as the objective function ($\mathcal{L}_{\text{main}}$), and consists of two components: An encoder e(x) which encodes the input, and a classifier c(e(x)) which predicts a single label using softmax activation in the single-label case and a set of labels using sigmoid activation in the multi-label case. The suitable methods are summarized in the taxonomy in Figure 14. We distinguish, from top to bottom: (1) Methods that post-process the output of a statement classifier to enforce hard constraints vs. methods that incorporate soft constraints into the end-to-end learning process; (2) among the latter, methods that decompose the parameters for the more specific classes vs. regularization methods; (3) among the regularization methods, we compare those which target the representation of the class vs. of the encoded instance. We now describe the application of these methods and assess their characteristics which are shown in Table 12.

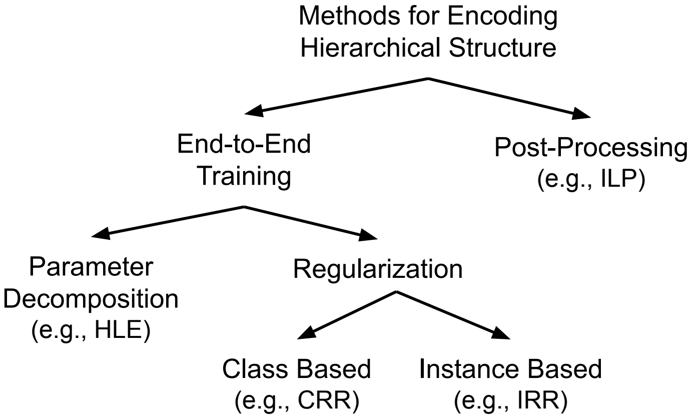

Methods for Encoding Hierarchical Structure
End-to-End Training
Post-Processing (e.g., ILP)
Parameter Decomposition (e.g., HLE)
Regularization
Class Based (e.g., CRR)
Instance Based (e.g., IRR)

Figure 14: Taxonomy of Hierarchical Class Structure Encoding Methods

8.2.1 Post-processing: ILP

Linear Programming is a family of constrained optimization problems where the goal is to optimize a given linear function with respect to a set of linear constraints where optimization means maximization or minimization of linear equations (Schrijver, 1984). An LP specification has two parts, an objective function and constraints. The general form of the objective function is:

	ILP	HLE	CRR	IRR
End-to-end training with the model	-	+	+	+
Imposing hard constrain on the output	+	-	-	-
Applicable to Multi-Label Classification	+	+	+	+
Applicable to Single-Label Classification	-	-	+	+

Table 12: Comparison of hierarchy encoding methods described in Section 8.2.

$$\max : f\left(X_1, \ldots, X_n\right) := y_1 X_1 + \ldots + y_n X_n \tag{8.1}$$

and the general form of the constraints is:

$$a_{i1}X_1 + a_{i2}X_2 + \ldots + a_{in}X_n \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} b_i \tag{8.2}$$

with $i = 1, 2, \ldots m$. X_i are variables, y_i, b_i and a_{ij} are constants. The goal is to maximize (or minimize) a n-ary function f, which is defined as the sum of $y_i X_i$. Integer Linear Programming (ILP) is a sub-type of Linear Programming which introduces the additional constraint that variables can take only integer values. ILP models have already been used in a number of probabilistic models in NLP in order to enforce global constraints. For instance, Punyakanok et al. (2004) use ILP to incorporate global information across arguments such as “arguments do not overlap” or “each verb takes at most one argument of each type.” which is useful information to resolve any inconsistencies of argument classification in Semantic Role Labeling. Similarly, Riedel and Clarke (2006) use ILP to include linguistically motivated constraints into their non-projective dependency parser for Dutch and they observe significant improvement over the state of-the-art parser (McDonald et al., 2005) of that time.

In our application, where a classifier might predict a subcategory with a mismatching supercategory, ILP can select the most likely legal output from the classifier probabilities so that (1) for each predicted subcategory, the matching supercategory is predicted, and (2) for each predicted supercategory, at least one matching subcategory is predicted. For each category we introduce a binary variable v_i indicating if the category is predicted. The objective function is the log likelihood of the model output (including predicted and non-predicted classes), using the estimates of the neural classifiers P_{NC}:

$$\phi_i = P_{\mathrm{NC}}(v_i = 1) \tag{8.3}$$

$$\mathcal{L} = \sum_i \log \phi_i v_i + \log[1 - \phi_i](1 - v_i) \tag{8.4}$$

Let sup(i) denote the supercategory for the subcategory i. Then we formalize constraint (1) as:

$$\text{for each subcat. } v_i : v_i - v_{sup(i)} \leq 0 \tag{8.5}$$

Correspondingly, let subs(i) denote the set of subcategories for supercategory i. Then the second constraint from above is formalized as:

$$\text{for each supercat. } v_i : v_i - \sum_{j \in subs(i)} v_j \leq 0 \tag{8.6}$$

Assessment: In contrast to the other methods introduced in this Section, ILP imposes hard constraints on the output. It does not introduce additional parameters. As our ILP formulation requires probability estimates (i.e. ϕ_i) of the neural claim classifiers for both super- and sub-categories, it is only applicable to multi-label classification. As a post processing step, it does not propagate the errors back into the representations.

8.2.2 Parameter Decomposition: HLE

Hierarchical Label Encoding (HLE), introduced by Shimaoka et al. (2017) for fine-grained named entity recognition, decomposes the representation of each subcategory into a sum of vectors, one for the subcategory itself and one for each of its supercategories. Formally, it creates a binary square matrix, $B \in \{0,1\}^{l \times l}$, where l is the total number of sub- and supercategories. Each cell in the matrix is filled with 1 either if the column class is a subclass of or the same as the row class, and filled with 0 otherwise. The matrix B is not updated during training and integrated into models by multiplying it by the weight matrix W_c of the final fully connected layer of the Base model:

$$W_c^{'} = (W_c^{\top} B) \tag{8.7}$$

where $W_c \in \mathbb{R}^{l \times hs}$, hs is the size of the hidden state of the encoder and W_c' is the modified parameters of the classifier. In this way, we introduce label parameter sharing between labels in the same hierarchy: The weight vector of each subcategory i in layer final fully connected layer becomes the sum of the label parameter of subcategory i and its parent category.

Assessment: HLE imposes soft constraints and does not introduce any parameters. Similar to ILP, HLE can only be used in multi-label classification.

8.2.3 Class Representation Regularization

Class representation regularization (CRR) falls under the umbrella of regularization methods. Regularization methods have been previously used for similar purposes in NLP such as enforcing SVM based hierarchical medical image classification models to learn that parent and children labels within the hierarchy are similar to each other (Naik and Rangwala, 2015); integrating external knowledge sources for enhancing embedding learning (Song et al., 2017); connecting neural networks based background subtraction methods to knowledge bases (Bui et al., 2018; Stretcu et al., 2019) and inserting prior knowledge such as sentiment lexicon and relative position information in Emotion Cause Analysis (Fan et al., 2019).

In our case, the goal is to increase the similarity between the weight vectors of the subcategories belonging to the *same* supercategory while keeping the weight vectors of subcategories *across* supercategories dissimilar. Formally, the classification layer is a weight matrix $W_c \in \mathbb{R}^{l \times hs}$, where l is the number of classes and hs is the output size of the encoder. We use S for the set of supercategories and S_i to denote the i-th supercategory, the set of its subcategories, and their weight vectors, depending on context. Then we define the centroid $\mu(S_i)$ of a supercategory, the average distance between two supercategories, d_{avg}, and the global intra- and inter-supercategory distances $d^{\text{inter}}/d^{\text{intra}}$ as:

$$\mu(S_i) = \frac{1}{|S_i|} \sum_{w \in S_i} w \tag{8.8}$$

$$d_{\text{avg}}\left(S_i, S_j\right) = \frac{1}{|S_i||S_j|} \sum_{\substack{w \in S_i, \\ w' \in S_j}} \text{dist}(w, w') \tag{8.9}$$

$$d^{\text{inter}} = \sum_{0 \leq i < j \leq |S|} d_{\text{avg}}\left(S_i, S_j\right) \tag{8.10}$$

$$d^{\text{intra}} = \sum_{i=1}^{|S|} \frac{1}{|S_i|} \sum_{w \in S_i} dist(\mu(S_i), w) \tag{8.11}$$

Finally, we regularize the learning objective ($\mathcal{L}_{\text{main}}$, cf. Section 8.2) as follows:

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \alpha d^{\text{intra}} - \beta d^{\text{inter}} \tag{8.12}$$

where the hyperparameters $\alpha, \beta \geq 0$ control regularization strength.

Obviously, this is not the only way to do regularize the loss function in order to make the weight representations of the subcategories from the same supercategory more similar to each other and increase their dissimilarity to the representations of other subcategories. As a simplest variation, one can try to combine the components of the loss, $\mathcal{L}$, differently such as $\mathcal{L} = \beta\mathcal{L}_{\text{main}}/d^{\text{inter}} + \alpha d^{\text{intra}}$. Results of preliminary experiments, however, showed worse results for this variant, possibly because one part of the regularization, d^{inter} is multiplied with the main loss, but the other part is not which makes the regularization terms asymmetrical overall. A more pronounced change would be to come up with logical rules that define the hierarchical structure of the target dataset, and integrate them in the form of additional terms in the loss function, in place of or in addition to the distance-based regularization terms. Although this has been demonstrated by Roychowdhury et al. (2021) as an effective way to encode information into the weights of a neural network in the context of image classification, an obvious drawback of this variation is that it requires developing logic rules that can describe the hierarchy.

Assessment: CRR imposes soft constraints, adds two hyper parameters, and is applicable to both single and multi label classification.

8.2.4 Instance Representation Regularization

Instance representation regularization (IRR) applies the same intuition as above, but at the level of the instance representations produced $e(x)$ by the encoder. The model is penalized whenever the encoder generates more similar representations for input pairs with different supercategories than for pairs with the same supercategories. A similar idea has been explored by Choi and Rhee (2019) for image classification and image reconstruction tasks in which authors design a regularizer that target to reduce the covariance of representations calculated from the same class samples for encouraging feature independence.

Formally, let X be the set of instances, and $s(x)$ be the supercategory of instance x. We consider the set of instance triplets where the first and second member share a supercategory and the third has a separate one, and measure the extent to which the distance across supercategories exceeds the distance within the supercategory:

$$d^{\text{diff}} = \sum_{\substack{x,y,z \in X \\ s(x)=s(y) \\ s(x) \neq s(z)}} \begin{aligned}&\max(0, \text{dist}(e(x), e(y)) \\ &\quad - \text{dist}(e(y), e(z)))\end{aligned} \tag{8.13}$$

We then regularize the learning objective as:

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \alpha \cdot d^{\text{diff}} \tag{8.14}$$

where $\alpha \geq 0$ controls the regularization strength. Since using the complete set of triples is computationally demanding, it may be necessary to sample instead. Therefore, we create triples from each mini-batch by combining its instances, which is an approximation to uniform sampling.

As in the case with other methods presented in this section, IRR can also be implemented in more than one way. For example, one may consider to punish absolute distances (as in Section 8.2.3.) as opposed to us choosing to introduce a "contrastive" regularizer and punish differences between distances by following Choi and Rhee (2019). As another alternative, instead of using the final output of encoder e(x), we could use representations from lower layers of the encoder. However, we do not expect this to perform better because as shown by previous work (Tenney et al., 2019), while the basic syntactic information appears earlier in the network, semantic features, which we need for our purpose, appear at the higher layers. **Assessment:** IRR also imposes soft constraints, adding one hyperparameter. IRR requires each instance to belong to a single supercategory.

Discussion. Before moving on to the experiments, we would like to clarify a point regarding the hierarchy encoding methods presented in this section: Although we formulate our hierarchy coding methods based on a two-level hierarchy, these methods can be easily generalized to datasets with hierarchies deeper than two levels with very minor changes or none at all in the case of HLE method introduced in Section 8.2.2. For the integer linear program and regularization based methods, the main design choice that needs to be decided is how to incorporate the relationship between a node and its parents since, unlike the two-level hierarchy where each category can have at most one super-category, in deeper hierarchies sub-categories can have more than one super-category, exactly, each category has d super-categories, where d is equal to its depth in the hierarchy. In this scenario, one can decide to take into consideration only the relationship between the sub-category and its most general super-category and as a result use the same formulae, or alternatively, one can try to also include intermediate relationships in the hierarchy into the network, which may provide an extra regularization signal but at the same time too many regulators may also introduce instability (Zhou et al., 2018; Han et al., 2021).

8.3 Experiment-1: Newspapers

We conduct our first experiment on DebateNet consisting of multi-labeled political claims where the claim is defined as any form of politically motivated demand or action (both verbal and non-verbal) of deliberate actors (Koopmans and Statham, 1999).

8.3.1 Experimental Setup

Dataset. As described in Chapter 3, DebateNet is annotated manually according to a two-level ontology for the migration domain, comprising 8 supercategories with 118 subcategories. There is a total of 3827 annotated textual spans. Again, as described in Chapter 6, we randomly sample 90% of our dataset for training and evaluate on the other 10% and we reduce the number of subcategories by combining subcategories with less than 20 instances under the preexisting subcategory x99, which exists for each supercategory as a 'catch-all' category for outlier cases. We acknowledge that that makes the catch-all subcategories are presumably challenging to learn, given their inhomogeneous nature, but we believe that this strategy is reasonable, since no instances are discarded in this manner, and they still retain the supercategory signal that we are interested in. Table 13 (duplicate of Table 6 in Chapter 6) shows the number of categories in each subcategory before and after filtering operation.

Base Classifiers. As the Base classifier, we use the four models described in Chapter 6, with no change: LSTM, BiLSTM, BiLSTM+ATTN and BERT. The detailed model descriptions and training details including hyperparameter selection can be found in Section 6.3 and Section 6.4.2 respectively.

Encoding Methods As IRR is not applicable to multi-label classification (cf. Section 8.2.4), we leave it out and experiment with eight model variations: Base; ILP, HLE and CRR; and the combinations HLE+ILP, HLE+CRR, CRR+ILP and HLE+CRR+ILP. We use Euclidean distance as distance metric in CRR based on our prelimi-

Code	Supercategory Label	Freq.	Number of subcats.	Reduced number of subcats.
1xx	Controlling Migration	998	16	12
2xx	Residency	726	18	12
3xx	Integration	475	15	6
4xx	Domestic Security	230	9	6
5xx	Foreign Policy	689	9	8
6xx	Economy	194	12	5
7xx	Society	749	19	12
8xx	Procedures	676	20	11

Table 13: Claim distribution for each supercategory before and after removal of extremely infrequent subcategories. Freq.: Frequency, number of textual spans in each supercategory. Total number of subcategories:118; Total reduced number of subcategories:72.

nary experiments. We perform hyper-parameter search in BERT to optimize parameters of encoding methods with the following lower- and upper-bounds α_{CRR}: [0.005,0.6], β: [0.01,0.6] and set both α_{CRR} and β to 0.01.

8.3.2 Results

Main results. Table 14 summarizes the main results of our experiments. As we discussed in detail in Chapter 6, LSTM and BiLSTM perform significantly worse than BiLSTM+Attention and BERT in the 'Base' setting. The addition of ILP (2nd row) and CRR (3rd row) to the Base classifier produces similar results. They lead to inconsistent changes in precision but always yields better Recall and F-Scores. LSTM and BiLSTM still perform significantly worse than the other two models. When we switch to HLE, all metrics for all

Method	LSTM			BiLSTM			ATTN			BERT		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Base	49.6	23.9	29.8	51.1	26.2	32.1	67.3	38.5	46.2	61.2	41.9	47.0
ILP	45.1	28.5	32.2	50.7	33.0	37.9	63.1	41.6	47.8	56.0	49.7	50.4
CRR	46.8	29.4	34.1	50.8	29.6	33.2	62.9	41.2	47.1	70.4	49.0	55.2
HLE	60.1	33.0	39.0	56.6	30.0	36.4	68.7	40.6	48.0	75.2	52.2	59.0
HLE+ILP	51.9	38.1	**41.2**	62.6	41.8	**47.6**	65.8	46.1	**50.8**	65.8	59.0	60.5
HLE+CRR	59.4	33.7	40.1	58.5	32.0	37.6	68.8	40.3	48.0	76.5	54.3	**60.8**
CRR+ILP	51.2	30.2	34.8	53.1	31.6	35.8	62.7	42.1	47.4	66.0	55.4	57.8
HLE+CRR+ILP	50.4	36.2	39.7	50.3	38.1	41.2	60.5	42.9	48.1	64.3	57.3	58.6

Table 14: Precision, Recall, F-Scores for multi-label statement classification on the DebateNet Dataset.

models are boosted significantly, showing that parameter sharing via the super/sub-category co-occurrence matrix is a successful across the board. We observe the largest improvement for BERT, where HLE yields an improvement of 12 points in F1.

Rows 5-7 in Table 14 present results when the two of the encoding methods are used in combination. HLE+ILP models consistently improve over both the HLE only and ILP only setting. Specifically, HLE+ILP models achieves better Recall scores than HLE models (+7 points on average) and better Precision (+8 points on average) scores than ILP models. HLE+ILP also leads to best performance in terms of F1 Score for all models except BERT, for which slightly better F1 Score is achieved with HLE+CRR. By comparing HLE+CRR with HLE across models, we find that HLE+CRR improves only slightly over HLE. This intuitively shows that having similar representations for categories from the same supercategory, either via parameter sharing or regularization, does most of the work, and explicitly pushing away weight vectors of subcategories with different supercategory does not add too much on top of that. Also, we observe that unlike HLE, CRR leads to instability when it is combined with ILP: Depending on the base classifier architecture, CRR+ILP leads to better or worse performance than CRR only and ILP only. Lastly, we see that HLE+CRR+ILP does not achieve the best overall F-score on any models indicating that there is a sweet spot in the number of hierarchy encoding methods that can maximizes the models' performances.

In short, the main conclusions that can be drawn from Table 14 is that: 1) In the base setting, as we already discussed earlier in detail in Chapter 6, LSTM and BiLSTM perform significantly worse than BiLSTM+Attention and BERT. 2) We find that besides simplicity, these encoding methods are highly effective to improve the model performance. Especially, combining HLE and ILP leads to large performance improvements regardless of the Base classifier architecture. 3) We previously pointed out that (cf. Section 6.6) one of the main reasons BERT can't outperform ATTN by large margin, although its large model capacity, is the relatively small size of DebateNet. We see that with the help of these methods, Bert achieves F1 scores close to 61, ∼13 points improvement over the Base Bert and 10 points improvement over the ATTN model. This shows that these encoding methods help big models to use their capacity under limited data.

Frequency Band Analysis. As discussed in the introductory section, fine-grained classification struggles in particular with infrequent classes. We therefore ask how hierarchical class structure affects performance in relation to frequency. To do so, we split the fine-grained categories, as we did in Chapter 7, into three equal-sized frequency bands[1], and analyze the performance of ATTN and BERT, our two best performing models. The results in Table 15 show that the prediction quality of the Base BERT and Base ATTN models differ significantly across frequency bands, meaning that these models exhibit strong frequency bias: They fail badly in the low freq band while doing a fair job in the mid and high bands.

The inclusion of hierarchical information leads to improvement

[1] Thresholds: high-frequency (265≥f≥ 67), mid-frequency (65≥f≥ 40) and low-frequency (20≥f≥ 39). Subcategory - frequency band assignments can be found in the Appendix B.

Model	Freq band	Base			ILP			HLE			CRR		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
ATTN	Low	10.5	6.5	7.5	22.8	12.9	14.4	16.9	9.7	10.5	10.5	10.0	10.2
	Mid	81.6	46.5	**56.2**	69.5	44.7	52.9	71.0	45.6	53.2	64.7	49.5	52.6
	High	73.7	42.1	50.4	69.1	46.5	52.8	78.6	45.1	53.8	65.9	48.4	52.4
BERT	Low	10.2	9.7	9.6	18.3	14.5	14.8	58.3	30.6	37.4	31.2	16.1	18.7
	Mid	58.0	36.0	41.8	65.0	47.4	50.4	77.4	55.3	62.2	75.8	49.1	55.8
	High	73.1	50.8	56.7	60.5	57.9	57.9	77.8	55.6	62.3	76.4	55.9	62.6

Model	Freq band	HLE+ILP			HLE+CRR			CRR+ILP			HLE+CRR+ILP		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
ATTN	Low	24.5	11.3	13.4	19.4	11.3	12.9	13.6	10.2	10.9	28.8	14.5	**17.3**
	Mid	63.9	51.8	55.6	72.0	41.2	50.5	65.6	50.1	53.5	63.0	45.6	51.4
	High	75.1	51.2	**56.9**	76.3	44.4	52.8	64.1	43.9	51.2	66.2	47.8	53.3
BERT	Low	48.1	30.6	34.8	54.8	29.0	35.8	35.5	19.4	21.9	52.2	33.9	**38.3**
	Mid	71.5	63.2	65.1	85.1	58.8	**66.2**	74.3	58.8	61.5	71.9	62.3	64.0
	High	67.3	63.3	**64.0**	77.7	57.9	**64.0**	69.1	61.6	63.8	63.9	60.3	60.8

Table 15: Precision, Recall, F-Scores for the DebateNet Dataset broken down by category frequency bands.

for all bands in general while the most substantial improvement is achieved for the low-frequency band. This is particularly important, as it means that besides lifting the fine-grained claim classification models to a usable level, these methods also reduce frequency bias in the systems by diminishing the performance gap across frequency bands. Adding HLE, CRR and ILP together to the model works best for the low frequency band. This configuration improves low frequency band F-Score by 10 and 28 points for the ATTN and BERT models, respectively. For mid and high frequency bands, however, dual configurations yield better results than HLE+CRR+ILP: HLE+ILP outperforms HLE+CRR+ILP in ATTN by 4 points on average. Similarly, BERT with HLE+CRR achieves best results, outperforming HLE+CRR+ILP model with 3 F-Score points gain on average. The modest improvements in results for higher frequency bands indicate that the more data available, the less gain is provided by external hierarchy encoding methods. Figure 15 shows the subcategories with the highest improvement: four belong to the mid- frequency and three to the low-frequency band. To investigate this further, we also perform a correlation analysis between amount of available data for each subcategory and change in model performance which is measured as the difference between the model's performance in Base and the configuration in which it has the best overall performance. High negative Spearman's correlation coefficients, as shown in Table 16, confirm the negative relationship between the frequency and the size of the improvement.

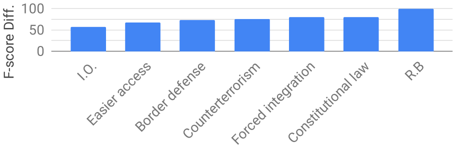


Figure 15: Subcategories with highest F1 increase. I.O:Integration Offers, R.B:Reducing Bureaucracy

In sum, our frequency band analysis shows three things: 1) Base models, both LSTM and ATTN perform significantly worse on less frequent categories than on the rest of the corpus; 2) By using simple methods that encode the hierarchy available in the codebook, it is possible to improve the classification quality and reduce the

PAIR	ATTN	BERT
Base - (HLE+ILP)	-0.19	-0.29
Base - (HLE+CRR)	-0.18	-0.26

Table 16: Spearman's correlation coefficients between subcategory size and improvement in F_1 score of best performing models

frequency bias by stabilizing models' prediction quality across frequency bands; 3) The best configuration for hierarchy encoding may vary depending on the architecture of the Base model and the target frequency band although we observe that combining HLE and ILP works fairly well all the time. The parameter sharing introduced by HLE particularly helps the lowest-frequency categories and increases both Precision and Recall. ILP generally boosts Recall by enforcing that both super- and a subcategories need to be predicted.

8.4 Experiment-2: Party Manifestos

Our second experiment targets political statements in party manifestos. Manifestos are official documents issued by parties to summarize their political program and unlike statements in newspapers that are issued only by the individual members of a party, they are authoritative for the entire party. Sentences in manifestos and newspapers are linguistically quite different. Manifestos are written in direct speech and express a party's position; on the contrary, statements from public (newspaper) discussion contain indirect speech attributable to an individual actor. Also, they make different lexical choices to describe similar events, e.g. colloquial ("sent back") vs. technical ("deport") terms in the case of a discussion about immigration topic (Blokker et al., 2020).

8.4.1 Dataset

We build on the Comparative Manifesto Project (Volkens et al., 2019), which manually coded manifestos from multiple countries and languages to create a resource for the systematic content analysis. Considering the availability of language specific transformer-based models and large annotated data, we focus on 5 countries with one language each: Finland (Fi), Germany (De), Hungary (Hu), Turkey (Tr) and United Kingdom (En) and collected all available manually topically-coded manifestos. Note that this is not a parallel corpus, and the amount of annotated data available for each language varies greatly. The codebook uses a two-level ontology of 7 policy areas as supercategories “designed to be comparable between parties, countries, elections, and across time”, and 56 subcategories. Sentences are split into segments if they discuss unrelated topics or different aspects of a larger policy, so each segment is assigned a single subcategory. The distribution of statements over the manifesto topics in the dataset is shown in Table 17. We split the dataset into train (65%), validation (15%), and test (20%) portions.

8.4.2 Experimental Setup

We model statement classification in the Manifesto corpus at the segment level as a single-label classification task. Unlike in Section 4.1, we do not apply any pre-processing to merge very infrequent

Supercategory	Fi		De		Hu		Tr		En	
	freq	#sub	f	#sub	f	#sub	f	#sub	f	#sub
Freedom, Democracy	758	4	5,672	4	3,553	4	5,211	4	2,091	4
External Relations	1,599	10	5,727	10	2,288	9	3,721	10	3071	10
Political System	1,129	5	5,661	5	4,040	5	3,299	5	2,530	5
Fabric of Society	2,677	8	6,095	8	5,500	8	5,555	8	3,328	8
Social Groups	2,113	6	5,865	6	3,625	6	5,157	5	2,075	6
Economy	4,556	15	15,185	16	10,380	16	17,899	16	6,753	15
Welfare, Life quality	7,787	7	16,592	7	15,121	7	11,120	7	10,246	7
Overall	20,619		60,797		44,507		51,962		30,094	

Table 17: Subcategory distribution by supercategories in the complete (100%) Manifesto dataset: frequency (*f*); number of subcategories (*#sub*). Overall: Number of instances per language.

subcategories, since almost all categories in the Manifesto corpus have more instances than threshold value (i.e. 20). For example, there is only one subcategory with less instances than the 20 in the DE portion.

Varying Training Data Size. With several hundred thousand sentences after years of annotation, the Manifesto corpus is one of the largest CSS datasets available and its size is arguably larger than typical for CSS projects. As an example, annotation of the 4k DebateNet instances took more than a year. For this reason, we introduce an experimental variable, namely the amount of the training data. Specifically, random draws of percentages (25%, 50% and 100%) of the full training set, keeping the test set constant. This allows us to do two things: 1) we can demonstrate the efficacy of hierarchy encoding models for a standard-size CSS dataset using 25% version, and 2) we can study the impact of varying amount of training data on hierarchy encoding methods.

Encoding Methods. Since HLE and ILP are only useful for multilabel classification, we experiment with the following model variations: **Base**; **CRR**, **IRR**; and **CRR+IRR**. As distance metric, we use Manhattan distance in CRR and Cosine distance in IRR as other choices led to worse results during our preliminary experiments.

Base Classifier. We continue with the best performing classifier model from Section 8.3, which is a standard pre-trained and finetuned BERT model. The only difference between the BERT model in Section 8.3 and here is that since each segment in Manifesto dataset

Lang	Train	lr	α_{CRR}	β	α_{IRR}	dp
Fi	25%	3e-5	0.1	0.1	0.1	0.4
	50%	2e-5	0.05	0.05	0.1	0.2
	100%	2e-5	0.05	0.05	0.1	0.2
De	25%	2e-5	0.2	0.2	0.4	0.2
	50%	2e-5	0.05	0.01	0.2	0.2
	100%	2e-5	0.1	0.2	0.1	0.1
Hu	25%	2e-5	0.4	0.05	0.1	0.2
	50%	2e-5	0.1	0.1	0.1	0.2
	100%	2e-5	0.01	0.01	0.05	0.2
Tr	25%	2e-5	0.2	0.2	0.4	0.2
	50%	2e-5	0.2	0.4	0.05	0.2
	100%	2e-5	0.01	0.01	0.1	0.2
En	25%	3e-5	0.05	-0.05	0.1	0.4
	50%	3e-5	0.2	0.2	0.4	0.4
	100%	3e-5	0.05	0.05	0.4	0.4

Table 18: Hyperparameters of CRR+IRR models. $\alpha_{\text{CRR/IRR}}$: α parameter of CRR/IRR.

is assigned to a single subcategory, we replace the sigmoid activation function with a softmax in the classifier, c(e(x)). For each language (Fi[2], De[3], Hu[4], Tr[5] and En[6]), we use a cased BERT variant that was trained specifically for the target language. Similar to Experiment-1, we again use AdamW as the optimizer, cross-entropy as the loss function. After performing a hyperparameter search on the develop-

[2] https://github.com/TurkuNLP/FinBERT

[3] https://deepset.ai/german-bert

[4] https://hlt.bme.hu/en/resources/hubert

[5] https://github.com/dbmdz/berts

[6] https://huggingface.co/bert-base-cased

ment set, we set the parameters for each language and training set as shown in Table 18.

8.4.3 Results

Tables 19, 22 and 23 show the overall results of the models with varying training data sizes (25%, 50% and 100%) on the Manifesto dataset. We discuss the effects of encoding methods based on our results on 25% data (cf. Table 19) since as stated above, the complete manifesto corpus is much larger corpus than many of the dataset used in CSS. When we investigate the impact of varying amount of training data on hierarchy encoding methods however, we consider all three data configurations (25%, 50% and 100%) together.

Lang	Base			CRR			IRR			CRR + IRR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Fi	39.0	38.4	37.4	40.6	40.0	39.3	41.5	39.2	38.6	42.2	40.8	**40.1**
De	33.3	31.3	31.4	35.4	34.1	34.2	34.6	34.7	34.3	36.8	34.8	**34.9**
Hu	41.1	38.8	38.7	41.7	39.8	39.7	42.2	39.0	39.2	43.7	39.3	**39.8**
Tr	45.6	42.5	42.4	47.9	41.7	43.0	48.9	42.4	43.3	49.0	42.5	**43.6**
En	31.5	30.8	30.5	34.6	32.5	32.3	32.7	32.7	32.1	34.4	32.5	**32.8**

Table 19: Results for the Manifesto dataset trained on 25% of the data.

Main Results. Similar to Section 8.3.2, we first analyze whether the hierarchical information improves overall performance. As shown in Table 19, the results are surprisingly similar across all languages, despite the typological differences and varying amounts of training data. The Base model consistently yields the worst results, in line with the findings of Experiment 1. The use of hierarchical structure, both through CRR and IRR, improves performance for all languages, leading to an improvement of up to 3 points in F-score. We observe that there is no clear winner between CRR and IRR methods: While CRR improves performance more than IRR in Fi, Hu, and EN; IRR slightly outperforms CRR in De and Tr the two languages with the largest data size. This intuitively makes sense because IRR operates on the level of instance representations and when the dataset size drops below certain threshold IRR doesn't encounter with enough unique input pairs to penalize the model and use the same input pairs repeatedly which can lead to sub-optimal performance. Next, as was the case in Experiment 1 for CRR+HLE, we see that the two methods can be beneficially combined: CRR+IRR yields the highest F-Score for each language: the gains over Base are between 1.1 points (Hu) and 3.5 points (De). This shows that although both CRR and IRR are regularization based methods, they affect the way the model performs differently.

Frequency Band Analysis. After analyzing the effect of hierarchy encoding methods on the overall performance, we continue by discussing how hierarchical structure and category frequency interact. As in Experiment 1, we analyze the impact of hierarchical structure on three equal-sized subcategory frequency bands[7], shown in Table 20, for the 25% condition. Similar to Experiment 1, the prediction quality of Base model differs significantly across frequency bands: It performs moderately on the mid and high frequency bands

[7] Threshold values for each frequency band and category-frequency band assigments can be found in the Appendix B

Lang	Freq	Base			CRR			IRR			CRR + IRR		
	band	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Fi	Low	18.4	15.2	13.7	20.7	17.7	16.7	22.6	16.6	15.4	25.5	19.6	**19.5**
	Mid	42.1	42.2	41.5	42.5	42.6	41.9	44.4	42.7	42.7	43.9	43.9	**43.0**
	High	56.6	57.8	57.0	58.7	59.8	**59.2**	57.4	58.4	57.7	57.3	58.9	57.9
De	Low	16.1	9.0	10.6	19.7	14.7	16.4	18.6	17.7	17.8	23.1	16.2	**18.0**
	Mid	36.9	38.3	37.4	38.3	40.3	38.7	37.3	40.8	38.5	38.7	40.5	**38.9**
	High	48.7	48.9	48.5	49.9	49.4	49.3	49.6	47.6	48.4	50.1	49.7	**49.7**
Hu	Low	24.5	15.4	17.3	26.4	18.4	19.9	28.4	16.9	19.1	33.6	17.5	**21.1**
	Mid	41.5	43.7	41.7	41.5	43.8	**42.1**	41.0	43.5	41.6	40.1	42.7	40.9
	High	57.3	57.2	57.0	57.2	57.2	57.0	57.3	56.7	56.7	57.3	57.7	**57.4**
Tr	Low	29.2	19.6	20.2	37.4	20.8	24.2	40.4	22.2	**24.9**	38.0	21.0	23.8
	Mid	46.4	47.3	**46.6**	45.8	43.2	44.1	46.0	44.1	44.8	48.8	44.9	46.4
	High	61.1	60.6	**60.7**	60.4	61.0	60.6	60.1	60.8	60.1	60.3	61.5	**60.7**
En	Low	13.3	8.3	9.7	20.1	10.8	12.9	14.6	10.7	11.9	17.2	11.3	**13.3**
	Mid	30.5	31.7	30.6	32.1	34.7	32.5	32.0	34.9	32.8	33.7	33.1	**32.9**
	High	50.7	52.4	51.3	51.7	52.0	51.6	51.6	52.3	51.6	52.3	53.2	**52.2**

Table 20: Precision, Recall, F1 scores for the Manifesto dataset trained on 25% of the data and broken down by category frequency bands.

while failing badly on the low frequency band with F1 between 9.7 (En) and 20.2 (Tr). The results also reveal that while the CRR only and IRR only configurations can improve the prediction quality, the highest improvement for this frequency band is obtained with the combination CRR+IRR, leading to improvements between 3 and 7 points F1 for all languages except Turkish in which IRR leads to slightly better F1 score (0.1 points). In the higher frequency bands, on the other hand, we notice that there is a higher variance: Depending on the language and frequency band the best F1 score can be achieved by CRR+ILP (7 cases) or CRR (2 cases). The gain can be up to 2.3 points F1 for the mid-frequency and 1.2 points F1 for the high-frequency band. These results indicate that as in Experiment 1, the gains are more modest on higher frequency bands.

This observation is supported by a correlation analysis that shows a significant negative correlation between subcategory size and the F1 improvement of CRR+IRR over Base, $p = -0.21$. Furthermore, Figure 16 shows the 7 subcategories with the largest improvement in F1: Three of them belong to the mid-frequency band, four to the low-frequency band, and none to the high-frequency band.

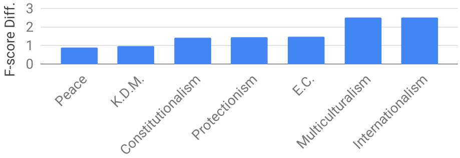


Figure 16: Seven subcategories with highest F1 increase for best model compared to base model. K.D.M: Keynesian Demand Management, E.C: European Community/Union. Peace, E.C and Protectionism belong to mid frequency class. The other four subcategory belong to low band.

We also look at some of the statements in the English part of manifesto corpus which were classified incorrectly by the Base model and correctly by the IRR+CRR model, as shown in Table 21: The fact that all these example involve arguably related subcategories shows that (1) the Base model is able to predict the correct supercategory but fails to predict the subcategory as a result of its tendency to choose the more frequent subcategory over the less frequent one; (2) our encoding methods are useful to counteract this substitution between more frequent and less frequent categories.

Input	Base Pred. (incorrect)	CRR+IRR Pred. (correct)
Our long-term economic plan is turning around Britain's economy.	Economic growth (Mid)	Economic planning (Low)
Face coverings such as these are barriers to integration.	National way of life (Mid)	Multiculturalism (Low)
Fairer corporate governance, built on new rules for takeovers executive pay and worker representation on company boards.	Market regulation (High)	Corporatism (Low)
This sent out terrible signals: if you did the right thing, you were penalised — and if you did the wrong thing, you were rewarded, with the unfairness of it all infuriating hardworking people.	Equality (High)	Welfare limitation (Low)

Table 21: Examples from Manifesto dataset correctly classified only by CRR+IRR.

Corpus size and hierarchical structure. As stated above, we also study the impact of varying amount of training data on hierarchy encoding methods using our results on 25%; 50% and 100% conditions. Figure 17 shows the mean improvement in F1 between Base and IRR+CRR for each language and data condition. We see that the improvement is largest for the 25% setting, which supports our previous finding that incorporating hierarchical information into the models is especially important in a low data regime.

That being said, we still obtain improvements for the 50% condition, as shown in Table 22: For most of the languages, using CRR or IRR alone lead to increase in performance of the Base model. Combining CRR and IRR further increases performance, as in 25%

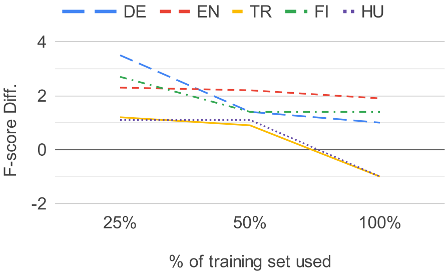


Figure 17: Change in F1 between the CRR+IRR and Base across training data sizes

Lang	Base			CRR			IRR			CRR + IRR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Fi	43.8	43.4	42.5	44.3	42.7	42.5	43.7	42.5	42.2	45.8	43.8	**43.9**
De	37.7	37.8	37.1	39.4	37.9	38.1	38.6	37.7	37.7	40.0	38.0	**38.5**
Hu	42.1	40.0	40.1	43.4	40.8	41.1	43.0	39.4	39.9	44.9	40.7	**41.2**
Tr	50.9	46.5	47.1	49.9	46.9	47.2	52.9	48.6	**49.2**	51.8	47.7	48.0
En	33.4	31.9	32.0	34.9	33.8	33.8	33.0	32.6	32.1	35.4	34.9	**34.2**

Table 22: Results for the Manifesto trained on 50% of the data

condition. CRR+IRR always yields better F1-Scores than Base (up to 2.2 F1 points) although the gap between performance of the Base and CRR+IRR is less pronounced under 50% training data case. These results show that both CRR and IRR are effective methods and can improve political statement classification models' performance even on the fairly large datasets.

On 100% condition (cf. Table 23) we see a pattern that highly differs from 50% condition, with much larger cross-lingual differences. While these methods lead to 1-2 points improvement in some lan-

Lang	Base			CRR			IRR			CRR + IRR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Fi	47.0	48.1	46.7	48.1	48.7	47.8	47.1	48.3	46.9	47.6	51.2	**48.1**
De	40.4	40.9	40.2	41.3	41.2	40.9	41.8	40.0	40.2	42.4	40.8	**41.2**
Hu	47.8	43.9	**44.6**	45.0	41.4	42.3	47.1	42.8	43.8	43.4	45.0	43.6
Tr	56.7	55.7	**55.5**	56.4	54.2	54.3	55.6	53.9	53.6	55.9	54.6	54.5
En	38.5	35.7	35.9	40.2	36.3	37.2	37.8	36.1	36.5	38.4	38.2	**37.8**

Table 23: Results for the Manifesto trained on 100% of the data.

guages such as En, Fi and De, they yield worse F1 score than Base in Tr and Hu. We believe these encoding methods are still useful, to some extend, for En and Fi since they are the two languages with the least amount of data in our experiments. What is surprising is to see that De, the largest manifesto corpus among the five languages, profits from CRR/IRR more than Tr and Hu. We believe this is mainly due to the difference in proportion of high-frequency band – where we see the least improvement – across De and other two languages. Indeed, further analysis shows that in Tr and Hu, the high-frequency band account for 76% and 79% of the data, respectively, while it only makes up 73% of the German data.

8.5 Conclusion

In this chapter, we have demonstrated that the hierarchically structured codebooks developed by political science projects are a source of domain knowledge that can be exploited to lift fine-grained claim classification to a usable level. To do so, we extend the standard classification models introduced in Chapter 6 with lightweight modules that implement this intuition in different ways. We evaluate these methods on two datasets, covering two codebooks, single-label and multi-label classification, and five languages. Our main findings are robust across the different setups: inclusion of hierarchical information always improves classification, and the methods we consider are sufficiently complementary that their benefits combine. We obtain improvements even for fairly large datasets, with diminishing benefits for very large datasets which is plausible, given that performance improves particularly for low-frequency categories.

The fact that the encoding methods are particularly useful for low-frequency categories is highly important with regard to two aspects: First, political discourse unfolds over time, and every prominent issue starts out as infrequent. Second, as discussed in Chapter 7, it is also important for algorithmic fairness, since in the case of rare categories, a small number of prediction errors is sufficient to substantially impact the reliability of downstream analyses. Indeed, multiple causes of low frequency categories exist. As one example, in analyses over time, statement frequencies co-vary naturally with topic prominence, and analyses like the (semi-)automatic extraction of network representations to assess dynamics of political debates (Haunss et al., 2020) may misrepresent the contribution of infrequent categories. As another example, work on the framing of immigration discourse on Twitter (Mendelsohn et al., 2021) has shown that employing issue-specific categories (e.g., "victim:war", "victim: discrimination", "threat:jobs", "threat:public order") reveal ideological and regional patterns which would be missed by the commonly employed generic frames such "economy" or "morality" (Card et al., 2015) – but at the cost of introducing many fine-grained categories which are sparse and attested with widely different frequencies. Our work in this chapter demonstrates that a well designed hierarchical codebook, combined with the right computational devices, can go a long way towards redressing the challenges that arise from this situation.

One aspect of this study that we have not considered yet is the evaluation of the downstream effect of these methods on construction of discourse networks (cf. Chapter 4). In order to evaluate how useful these hierarchy encoding methods can be for automatic construction of discourse networks, we can compare a discourse network whose automatic construction involves a claim classifier with a hierarchy encoding method, with a discourse network which is created using a plain claim classifier. We leave this as a future work.

9 Investigating Adversarial Debiasing Under Correlated Attributes

In Chapter 7, we discussed the well-known fact that statistical models tend to rely on spurious correlations between labels and input features to maximize their performance and we showed that our claim detector model is not an exception: It exploits the spurious correlation between the actor frequency and the task label which makes the model to recognize claims made by infrequent actors with much worse recall.

In this chapter, we continue to look at those spurious correlations build in NLP models. Now, however, we turn our attention to another task and subdomain of CSS. Namely, we focus on a text classification task, a fundamental NLP task, on social media domain because (1) it has become a very active line of research in recent years and (2) it provides information about demographic attributes of authors via their publicly available profiles.

9.1 Introduction

As described in Chapter 2 in detail, NLP applications, including text classification, recently received a significant deal of criticism because of the frequent presence of bias in the predictions. In text classification tasks, a principal source of such biases are demographic attributes of authors, such as gender, age, or race[1]. The reason is that these attributes shape speakers' language use substantially (Hovy, 2015). NLP models are not only able in principle to pick up such cues, as studies on modeling demographic attributes show (Koppel et al., 2004), but they actually have a motivation to do so whenever some demographic attribute is strongly *correlated* with the model's classification target and therefore supports its recognition. As an example, in social psychology, Gross et al. (1997) report that elderly people experience and express their emotions less intensely than younger people. Therefore, in a corpus of emotional expressions across age groups, it is reasonable for a model that predicts emotion intensity to look out for linguistic cues regarding author age, even if these cues are not really related to emotion intensity, such as typical markers of youth language such as "rad", "fam", "FTW", etc.

This focus is arguably problematic, though, since it can give rise to a form of *age bias*, overestimating emotion intensity for documents exhibiting youth language. More generally, the bias-inducing role of demographic attributes is dangerous for studies that use texts from a multitude of authors – often gathered from social media – to draw inferences about the authors (Sobkowicz et al., 2012; Cheng et al.,

[1] A subset of these has specific legal protection in many jurisdictions under the name of *sensitive* or *protected* attributes.

2015). In such studies, demographic biases can lead to erroneous causal attributions.

As we pointed out previously, to counteract the presence of biases in NLP, researchers have devised various debiasing methods and due to its general applicability and high effectiveness, adversarial debiasing has become one of the most widely used methods for bias mitigation. Unfortunately, these advances are not accompanied by an analysis of the prerequisites that need to be satisfied for adversarial training to perform successfully. It has been empirically shown that adversarial training works well for many cases in NLP (cf. Chapter 2); nevertheless, we demonstrate that there are relatively simple setups where it can fail. In this chapter, we analyze what factors contribute to the failure. Concretely, we consider the correlated attributes of document topic (scientific / non-scientific) and author gender on a self-collected multilingual corpus of TED talk transcripts in French, German, Spanish, and Turkish. We choose gender and topic as the target attributes because gender bias in NLP has been an issue of great importance and topic is a highly correlated attribute with gender as shown by previous work (Newman et al., 2008; Hovy, 2015; Schwemmer and Jungkunz, 2019). This setup enables us to observe the interplay between linguistic properties and adversarial debiasing.

Our investigation proceeds in three steps. First, we train independent classifiers for each attribute and evaluate them with regard to overall performance and with regard to the bias they exhibit. In the second step, we apply adversarial debiasing to the predicting of each attribute with respect to the other, and re-evaluate the debiased models. Finally, in the third step we discuss the differences observed in the previous step: (a), both document topic and author gender can be classified reasonably well by independent classifiers, but exhibit considerable bias; (b), author gender bias in topic classifiers can be reduced by adversarial training; however, adversarial debiasing in the opposite direction fails completely; (c) this effect is true for all languages except French. Our interpretation is that the failure of adversarial debiasing is due to the fact that feature space for author gender is *subsumed* the topic feature space for all languages except French, where gender is expressed overtly by morphological cues that can be picked up by the model.

9.2 Dataset

In order to conduct a study on the relationship between topic and author gender in multiple languages, we require a multilingual comparable corpus for which topic and gender information are available. The corpus should be as parallel as possible so that any differences in outcome across languages are not simply due to differences in the evaluation data. Among the available multilingual parallel data sets, arguably the two most prominent ones are WIT^3 and OPUS. WIT^3 (Cettolo et al., 2012) consists of lecture translations automatically crawled from the TED talks in a variety of languages and was used in the evaluation campaigns IWSLT 2013 and 2014. OPUS (Tiedemann, 2012) is a collection of data from several sources which provides sentence alignments as well as linguistic markup (for some languages). Unfortunately, neither corpus provides topic or gender labels.

For this reason, we create a new multilingual parallel dataset with these annotations, based on TED talks (http://ted.com/talks). A TED talk is a presentation at the TED conference or one of its international partner events. TED talks are limited to a maximum length of 18 minutes and may be on any topic. TED talks are rehearsed talks and at least semi-formal, while still definitely belonging to the category of spoken language. In this regard, they are comparable to the widely used Europarl corpus (Koehn, 2005). The talks are divided according to the languages, topics and posted dates. All original talks are presented in English, but volunteers provide (and double check) translations into other languages. Authors are identified by name.

Checking for which languages the TED webpage provided substantial numbers of transcripts (as of February 2020) led us to select German (DE), Spanish (ES), French (FR) and Turkish (TR) as target languages. We crawled all 1518 TED talks for which transcripts in all four target languages were available. We conducted some preprocessing: we cleaned transcripts by removing extra line breaks, extra spaces, and punctuation marks. Inspired by the work in open-domain Question Answering (Yang et al., 2019), we then segmented the transcripts into a sequence of segments. Rather than using paragraphs or sentences as segments directly, we split articles into segments with the length of 60 words by sliding window as Wang et al. (2019) demonstrated that splitting articles into non-overlapping fixed-length segments leads to better results in Question Answering.

Finally, we annotated the transcripts with topic and author gender information. For topic, we grouped transcripts into two classes according to the community-provided tags. The instances that have either Technology or Science tag were labeled as *SciTech* while the rest was labeled as *Other*. This grouping strategy led to a balanced dataset (53% Science, 47% Other). For author gender, we assume a binary gender classification (male/female) to be compatible with existing datasets (Verhoeven et al., 2016; Pardo et al., 2016). This should not be understood as a rejection of non-binary gender. We manually determine the author's gender information on the basis of gender indicating pronouns such as *he, she, his, her* that are used to refer to the authors in their biographies published in the authors' TED Talks profile or on other websites, keeping only clear cases.The majority gender is male (69%). Table 24 describes the final dataset[2].

The corpus has very similar properties across languages. The main exception is the lower number of words in Turkish which is due to the agglutinative nature of Turkish morphology. For instance, the English sentence with four words "I am at your house." is translated into a single word Turkish sentence "Evinizdeyim."

# TED Talks	1518			
Author Gender	1042 (Male) / 476 (Female)			
Talk Topic	704 (SciTech) / 814 (Other)			
	DE	ES	FR	TR
# Tokens/doc	2093	2110	2280	1632
# Sentences/doc	115	110	111	114

Table 24: Statistics of TED multilingual corpora.

[2] The data can be found at `https://github.com/wassa21/adv`

9.3 Experimental Design

Table 25 shows a correlation matrix for the two attributes of topic and author gender in our TED corpus. Indeed, the corpus shows a clear correlation between the two: while male authors are represented about equally in TED for scientific-technological topics and other topics, female authors are underrepresented for scientific-technological topics. As motivated in the Introduction, this situation can lead to the model mistakenly picking up linguistic cues from one attribute to predict the other, leading to systematic biases.

Document Topic		SciTech	Other
Author Gender	Male	524	518
	Female	180	296

Table 25: Topic–gender correlation: Number of documents in TED corpus for each combination

We therefore believe that this corpus can serve as a reasonable case study for correlated document attributes. We proceed in two steps: First, we learn individual neural models for topic and gender classification. We expect, for each attribute, that predictions are biased regarding the other attribute (Experiment 1). Second, we debias these models by adversarial training. We expect the models to focus better on features that are predictive of the individual attributes, and to show less bias (Experiment 2). In our experiments, we follow Li et al. (2018) and Zhao et al. (2020) by measuring the *amount of bias* in the models as the average difference in classification performance between documents aggregated by author gender

(Male vs Female) or aggregated by topic (SciTech vs Other).

9.4 Experiment 1: Simple Classification and Bias Analysis

In our first experiment, we set up neural classification models for the two tasks of topic and gender classification individually and evaluate them for the presence of bias in their predictions.

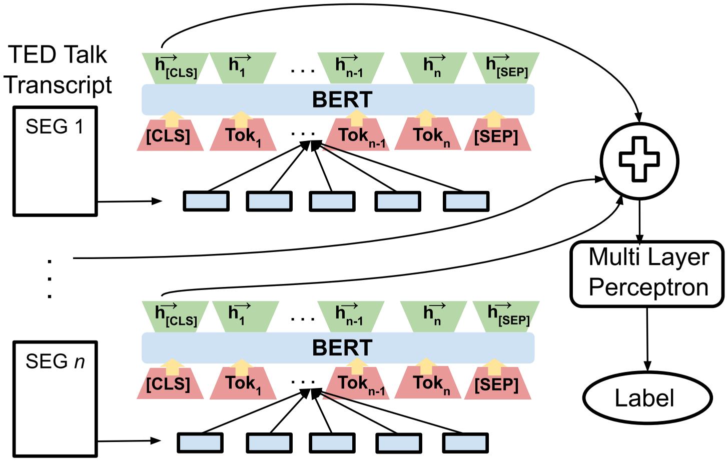


Figure 18: Visualization of classification architecture for topic and author gender

9.4.1 Method

Both Topic and Gender classification tasks can be regarded as a specific type of text classification. Therefore, we use a similar model to BERT based claim classification model used in Chapters 6 and 8, but with some adjustments. BERT and many transformer-based model in general can only encode and generate representations for a fixed length token sequence – e.g., BERT implementations are often limited to 512 tokens per sequence. However, as presented in Table 24, the average token number per TED talk is much larger. To address this limitation, we perform a couple of changes on the BERT based text classifier that we used in the previous chapters. Specifically, we encode the input at the paragraph level (cf. Section 9.2) using the final hidden state corresponding to a special classification token, [CLS], as the representation for the corresponding paragraph. We then obtain the global context vector for the input by summing paragraph representations element-wise. Finally, the global representation of the input is fed through a Multi-Layer Perceptron with a Softmax layer. Note that an alternative, and also easier, approach for using BERT on long input text would be text truncation (e.g., cutting the longer texts off and only using the first 512 Tokens only) but we do not apply this alternative approach as it leads to the loss of global information related to the task. Figure 18 depicts the model architecture we use for both classification tasks.

Training details. We randomly sample 80% of our dataset for training and evaluate on the other 20%. Sampling is made at the talk level and in parallel for all languages to ensure data splits remain the same across languages. Since not all languages created equal in multilingual BERT (Wu and Dredze, 2020), for each language we consider, we use a cased BERT variant that was trained specifically for the target language.[3] We use the Adam optimizer with learning rate of 5e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We set gradient clip threshold to 1.0, batch size to 48 and apply dropout with 0.5 probability on all layers and train the model for 15 epochs. The Multi Layer Perceptron consists of a single hidden layer with 300 hidden units.

9.4.2 Topic Classification

We first set up the model for topic classification. As stated in Section 9.4.1, we approach it as a document-level binary classification task. The input to the model is the full transcript, and the model labels each transcript either as "SciTech" or "Other". Table 26 shows F_1 scores for the topic classification models in four languages as well as the majority baseline.

Language	Overall	By Gender		
		Male	Female	Bias
DE	81.2	80.0	84.0	4.0
ES	80.0	79.0	82.7	3.7
FR	81.5	80.2	83.7	3.5
TR	80.2	78.7	83.0	4.3
Majority BL	37.6	33.6	47.6	14.0

Table 26: F1 scores for topic classification (bottom line: majority baseline, identical for all languages)

[3] DE: `deepset.ai/german-bert`, ES: `github.com/dccuchile/beto`, FR: `camembert-model.fr`, TR: `github.com/dbmdz/berts`

First, we compare the overall performance across languages. A majority baseline performs at 37.6% for all languages, due to the parallel design of the dataset. The neural topic classifiers do substantially better, all showing very similar results around 81% F-Score. Their similar performance may be expected from the parallel nature of the corpus, but it also provides support to our assumption that the texts and transformer models perform comparably across languages. When we break down these results by the other attribute we are interested, namely author gender (Male vs Female), we find that the prediction quality of the topic classifier is an average of 3.6 points lower for male than for female authors. In other words, the topic classifiers show a consistent gender bias across languages, presumably due to the higher-entropy (more equal) topic distribution for male authors, as shown in Table 25. While this bias is lower than the bias of a majority baseline which directly reflects the correlation between the two attributes, it is still substantial and arguably worth mitigating.

9.4.3 Author Gender Classification

After training and analyzing the results of topic classification models, we continue with the opposite task, author gender classification. Similar to previous section, we model the task as a document-level binary classification task using the same BERT based model. The input to the model is the full transcript, and the model tries to predict the author gender.

Table 27 summarizes the results. We see a pattern that differs substantially from topic classification, with much larger cross-lingual

Language	Overall	By Topic		
		SciTech	Other	Bias
DE	70.8	69.0	75.0	6.0
ES	72.4	69.2	75.8	6.6
FR	82.4	82.0	83.0	1.0
TR	70.4	66.0	74.8	8.8
Majority BL	57.0	64.4	50.8	13.6

Table 27: F1 scores for gender classification (bottom line: majority baseline, identical for all languages)

differences in performance. The results are again substantially above the 57% baseline. We obtain the best result for French (82%), and the worst for Turkish (70%), with a difference of 12% F-Score. This indicates that gender classification builds much more on language-specific information than topic classification. Arguably, for a word-piece based neural model like BERT, a primary source of evidence on author gender are linguistically marked expressions in the text where the author refers to themselves. Thus, prediction of the author gender should be easiest if a language has a frequent and unambiguous mechanism for *gender marking* (Corbett, 1991; Zmigrod et al., 2019). Table 28 shows a multilingual example where French marks gender inflectionally, while the other languages do not. This is indicative of the general case: The languages that we consider in our experiment provide gender marking to different degrees. At one extreme, French marks most adjectives and many nouns consistently for gender. In contrast, Spanish marks gender only for a subset of the lexicon, and morphologically inconsistently (Harris, 1991); German marks only (some) nouns, and marking is sometimes optional.

DE	Genau hier wurde ich geboren und verbrachte die ersten sieben Jahre meines Lebens.
ES	Esta es la tierra en la que nací y pasé los primeros siete años de mi vida.
FR	Je suis **née** ici même, et j'y ai passé les sept premières années de ma vie.
TR	Doğduğum yer burası ve hayatımın ilk yedi yılını burada geçirdim.

Table 28: Example of inflectional gender marking in different languages (marking only present in French)

At the other extreme, Turkish does not mark gender at all.

On this basis, we would expect French to perform best, and lower performance for the other three languages – exactly what we find. However, the performances for TR, DE, and ES are surprisingly close to one another, and substantially above the baseline: on the basis of what information in the texts do these classifiers base their predictions? A look at the size of the biases suggests an explanation: The gender classifiers for DE, ES, and TR make substantial use of *topic* cues, which enables them to proceed to some extent due to the correlations between topic and gender, but also lead to biases of 6–9% (highest for Turkish, consistent with the analysis above). In contrast, the French classifier is least biased, indicating that its text contains enough cues for proper gender classification. We illustrate this in Table 29, where we report results on SciTech documents with female authors, that is, the smallest subcategory in our corpus. We find that the gender classifier for FR significantly outperforms the others, which provides additional evidence that the model relies less on the topic cues for gender classification.

	DE	ES	FR	TR
SciTech/Female	75.0	75.8	83.0	74.4

Table 29: F1 scores for gender classification on SciTech talks with a female author.

Summary of Experiment 1: For both topic and author gender classification tasks, we find that the classification performance shows a bias with respect to the other attribute. However, the two tasks differ with respect to the cross-lingual component: Topic classification works about equally well in all languages. In contrast, author gender classification only works properly in the one language that has consistent linguistic marking of gender, while there is evidence that the other languages fall back on topic features also for this task, which directly leads to biased predictions. These observations motivate experiments into how well these models respond to debiasing.

9.5 Experiment 2: Adversarial Debiasing

As previously discussed in Section 7.4.2, adversarial debiasing makes a classifier to ignore some bias attribute P such as gender, race, age etc. while learning to solves another task T (Zhang et al., 2018; Elazar and Goldberg, 2018; McHardy et al., 2019). It seeks to achieve this by constraining representations in a way so that representations do not rely on P in any substantial way. To this end, the model is trained to simultaneously predict the correct label for

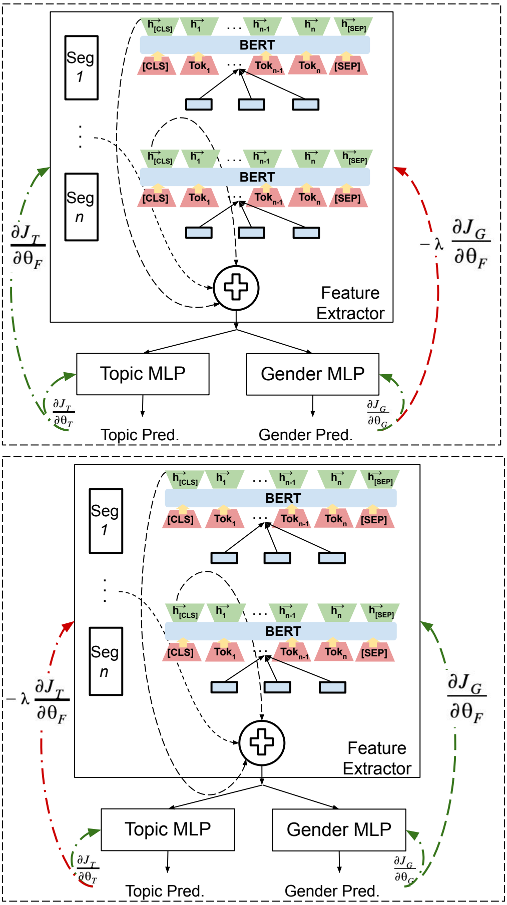

$\vec{h}_{[CLS]}$ $\vec{h}_1$ … $\vec{h}_{n-1}$ $\vec{h}_n$ $\vec{h}_{[SEP]}$
BERT
[CLS] Tok$_1$ … Tok$_{n-1}$ Tok$_n$ [SEP]
Seg 1
Seg n
Feature Extractor
Topic MLP
Gender MLP
Topic Pred.
Gender Pred.
$\frac{\partial J_T}{\partial \theta_F}$
$-\lambda \frac{\partial J_G}{\partial \theta_F}$
$\frac{\partial J_T}{\partial \theta_T}$
$\frac{\partial J_G}{\partial \theta_G}$
$-\lambda \frac{\partial J_T}{\partial \theta_F}$
$\frac{\partial J_G}{\partial \theta_F}$

Figure 19: Debiasing by adversarial training. Top: Adversarial training of topic classifier on author gender, Bottom: Adversarial training of author gender classifier on topic.

task T (“main component”) and to prevent a jointly trained adversary (“adversarial component”) from predicting P. To do so, we use the same training procedure as described in Section 7.4.2: Let J_M and J_A be the loss functions of the main and adversarial components; θ_A, θ_M are the parameters of adversarial and main components; λ be the meta-parameter controlling the intensity of the adversarial training; and η be the learning rate. Then the following equations describe update rules for each component in the model:

$$\theta_M := \theta_M - \eta \frac{\partial J_M}{\partial \theta_M} \tag{9.1}$$

$$\theta_A := \theta_A - \eta \frac{\partial J_A}{\partial \theta_A} \tag{9.2}$$

$$\theta_F := \theta_F - \eta \left(\frac{\partial J_M}{\partial \theta_F} - \lambda \frac{\partial J_A}{\partial \theta_F} \right) \tag{9.3}$$

Note that the adversarial and main components share the same feature extractor (i.e., BERT) whose parameters (θ_F) are therefore updated by the gradients coming through the objective functions of both model parts. Our application of this training method is shown in Figure 19. We first debias the topic classifier by author gender (left-hand box); then we proceed to debias author gender classifier by topic (right-hand box). For example, to de-bias the topic classification, J_M is the topic loss and J_A the author gender loss.

9.5.1 Gender-Debiased Topic Classification

First, we debias topic classification to reduce the gender bias. Figure 20 compares overall topic classification results across a range of values of λ between 0 (no adversarial training) and 1 (equal weight of main and adversarial loss).

We find that, similar to simple topic classification in Experiment 1, the results are essentially identical across languages. Furthermore, the choice of λ hardly matters in this interval: adversarial training does not have a major impact on topic classification. We report detailed results for λ=1 in Table 30. The small differences between the Overall results of the Original and Debiased models show that topic classification overall does not lose much by debiasing for gender. The breakdown by gender shows that gender bias is substantially reduced overall. However, there are noticeable differences among languages.

	Original		Debiased			
	Overall	Bias	Overall	Male	Female	Bias
DE	81.2	4.0	81.2	80.8	81.2	**0.4**
ES	80.0	3.7	80.0	79.2	81.6	**2.4**
FR	81.5	3.5	80.2	79.4	81.4	**2.0**
TR	80.2	4.2	78.4	76.8	80.7	**3.9**

Table 30: Detailed topic classification results for λ=1. Original: Results from Experiment 1 (cf. Table 26). Lower bias for each language bolded.

For Spanish and German, we see no overall loss of performance in topic classification, and a substantial reduction in gender bias. For French and Turkish, in contrast, we see a decrease of about 1.5

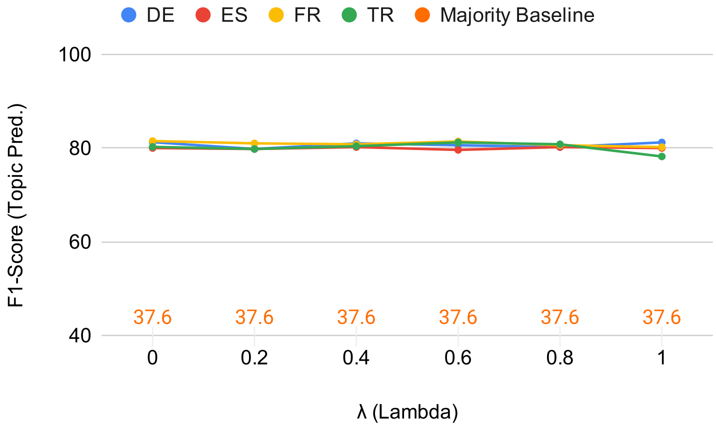


Figure 20: F1 Scores for topic classification with adversarial author gender training using a range of values of λ

points in topic classification. Gender bias is reduced for French but hardly for Turkish. This is a somewhat surprising result, given the typological differences between the two languages. Our explanation is that in French, as discussed above, many words are morphologically marked for gender. Due to the correlation between the two attributes, these can be re-used by the topic classifier, but when they are penalized through adversarial training, we see a mild decrease in topic classification accuracy. In Turkish, as we have argued in Experiment 1, gender classification depends almost entirely on topic features since there is no linguistic marking of referent gender. Consequently, the adversarial training works against itself to an extent, resulting in a mildly worse topic classification but hardly any decrease in gender bias.

9.5.2 Topic-Debiased Gender Classification

After debiasing topic classification to reduce the gender bias, we swap the main and adversarial tasks (cf. the bottom box in Figure 19) to debias author gender classification with regard to topic.

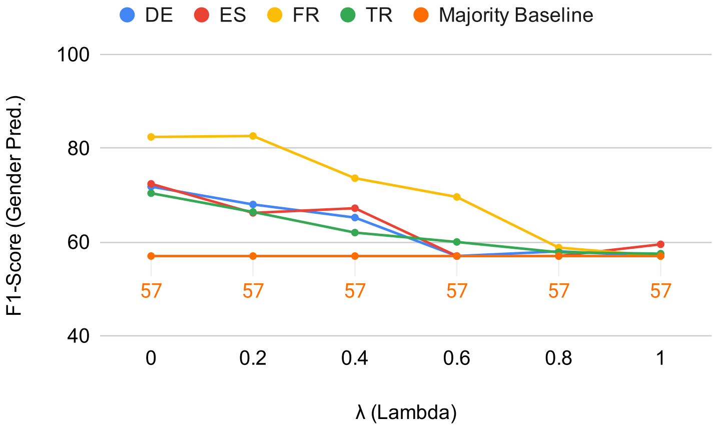


Figure 21: F1 Scores for author gender classification with adversarial topic training (F1 scores). Left: Overall results for different λ values.

Similar to above, Figure 21 compares overall results across a range of values of λ between 0 (no adversarial training) and 1 (equal weight of main and adversarial loss). We find that varying λ has a substantial effect this time. If we set λ to a value close to 1 – a good choice for gender-debiased topic classification, as we have established in the previous subsection – this leads to a breakdown of the gender classification model. Performance for all languages drops to a F-Score of around 57, the level of the majority baseline (cf. Table 27). Apparently, debiasing author gender classification by adversarial training against topic breaks the author gender classifier for all but small values of λ.

	Original		Debiased			
	Overall	Bias	Overall	SciTech	Other	Bias
DE	70.8	**6.0**	68.0	63.8	72.2	8.4
ES	72.4	6.6	66.2	62.8	69.2	**6.4**
FR	82.4	1.0	82.6	82.6	82.6	**0.0**
TR	70.4	8.8	66.4	64.0	68.8	**4.4**

Table 31: Detailed author gender classification results for λ=0.2. Original: results from Experiment 1 (cf. Table 27. Lower bias for each language bolded.

As in the first experiment, we observe differences among languages: French stands out as the language for which the gender classification 'holds out' the longest for high values of λ. Its ultimate failure indicates that even for French, gender marking on its own is not strong enough to support the author gender identification task – or at least our models are not powerful enough to pick up on these cues. The other languages, which, as we have argued in Experiment 1, make substantial use of topic cues for gender classification, fail even earlier.

Table 31 reports detailed results for λ=0.2. In line with our analyses above, debiasing works for French but not for the other languages: We find clear decreases in performance (up to 6.2 points, for Spanish), and inconclusive changes in bias (decrease for Turkish by 4.4 points, increase for German by 1.6 points). Overall, these results indicate that topic-debiasing author gender is a failure both with regard to model performance and reduction of bias.

9.6 Discussion

The results of our two experiments show an intriguing asymmetry between the two tasks of topic and author gender classification when debiased for the respective other attribute. Reducing author gender bias in topic classification with adversarial training proceeds as expected, is relatively robust to the choice of λ in the interval between 0 and 1, and shows a consistent pattern across languages which can be explained by the properties of the languages involved. In contrast, reducing topic bias in author gender classification relies heavily on λ, quickly deteriorating to baseline level for large values of λ, and does not consistently manage to reduce bias in any case. This asymmetry cannot be an artifact of model architecture or data alone, since we use exactly the same model architecture on the same data. Furthermore, we think that this also cannot be just the result of the limitations of the encoder we use (i.e., the fact that we didn't use a document-level encoder such as Transformer-XL (Dai et al., 2019) or Longformer(Beltagy et al., 2020) which can process much longer text sequences by design) because we found that this encoder at least works reasonably well for French (cf. Table 31, Row 3). If this was solely due to the encoder, then we would expect to see that it fail for French too, due to the fact that experiments were conducted on a parallel dataset.

Instead, we believe that these patterns result from an interaction between the representation learning of the model and the information that the model can draw from the data. They can be understood through the *latent feature space* of the final shared layer in our architecture below the two heads (cf. Figure 19), where each class can be characterized by a region of informative features.

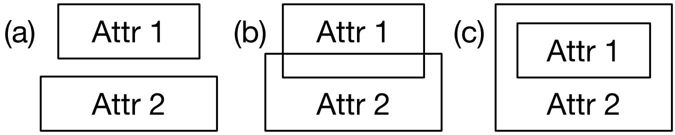


Figure 22: Three cases of latent feature space geometry for two attributes: (a) independent, (b) correlated, (c) subsumed

Figure 22 shows Venn diagram-style depictions of the three possible cases for a pair of attributes. In the left-hand case, (a), there is no overlap between the latent features of the two attributes. That is, the two attributes are independent of one another, and so is learning. However, this is by definition the case without correlations among attributes that we do not consider. In the center case, (b), there is an intersection between the latent features of the two attributes. The classifiers' use of this overlap potentially creates biases, but adversarial training exactly punishes the use of this region of latent feature space. Thus, debiased classifiers can learn either attribute to the extent that the part of the feature space outside the intersection is still sufficiently informative. The right-hand case, (c), is the limit case when one of the two attributes does not have an independent standing, that is, the informative latent features of attribute 1 are completely contained in the informative feature space of attribute 2.

This leads to biases in either classifier just as case (b), but also creates an asymmetry in the effect of adversarial debiasing: Attribute 2 can be debiased by simply 'cutting out' the informative space of attribute 1, but debiasing attribute 1 in the opposite manner results in an empty feature space for attribute 1, and we would expect the classifier to revert to baseline performance.

This set theoretic visualization is a major simplification of the latent feature space in neural models, where the three cases cannot apply categorially — they rather represent different points on a continuum. Nevertheless, the predictions of the subsumption case, (c), match our experimental results well: Assuming that author gender features are included in topic features, we would expect to find successful debiasing of the topic classifier, but breakdown of the debiased author gender classifier. This is exactly the pattern of results that we have observed.

Note that this analysis builds on the behavior of the features of the attributes in the training data, in particular in a representation learning approach like the one we have pursued. In other words, changes of the data – or differences within the data, such as between languages – are expected to influence the outcome. Again, this is what we see: French, due to its consistent morphological marking of gender, is closer to case (b), while the other languages are closer to case (c).

9.7 Conclusion

In this chapter, we conducted a case study to focus on text classification for social media analysis in the context of correlated attributes.

We specifically analyzed the relationship between document topic and author gender using a novel multilingual parallel corpus of TED talk transcripts. We first trained independent classifiers for each attribute and evaluate them with regard to overall performance and with regard to the bias they exhibit. In the second step, we applied adversarial debiasing to the predicting of each attribute with respect to the other, and re-evaluate the debiased models and discussed the differences. Through our experiments, we established that (1) topic classifiers exhibit gender bias and author gender classifiers show topic bias, and (2) adversarial debiasing corrects gender bias in topic classification but breaks down in the opposite direction; and that this effect varies by language.

Another contribution of our work presented in this chapter is to draw attention to the general question of prerequisites for successful adversarial debiasing, which, to our knowledge, has not received much attention. As discussed in Section 9.6, our results indicate that when the target attribute and the bias attribute are too strongly correlated – or, indeed, when the target attribute is subsumed by the bias attribute – adversarial debiasing fails: with a small weight on the bias component, no debiasing takes place; with a large weight, target attribute classification deteriorates to baseline level. Furthermore, we find that the linguistic expression of the attributes matters greatly: the only language for which we achieved satisfactory results was French, due to the consistent morphological marking of gender which can be captured independently of topic (Zmigrod et al., 2019).

10 Bias Identification and Attribution in NLP Models With Regression and Effect Sizes

As we discussed in the Background Chapter in detail and also in Chapters 7-9, there is a quickly growing body of work that has found that NLP systems exhibit unintended biases where biased systems are defined as systems that "systematically and unfairly discriminate against certain individuals or groups of individuals in favor of others" (Friedman and Nissenbaum, 1996). While gender was one of the first bias variables under consideration (Bolukbasi et al., 2016), attention has since branched out to many other bias variables such as race (Davidson et al., 2019) and age (Díaz et al., 2018). At the same time, the techniques often used for the statistical analysis of biases in NLP systems are still relatively basic. Typically, studies test for the presence of a significant difference between two levels of a single bias variable (e.g., gender: male vs. female) without attention to potential confounders, and do not quantify the importance of the bias variable.

In this chapter, we make a general methodological contribution: We argue for the use of regression modeling in order to overcome the limitations of current bias analysis methods. Our multivariate regression based method is a robust and more informative alternative which (a) generalizes to multiple bias variables, (b) can take covariates into account, (c) can be combined with measures of effect size to quantify the size of bias. We apply our method to analyze a range of current models on both regression and classification tasks.

10.1 Introduction

The identification of bias in the output of NLP systems involves the establishment of systematic differences in system performance between two parallel stimuli sets for different levels of a *bias variable* such as gender or race. For example, consider the following question: Does, the gender of an author have a systematic influence on the output of an NLP system (e.g., are texts written by women predicted to be less positive?), or on the quality of the NLP system? (e.g., are text written by women analyzed less reliably?) This question can be answered using statistical analysis techniques of increasing complexity, shown in Table 32. To our knowledge, all existing studies on bias fall into either the first or the second group. Studies in the first group only quantify the *performance differences*. For instance, studies investigating gender bias have generated predictions for sentence pairs which differ only in gendered expressions (e.g. cf. Table 33) and reported the difference between these sets (Zhao et al., 2018a; Stanovsky et al., 2019). Without considering between-system and between-item variance, it is not clear that such

	Performance Difference (Rudinger et al. 2018, Zhao et al. 2018, etc.)	Performance Difference plus Hypothesis Testing (Caliskan et al. 2017, Kiritchenko et al. 2018, etc.)	Regression Modeling with Effect Sizes (Ours)
Assessing statistical significance	-	+	+
Quantifying the impact of multiple variables	-	-	+
Diagnosing system behavior	+	+	+

Table 32: Comparison of different approaches to statistical analysis of bias.

differences are indeed *systematic*, as required by the definition of bias. For this reason, studies from the second group additionally carry out *hypothesis tests*, typically t-tests, to assess the statistical significance of the differences (Kiritchenko and Mohammad, 2018).

Although this procedure is conceptually simple and straightforward, it is problematic for two reasons. First, the pairwise hypothesis tests that are being employed in existing work assume that differences between the two sets of stimuli are due to the selected bias variable. They cannot ensure that the putative effect of bias is not due to a *covariate* that acts as a *confounding variable* (McNamee, 2005). For instance, studies on gender bias often use sets of male and female names as part of their stimulus sets (cf. Table 33). Across genders, these names may differ in the average age of the bearer, or simply in their frequency in texts, both of which may influence the performance of NLP systems (Díaz et al., 2018; Gerz et al., 2018). Similarly, as we showed in Chapter 9, author gender may be correlated with topic which can also have an impact on analyses. Therefore, even when an analysis of performance differences by gender may yield a significant performance difference, it is advisable to rule out that there are competing explanations of the difference in performance in terms of other factors.

Second, bias studies in NLP currently generally test for *statistical significance*, but very few consider *model fit* and *effect sizes* (with the notable exception of Caliskan et al. (2017)). Significance ensures that an identified effect is not a random fluke, but does not quantify how much of the variance in the predictions is due to the bias. Given a sufficiently large dataset, even very small differences that are not practically relevant can reach significance. In contrast, the computation of effect sizes permits users to understand the practical impact of biases (Sullivan and Feinn, 2012).

In this chapter, we propose that these two limitations can be alleviated by adopting *multivariate regression models* such as linear and logistic regression for bias identification. This solution has already become standard in neighboring disciplines like linguistics and psychology. In regression models, bias variables and their covariates form the independent variables, and the predictions of NLP systems for corresponding instances constitute the dependent variable of the equation. As the last column in Table 32 presents, multivariate regression models have many advantages over the other two approaches for bias analysis: (a), they generalize to multiple bias variables; (b), they offer a principled treatment of covariates; (c), they come with measures of effect size that quantify the size of the bias, and (d), they provide a rich diagnosis of system behavior and can be mined easily to extract informative datapoints.

While regression models of various kinds have been used widely as *predictive* models in NLP, we focus on their use as *explanatory* models, where the focus is on building an interpretable model. Models of this type have been applied to analyze the influence of task and data properties on the performance of sequence labeling models (Papay et al., 2020a) or the influence of various textual properties of author responses on the peer review process (Gao et al., 2019). As a final note, we stress that the goal of this procedure is not to "explain away" biases, but rather to propose a more stringent procedure to identify them, in order to strengthen their empirical standing.

In the rest of this chapter, we first introduce our proposed workflow and a set of best practices for designing, computing and interpreting multivariate regression models for this task (Section 10.2). Then, we apply our workflow to two tasks: emotion intensity prediction, a regression task (Section 10.3) and coreference resolution, a classification task (Section 10.4). Finally, we conclude this chapter in Section 10.5.

10.2 Bias Identification With Regression Models: A Workflow

The task of bias identification is to establish that a bias variable – in contrast to other covariates which act as confounders – is primarily responsible for systematic variance in an observed variable, namely the performance of some computer system. This is, of course, a very general task that arises in many empirical fields. A prominent family of techniques to address this task is *matching* (Rubin, 1973), which aims at generating two datasets that differ in the bias variable, but are as close as possible in their distribution over the covariates, so that any difference between the two datasets can be attributed to the bias variable. Matching is widely used in social sciences, economy, and medicine and many specific methods exist; see Stuart (2010) for an overview.[1]

Importantly, matching takes place *a priori*, before the experiment is carried out. This poses two challenges for applications in natural language processing: (a), dataset creation is dependent on the selection of covariates, so that it is not possible to assess the impact

[1] Note that the term *bias* is used differently in the matching literature, namely as the effect of confounders on the observed variable.

of new covariates on existing datasets without loss of comparability; (b), matching samples from the set of all datapoints, creating controlled rather than natural datasets, which may conflict with the desideratum of estimating model performance in broad-coverage scenarios.

The alternative is to carry out a *post-hoc* analysis that assesses the effects of the various covariates. The intuition is to start from a simple pairwise comparison of two levels of a bias variable (cf. the first and second column in Table 32) and add covariates to see whether the effect of the bias variable remains unaffected. This procedure has become standard in the last decade in neighboring fields like linguistics and psychology which have moved from significance tests (Student's t-test, analysis of variance) to the family of *multivariate regression models* (Bresnan et al., 2007; Baayen, 2008; Jaeger, 2008; Snijders and Bosker, 2012). Regression models estimate the relationships between the dependent (previously called observed) variable – in this case, system performance – and one or more independent variables – in this case, the putative bias variable and its covariates, each of which is assigned a direction and a significance. Since dataset creation is dependent from covariate analysis, regression models can be used to test new candidates for confounders on existing datasets.

At this point, it can be whether the fundamentally linear regression models are the right tool for the job, in particular given the broad success of non-linear deep learning models in NLP over the last years. We believe that it makes sense to distinguish carefully between the task of *output prediction* (given language input, predict language output) on which non-linear models indeed excel and the task of *performance prediction* (given [meta data for an] input and a model, predict how well the model does on the input). The latter is a considerably simpler problem which permits the use of linear models, as evidenced by a number of successful studies taking this approach Beinborn et al. (2014); Papay et al. (2020b); Caucheteux and King (2022).

This section provides a practical workflow to set up a regression model for bias analysis, shown in Figure 23. Our starting point is the presence of a dataset with system predictions. Step 1 is the selection of an appropriate regression model. In Step 2, we choose a set of predictors with the potential to systematically influence the predictions of the systems, (i.e., the putative bias variable and plausible confounders) and carry out a regression analysis. Next, Step 3, model validation, ensures that the regression model is well specified and interpretable. Finally, Step 4 utilizes effect size analysis methods to explore how much of the system predictions can be attributed to the influence of the predictors.

Running example. We will illustrate the steps of the workflow on an actual (non-NLP) example, namely the effect of smoking on mortality, a topic of long-running interest in public health that has been analyzed extensively with regression models. The most basic finding is that smoking, overall, causes a strong increase in mortality (Doll et al., 2004). Why it is still reasonable to carry out a regression analysis in this case is that other lifestyle choices (alcohol consumption, diet, etc.) also presumably influence mortality, but exhibit correlations (Padrão et al., 2007). These are sometimes surprising – e.g., Tjønneland et al. (1999) found a correlation between wine and healthy diet. At the same time, approaches like matching are not

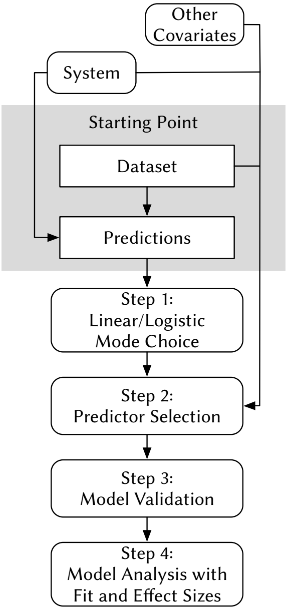

Other Covariates
System
Starting Point
Dataset
Predictions
Step 1: Linear/Logistic Mode Choice
Step 2: Predictor Selection
Step 3: Model Validation
Step 4: Model Analysis with Fit and Effect Sizes

Figure 23: Workflow for regression-based bias analysis

applicable since the lifestyle properties of the participants cannot be influenced retroactively.

10.2.1 Step 1: Choice of Regression Model

The most common two forms of regression analysis are linear regression and logistic regression. When used to analyze the output of computational models, linear regression is appropriate to analyze the output of regression tasks, and logistic regression for the output of classification tasks.

Linear regression predicts the outcome of a continuous random variable y as a linear combination of weighted predictors x_i:

$$y \sim \alpha_1 x_1 + \cdots + \alpha_n x_n \tag{10.1}$$

where the coefficients α_i can be interpreted as the change in y resulting from a change in predictor x_i, keeping the other predictors constant.[2]

In contrast to linear regression, logistic regression does not model the outcome of the binary random variable y directly. Instead, it models the probability $P(y = 1)$, assuming that $P(y = 1)$ stands in a linear relationship to the logistically transformed linear combination of weighted predictors:

$$P(y = 1) \sim \sigma(\alpha_1 x_1 + \cdots + \alpha_n x_n) \tag{10.2}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function. Here, the coefficients α can be interpreted as the change in the logit for a unit change in the predictor.

Both types of regression support continuous, binary, and categorical predictors; the latter type is generally represented as a set of binary indicator predictors. As indicated above, these models assume that the predictors have an additive effect on the dependent variable (in the linear case) or its logit (in the logarithmic case).

[2] If the dependent variable is not (approximately) normally distributed, other types such as Poisson or negative binomial regression may be more appropriate.

Running example. In our mortality example, the outcome of the regression model is (some variant of) a death rate. Depending on the exact choice of measure, it might be appropriate to choose a linear regression model, when the death rates are approximately normally distributed (Gardner, 1973); or it might be appropriate to choose a logistic regression model, when the death rates can be interpreted as probabilities (Zhu et al., 2015b).

10.2.2 Step 2: Selection of Predictors

Maybe the most central step in the use of a regression model for bias analysis is the selection of the set of predictors for the regression model – that is, the putative bias variable and a set of plausible confounders to assess the respective roles of these variables is explaining the variance of the dependent variable. This task is the responsibility of the user and typically involves domain knowledge. Typically, a user carrying out a bias identification analysis will have one or a small number of bias variables in mind, but need to select plausible confounders.

The five primary sources of bias variables given by Hovy and Prabhumoye (2021) can also serve as sources of confounders. The most straightforward of these are *data* and *input representations*, that is, properties of the text underlying the model, many of which are known to impact model performance. For example, as we discussed in Chapter 7 and 8, low-frequency words and classes are modeled less reliably. Furthermore, Poliak et al. (2018) showed that longer stretches of text are harder to analyze. Similarly, differences among *annotators* (age, social and cultural background, task familiary) can impact model performance through labeling decisions (Sap et al., 2019), and obviously design decisions of the *system*, such as the choice of neural network architecture, contribute as well (Basta et al., 2019). Hovy and Prabhumoye's fifth category of *research design* is least relevant for our purposes, since it is concerned with systematic gaps in the field as such rather than analysis of individual studies.

Thus, for many problems, there will be a range of theoretically motivated covariates. The actual analysis will proceed in an interlocking fashion between exploratory data analysis based on domain knowledge – to identify interesting candidates for covariates – and regression modeling – to obtain statistically sound assessments of these covariates. In practical terms, the limiting factor is often that covariates need to be available as annotation on the dataset under consideration. While this is often relatively simple for the domains of input representation and systems, and doable for the domain of data, only recently has natural language processing started to record and analyze annotator properties (Sap et al., 2019), and there is an inherent tension between insights into annotation biases and annotator privacy.

Running example. In our lifestyle example, the covariates ideally include as many lifestyle factors as possible (such as alcohol consumption, diet, exercise, occupational hazards) as well as environmental factors (housing, climate) and personal factors such as family history of certain diseases. In practice, again, only a limited range of such factors is likely to be available.

10.2.3 Step 3: Model Validation

While regression models technically support arbitrary covariates, strong correlations among predictors, so-called *multicollinearity*, can distort the estimation of coefficients to the point that predictors are suggested to be significant when they are not, and vice versa (McNamee, 2005). Therefore, models should be checked for the presence of multicollinearity. Following the previous work, we use we use variance inflation factor (VIF), one of the most widely used methods on multicollinearity analysis (Yoo et al., 2014). VIF measures how much the variance of a predictor's coefficient is inflated due to correlations with other predictors. The VIF is computed for each independent variable V_i as

$$\mathrm{VIF_i} = 1/\left(1 - \mathrm{R_i^2}\right) \tag{10.3}$$

where R_i^2 is the correlation coefficient obtained when predicting α_i from all other predictors. Thus, the more collinearity is present, the higher VIF_i. VIF values of 4 or greater indicate severe multicollearity, and values above 2.5 call for further investigation (Salmerón et al., 2018). In these case, a number of strategies are available, including dropping covariates, dimensionality reduction, and regularization methods.

Another possible component of model validation is *predictor (feature) selection* based on an analysis of feature contributions. In many NLP tasks, irrelevant or unimportant features are removed for reasons of efficiency or to avoid overfitting (Li et al., 2009). In fields like psychology, where models serve explanatory purposes, predictor selection is discussed more controversially (Barr et al., 2013; Bates et al., 2018). In bias analysis, the goal is to test whether the effect of the putative bias variable stands up to the addition of covariates – the more covariates added to the model while retaining a significant contribution of the bias variable, the stronger the evidence for a specific role of the bias variable. For this reason, we believe that regression based bias analysis should be carried out on a comprehensive set of predictors, without feature selection (Barr et al., 2013).

Running example. In our lifestyle example, is it arguably important to check for multicollinearity, since the various covariates may be predictive of one another. For example, cramped housing conditions and occupational hazards are strongly linked through the shared cause of poverty (Hajat et al., 2015).

10.2.4 Step 4: Computing Model Fit and Effect Sizes

The coefficients α computed by regression models (cf. Step 1) are accompanied by indications of the confidence level at which they are different from zero (i.e., whether the predictor has a significant effect). Furthermore, the global quality of regression models can be assessed by a number of statistics. Among them, we use *goodness of fit* which describes the proportion of the variance in the data that is explained by the independent variables of a regression model. The goodness of fit of a linear regression model is measured by R^2:

$$R^2 = \frac{\sum_{i=1}^{n} \left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2} \tag{10.4}$$

where $\hat{y}_i$ is the model's prediction for data point i and $\bar{y}$ is the mean of the observations.

In logistic regression, there is no exact equivalent of R^2. Among the various proposed pseudo R^2 measures, Aldrich-Nelson pseudo-R^2 with Veall-Zimmermann correction (R^2_{VZ}) most closely approximates the R^2 in linear regression (Smith and McKenna, 2013):

$$R^2_{\mathrm{VZ}} = \frac{2[\mathrm{LL(Null)} - \mathrm{LL(Full)}]}{2[\mathrm{LL(Null)} - \mathrm{LL(Full)}] + N} \frac{2\mathrm{LL(Null)} - N}{2\mathrm{LL(Full)}} \tag{10.5}$$

where LL(Full) and LL(Null) are the log-likelihood values for the model with all predictors and for the empty model (without predictors), respectively.

Goodness of fit measures the overall ability of the model to explain the dependent variable. *Relative importance*, on the other hand, refers to the contribution of individual predictors (Achen, 1982). While assessment of relative importance in linear models with uncorrelated independent variables is simple (the impact of each predictor is its R^2 in univariate regression), in real-world datasets variables are generally correlated, as a result of which their impacts are not additive (Grömping, 2006). Lindeman-Merenda-Gold (LMG) scores (Lindeman et al., 1980) and Dominance Analysis (Budescu, 1993) are two popular techniques to figure out the individual contributions to the R^2 of the model of the predictors in linear and logistic regression, respectively.

The LMG method adds predictors to the regression model sequentially, and considers the resulting increase in R^2 as its contribution. Since this method depends on the possible orders in which predictors are added, the LMG score of a predictor x_k when added to a model with a set of predictors P is defined as the average of the increase in R^2 when adding x_k to all subsets of P (Grömping, 2006):

$$\operatorname{seq} R^2(M|S) = R^2(M \cup S) - R^2(S) \tag{10.6}$$

$$\operatorname{LMG}(x_k) = \frac{1}{n} \sum_{j=0}^{p-1} \sum_{\substack{S \subseteq P \\ n(S)=j}} \frac{\operatorname{seq} R^2(\{x_k\}|S)}{\binom{p-1}{j}} \tag{10.7}$$

where $R^2(S)$ corresponds to the goodness of fit measure of a model with regressors in set S (cf. Eq 1) and $\operatorname{seq} R^2(M|S)$ refers to the increase in R^2 when the regressors from M are added to the model based on the regressors S.

For logistic regression, there is again no direct counterpart. We propose Dominance Analysis (Budescu, 1993) as a measure of the relative importance of each predictor. Dominance analysis considers one predictor (x_i) to completely dominate another (x_j) if x_i's additional contribution to every possible model which does not include these two predictors is greater than contribution of x_j. In cases where complete dominance cannot be established, general dominance can also be used. One predictor generally dominates another if its average conditional contribution over all model sizes is greater than that of the other predictors (Azen and Traxel, 2009).

We propose the following interpretations for the regression scores outlined above: (a) At the system level, R^2 and pseudo-R^2 are indicators of the amount of variance in the system predictions that can be explained by the predictors and measure the *systematic bias* of a system. (b) At the predictor level, the selection of a predictor indicates the *presence of a specific bias*, and its effect size measures its *practical impact*; (c) the sign of a coefficient indicates the *direction* of a bias.

Regarding (b), an important difference between the application of significance testing in bias analysis and the usual use in NLP to compare competing models is that in our case, null results are arguably informative: they indicate the *absence* of a particular bias, according to the standards of significance. Naturally, the usual disclaimers regarding null results apply: care should be taken to ensure that they are not the result of faults in the experimental setup.

Running example. In our lifestyle example, the outcome of this step is a better understanding of individual risk factors, such as smoking, as opposed to the cluster of ’smoking and associated factors’ that is obtained from a simple smoker-vs.-non-smoker analysis. Such an understanding is crucial to better assess the risk of individual patients based on their individual risk profile which might include compounding factors (high blood pressure, alcohol consumption) or mitigating factors (exercise, healthy diet). Again, note that the goal of this analysis is not to detract from the hazardous nature of smoking, but to better estimate of the effects of the relevant predictors on the outcome, namely mortality.

10.3 Experiment 1: Emotion Intensity Prediction

We now employ regression models to reanalyze model predictions on two experiments on standard datasets from the bias literature using the workflow defined in Section 10.2.

Our first experiment is concerned with emotion intensity prediction. This task aims at combining discrete emotion classes with different levels of activation. Given a tweet and an emotion, the task requires to determine a score between 0 and 1 which is the intensity expressed regarding an emotion. Emotion intensity prediction was among the first NLP tasks to receive attention from a bias angle, when Kiritchenko and Mohammad (2018) found that among more than 200 emotion intensity prediction systems, almost all were biased with regard to gender or race. In the remainder of this chapter, we will use *"system"* to refer to models performing the task at hand, and *"model"* to refer to the regression models we use for analyzing the systems' performance.

10.3.1 Dataset and Previous Analysis

We use EEC, the same dataset used for the large-scale bias analysis of sentiment analysis mentioned above (Kiritchenko and Mohammad, 2018). EEC is a bias analysis benchmark created to evaluate fairness in sentiment analysis systems. It consists of 11 sentence templates instantiated into 8,640 English sentences for four emotions: Anger, joy, fear and sadness. Instantiated templates differ only in

Template
1. [PER] feels [EMO].
2. The situation makes [person] feel [EMO].
3. I made [person] feel [EMO].
4. [PER] made me feel [EMO].
5. [PER] found herself in a [EMO] situation.
6. [PER] told us about the recent [EMO] events.
7. The conversation with [person] was [EMO].
8. I saw [person] in the market.
9. I talked to [person] yesterday.
10. [PER] goes to school in our neighborhood.
11. [PER] has two children.

(a) [EMO]: an emotion adjective

African American		**European American**	
Female	**Male**	**Female**	**Male**
Ebony	Alonzo	Amanda	Adam
Jasmine	Alphonse	Betsy	Alan
Lakisha	Darnell	Courtney	Andrew
Latisha	Jamel	Ellen	Frank
Latoya	Jerome	Heather	Harry
Nichelle	Lamar	Katie	Jack
Shaniqua	Leroy	Kristin	Josh
Shereen	Malik	Melanie	Justin
Tanisha	Terrence	Nancy	Roger
Tia	Torrance	Stephanie	Ryan

Table 33: (a): Sentence templates in EEC dataset, (b): female and male first names associated with being African American and European American.

the name[3]. The dataset compares (a) male vs. female first names,

[3] The EEC templates can also be instantiated using gendered noun phrases, but since these are unspecific with regard to the race variable, we focus on

and (b) European American vs. African American first names, using ten names of each category. Table 33 shows examples of such template sentences along with names that tend to belong to African American or European American demographic groups. Kiritchenko and Mohammad (2018) used the EEC as a secondary test set for systems submitted to the SemEval 2018 Task 1 (Mohammad et al., 2018). For each system, they compared the average emotion intensities across different demographic groups using t-tests. They found that almost all systems consistently scored sentences of one gender and race higher than another, but bias directions were not consistent: e.g., some systems assigned higher emotion intensities to African Americans and lower ones to European Americans, while others show the opposite behavior. This apparently random behavior of the systems has no clear explanation and arguably raises concerns about a possible role of randomness in the analysis.

	train	dev	test	task
EI-reg	1701	388	1002	EIP
EEC	-	-	2100	EIP
GAP	-	2000	2000	CR

Table 34: Number of examples in the datasets used in our emotion intensity prediction (EIP) and coreference resolution (CR) experiments.

the version with proper nouns. This corresponds to the race analysis of the original study.

10.3.2 Systems

Since the predictions of the systems that participated in SemEval 2018 Task 1 are not publicly available[4], we instead implement and analyze five systems ourselves. Four systems represent the main architectures submitted to the shared task (Kiritchenko and Mohammad, 2018): A SVM unigram baseline and three neural systems based on word2vec word embeddings. To extend the model set to the current state of the art (2021), we include a transformer-based architecture as fifth system.

Support Vector Machine (SVM) We implement the unigram-based SVM used as baseline system in Mohammad et al. (2018).

Convolutional Neural Network (CNN) Based on Aono and Himeno (2018), this system predicts an intensity score by first performing convolutions of different sizes on input word embeddings, followed by max-pooling and a shallow Multi-Layer Perceptron (MLP).

Recurrent Neural Network (RNN) Our RNN is comparable to Wang and Zhou (2018). A two-layer BiLSTM traverses the input. The final hidden states in both directions from the final layer are concatenated and fed to a fully connected layer.

Attention Network (ATTN) This system is based on a CNN-LSTM architecture with attention similar to Wu et al. (2018). The input is fed to a single-layer BiLSTM. Next, an attention mechanism

[4] Personal communication with the authors of shared task.

weights the hidden states, which are then passed through a CNN. The outputs of the CNN feature maps are concatenated and passed through a pooling layer and two fully connected layers.

Transformer-Based Neural Network (BERT) Our last model is based on BERT. Similar to the BERT based claim classification model that we used in Chapter 6, it consists of two modules: a pretrained BERT and a dense output layer. Input text is tokenized using WordPiece; the [CLS] special token is added to the beginning of the token sequence. The resulting input representations are then fed into BERT to generate latent context-aware representations. We treat the final hidden state of BERT model corresponding to [CLS] token as the contextualized representation of the input sequence and feed it to the output layer with a sigmoid activation function.

We train all the systems on the Anger partition of the EI-reg corpus (Mohammad and Bravo-Marquez, 2017) using the official training, development and test splits and evaluate them on EEC. EI-reg was created by querying tweets in three languages (English, Arabic, Spanish) and for four emotions (Anger, Fear, Joy, Sadness). For each emotion, authors select 50 to 100 terms that are associated with that emotion at different intensity levels (eg. *angry, annoyed, irritated* for Anger) and identify tweets that contain at least one term from the list. The tweets are then annotated via crowdsourcing. Table 34 shows data statistics for EI-reg and EEC datasets.

Example	Properties				Intensity
	Gender	Race	Age	Freq	
Frank feels angry	Male	EA.	Old	0.05	0.55
Alonzo feels angry	Male	AA.	Old	0.24	0.48
Justin feels angry	Male	EA.	Young	0.27	0.46
Lamar feels angry	Male	AA.	Young	0.42	0.49
Jasmine feels angry	Female	AA.	Young	0.47	0.47
Ellen feels angry	Female	EA.	Old	0.19	0.50

Table 35: Example sentences for the first template from Table 33 with their properties (EA.: European American, AA.: African American). Intensity predicted by the the RNN system.

10.3.3 Setup of the Regression Model

Bias Variable. In the EEC setup, the input sentences differ only in the person names that are filled in. We use the same two bias variables considered by the original study, namely Race and Gender.

Covariates. Due to the minimalist nature of the templates, coupled with the fact that the only part of the templates that is manipulated across conditions is the names, there is a limited range of linguistic properties that can systematically covary with bias. We consider two that we consider promising candidates. The first one is the (perceived) Age of a name is computed as the mean age for each name from US Social Security data.[5] We discretize age, using 40 as the young/old boundary, following the assumption that "older"

[5] We use data from `https://bit.ly/34cgjki` and the methodology from `https://bit.ly/30f8lps`.

names occur in different contexts than "younger" names. The second covariate is the linguistic frequency of the name in the training data, since as previously shown in Chapter 7, low-frequency names can be a source of low performance in NLP models. Since no explicit frequencies are available for the Google News skipgram vectors (Mikolov et al., 2013a), we approximate frequency by vector length, which correlates highly with frequency (Roller and Erk, 2016). This is different from the "real world" frequency of the name, which arguably is less likely to reflect in the behavior of an NLP model. Table 35 shows examples from the EEC with their properties.[6]

Model Shape We analyze the intensities predicted by our systems as in the original study, performing linear regression analysis at the level of each template with the following model:

$$\text{Intensity} \sim \text{Race} + \text{Gender} + \text{Age} + \text{Freq} \tag{10.8}$$

For Race, 1 means African American and 0 European American. For Gender, 1 means male and 0 female. For Age, 1 means young and 0 old. Recall that on this task, there is no right or wrong answers. Instead, the focus of interest is whether the systems assign different intensities to a template dependent on the properties of the instantiating name. If they do not, none of the predictors will show a significant effect; if they do, significant effects will emerge.

[6] We also performed experiments using a non-discretized version of age and including real-world frequency. We observed a substantially similar outcome (same levels of significance, coefficient signs for predictors, and almost the same overall R^2 values).

Model Validation. Table 36 shows the variance inflation factors for the variables. As only a single VIF value is larger than 2.5, and only marginally so, we conclude that multicollinearity is not a problem.

	Race	Gender	Frequency	Age
VIF	2.03	1.42	2.68	1.29

Table 36: VIF scores for the full set of variables.

10.3.4 Results

Table 37 shows the main results. We omit intercepts in the table. The columns correspond to systems, and the rows describe the effects of bias variables for each system. For each predictor, we show a coefficient, a confidence level,[7] and an LMG effect size score.

Overall results As discussed in Section 10.2, we treat R^2 as a measure of systematic bias in a system. Inspection of the R^2 scores indicates that there is a certain amount of systematic bias in all systems, but that the three static-embedding neural systems do a very good job (R^2 between 0.17 and 0.19) compared to the SVM (R^2=0.60). BERT, the only neural system using contextualized embeddings, does an even better job and contains the least amount of systematic bias (R^2=0.14).

Comparison among systems None of the neural systems exhibits a significant gender bias, as the LMG scores show. Unlike Gender,

[7] We use * for α=0.05, ** for α=0.01, and *** for α=0.001.

		CNN	RNN	ATTN	BERT	SVM
Race	Coef.	-0.010^{*}	-0.010^{*}	-0.002	-0.008	0.001
	Abs. LMG	0.080	0.082	0.010	0.068	0.018
	Per. LMG	0.42	0.47	0.06	0.48	0.03
Gender	Coef.	0.006	0.002	0.001	-0.001	-0.003^{***}
	Abs. LMG	0.037	0.003	0.020	0.025	0.523
	Per. LMG	0.20	0.02	0.12	0.18	0.86
Age	Coef.	0.005	0.001	0.001^{*}	-0.003	0.001
	Abs. LMG	0.049	0.060	0.070	0.027	0.014
	Per. LMG	0.26	0.34	0.40	0.19	0.02
Frequency	Coef.	0.016	0.019	0.015	0.010	-0.001
	Abs. LMG	0.023	0.029	0.073	0.021	0.048
	Per. LMG	0.12	0.17	0.42	0.15	0.08
Model fitness (R^2)		0.19	0.17	0.17	0.14	0.60

Table 37: Regression-based bias analysis on EEC (Abs:Absolute, Per. Percentage)

the Race variable is responsible for the significant portion of the amount of variance in the system predictions. The CNN and the RNN systems both show a significant race bias which accounts of about 42–47% (LMG score: ∼ 0.08) of the variance in the intensity predictions. Note that Age, even though it misses significance, also accounts for 25–35% of the variation in intensity in the CNN and RNN. Interestingly, the ATTN architecture shows a different picture: there is a considerable amount of Age bias (40% of variance), but a much smaller race bias; instead, this system shows a frequency bias, which accounts for another 40% of the variance. In the BERT system, none of the bias variable achieve significance. In terms of relative contribution of individual predictors, BERT is more similar to CNN and RNN than to ATTN: Race is still making the largest contribution to the overall bias of the system, with 48%. The SVM differs strikingly: there are hardly any Race and Age biases, but an extremely strong effect of gender (86% of variance). Since this system does not use embeddings, the most likely source of this bias is the training corpus (EI-Reg), as also pointed out by the authors of the original study (Kiritchenko and Mohammad, 2018).

Interpretation While we can confirm the overall race bias found by Kiritchenko and Mohammad (2018), our picture differs substantially: (a) the direction of the bias is consistent among systems: all neural systems predict lower intensity scores for African Americans; (b) we do not observe a significant gender bias among neural systems; (c) we achieve a richer understanding of the systems' predictions, by quantifying the role of these factors, and by adding age and frequency into the picture.

Inspection of Examples Following up on (c), Table 35 presents three pairs of examples from the EEC dataset with their associated intensity values, as predicted by the RNN system. We have selected these instances to highlight the usefulness of the regression model to identify interesting instances. They show that the effect of Race variable (African Americans are assigned lower intensities) can be nullified by age (third example) and frequency (first and second examples). Such considerations remain hidden in an analysis that simply compares means between different groups of predictions.

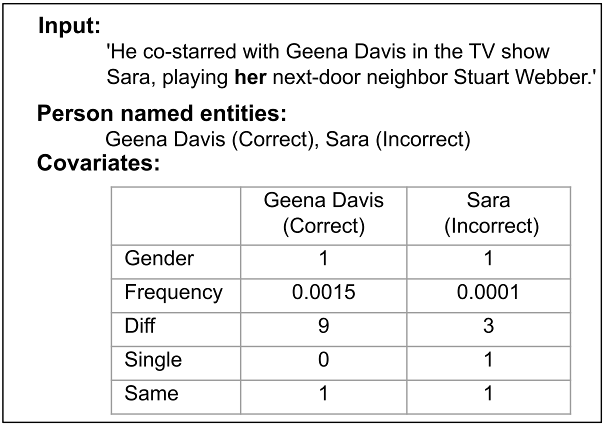
10.4 Experiment 2: Coreference Resolution

Our second experiment analyzes several coreference resolvers in order to show how the logistic regression version of our approach can perform bias analysis on classification models. We choose coreference resolution as our task because of its established status in bias analysis; previous work has established that bias, in particular gender bias, is present in numerous coreference systems (Webster et al., 2018; Rudinger et al., 2018; Zhao et al., 2018a). At the same time, coreference resolution, as a discourse level task, is faced with more complex data than more local (i.e.,sentence-level) tasks, with a correspondingly larger set of potential confounders. We re-analyze a well-known coreference resolution dataset to verify the presence of gender bias in a manner that is robust against possible covariates.

10.4.1 Dataset and Previous Analysis

We use GAP (Webster et al., 2018), a human-labeled corpus of ambiguous pronoun-name pairs from English Wikipedia snippets. Each instance in the corpus contains two person named entities of the same gender and an ambiguous pronoun that may refer to either, or neither. System clusters were scored against GAP examples according to whether the cluster containing the target pronoun also contained the correct name (True Positive) or the incorrect name (False Positive). Figure 24 shows an example from the GAP development set (more statistics in Table 34).

In line with previous work (Webster et al., 2018), we use the de-



Input:

'He co-starred with Geena Davis in the TV show Sara, playing **her** next-door neighbor Stuart Webber.'

Person named entities:

Geena Davis (Correct), Sara (Incorrect)

Covariates:

	Geena Davis (Correct)	Sara (Incorrect)
Gender	1	1
Frequency	0.0015	0.0001
Diff	9	3
Single	0	1
Same	1	1

Figure 24: Example from the GAP dataset.

velopment set of GAP to carry out our analyses. Below, we report overall system performance on the complete development set, in line with previous work. However, we exclude ≈200 instances from the development set, for which the pronoun does not refer to either of the two candidate named entities, from the regression analysis, since this makes it impossible to compute some of our covariates (cf. Section 10.4.3).

10.4.2 Systems

We experiment with six diverse coreference resolvers and analyze their predictions with our approach. As trained versions of all systems were publicly available, we did not need to train any systems ourselves.

All systems except the BERT-based one were trained on the English portion of the 2012 CoNLL Shared Task dataset (Pradhan et al., 2012). It contains 2802 training, 343 development documents, and 348 test documents. BERT$_{large}$ (Joshi et al., 2020) was pre-trained on BooksCorpus (Zhu et al., 2015a) and English Wikipedia using cased *Wordpieces* tokens (Schuster and Nakajima, 2012) and fine-tuned on the 2012 CoNLL ST dataset.

Lee et al. (2013) This system is a collection of deterministic coreference resolution modules that incorporate lexical, syntactic, semantic, and discourse information, incorporating global document-level information. The system won the CoNLL 2011 shared task.

Clark and Manning (2015) This system uses a feature-rich machine learning approach. It performs entity clustering using the scores produced by two logistic classifier-based mention pair classifiers features. Both mention pair classifiers use a variety of common features such as syntactic, semantic and lexical features for mention pair classification.

Wiseman et al. (2016) This was the first neural coreference resolution system which showed that the task could benefit from modeling global features about entity clusters. It uses a neural mention ranker which is augmented by entity-level information produced by a RNN running over the cluster of candidate antecedents.

Lee et al. (2017) This was the first neural end-to-end coreference resolution system that works without a syntactic parser or hand engineered mention detector. It uses a combination of Glove and character level embeddings learnt by a CNN to represent the words of annotated documents. Next, the vectorized sentences of the document are fed into a BiLSTM to encode sentences and obtain span representations. The system also uses an attention mechanism to identify the head words in the span representations. Finally, the scoring functions are implemented via two feed-forward layers.

Lee et al. (2018) This system is an extension of Lee et al. (2017), which improves on two aspects. First, it uses gated attention mechanism which allows refinements in span representations; second, the system applies antecedent pruning which alleviates the complexity of running on long documents. It formed the state of the art for two years.

Joshi et al. (2020) SpanBERT is a variant of the BERT transformer (Devlin et al., 2019) designed to better represent spans of text. It works by (1) masking contiguous random spans, rather than random tokens, and (2) introducing a new objective function called span-boundary objective (SBO) which forces the model to learn to predict the entire masked span from the observed tokens at its boundary. BERT$_{large}$ trained with the SpanBERT method improves the state of the art on many tasks including coreference resolution.

10.4.3 Setup of the Regression Model

Bias Variable. As in the original study, we use *Gender* as designated bias variable.

Covariates. In contrast to the first experiment, we do not use Age and Race, since the GAP dataset contains numerous named entities that are either not generally known or fictional (such as "the Hulk"). Therefore, these variables are either inapplicable or unknown to the typical annotator. Instead, we use discourse-related properties of the antecedents as covariates, since in the task of coreference resolution the structural properties of the discourse arguably play a role in the difficulty of the task:

- *Diff* is the number of tokens between the named entity and target pronoun, normalized by the maximal distance in the corpus;
- *Single* states whether the named entity is a single word or a Multi-word Expression (MWE);
- *Same* indicates whether the pronoun and named entity are in the same sentence;
- *Freq* defines the log-transformed corpus frequency of the entity, computed on the English Wikipedia (*en-wikipedia*) released on 20th March 2019, normalized by the maximal frequency in the corpus. The frequencies for MWEs are calculated based on the syntactic head of the expression.

Since the correct and the incorrect antecedent can differ regarding these properties, each property exists twice. We use the prefix C_for the correct and I_for the incorrect one. For gender, both antecedents have the same gender by design. The bottom part of Figure 24 shows how these covariates are initialized for the given example.

	Gender	C_Freq	C_Diff	C_Single	C_Same
VIF	1.03	1.03	1.88	1.02	1.53
	Gender	I_Freq	I_Diff	I_Single	I_Same
VIF	1.03	1.04	1.58	1.04	1.24

Table 38: VIF scores for the predictors. C_: Correct, I_: Incorrect

Model Shape We analyze the performance of the coreference resolvers at the level of individual predictions using following logistic regression model:

$$\begin{aligned} \text{p(Correct)} \sim \sigma(&\text{Gender} + \\ &\text{C\_Freq} + \text{I\_Freq} + \\ &\text{C\_Diff} + \text{I\_Diff} + \\ &\text{C\_Single} + \text{I\_Single} + \\ &\text{C\_Same} + \text{I\_Same}) \end{aligned} \tag{10.9}$$

where σ is the logistic function. p(Correct): is 1 if the resolver matches the pronoun with the correct named entity in corresponding instance and 0 otherwise. For Gender, 1 means female and 0 male. For Single, 1 means the entity is a single word, 0 otherwise. For Same, 1 means the entity is in the same sentence as the pronoun, 0 otherwise. We use Dominance Analysis to determine relative importance of each predictor.

In this setup, the regression model predicts whether each of the system predictions is correct or incorrect. To the extent the correctness is affected by the properties of the discourse captured by our predictors, we will obtain significant effects; conversely, should the correctness be fully random or dependent on properties independent from our predictors, we will not see significant effects.

Model Validation Table 38 shows the results of multicollinearity analysis on the set of predictors. All VIF values are smaller than 2, which indicates the absence of problematic multicollinearity.

10.4.4 Results

Table 39 shows the performance of six resolvers on the complete GAP development set (overall and separately for Male and Female). It probably does not come as a surprise that performance increases over time; it is positive to note, though, that the Bias decreases correspondingly. Table 40 shows the main results of our regression analysis on the subset of the GAP development set with a correct solution (cf. Section 10.4.1), organized by columns (systems). Each row provides a regression coefficient with its confidence level as well as the relative importance score for the predictor, using Dominance Analysis (DA). R^2_{VZ} indicates the goodness of fit values at the level of complete systems. Note that these numbers, computed for logistic regression models, are not comparable to the numbers for linear regression models from Experiment 1.

We also report accuracy values for the predictions of our logistic regression model, averaged over 10-fold cross-validation (*Acc*). Numbers in parentheses indicate the accuracy of corresponding majority baselines. The differences in baseline scores across systems are due to the fact that gold labels (i.e., the p(Correct) variable in the equation) are dependent on system predictions.

	Male	Female	All	Bias
Lee et al. (2013)	55.4	45.5	50.5	0.82
Clark & Manning (2015)	58.5	51.3	55.0	0.88
Wiseman et al. (2016)	68.4	59.9	64.2	0.88
Lee et al. (2017)	67.2	62.2	64.7	0.92
Lee et al. (2018)	75.9	72.1	74.0	0.95
Joshi et al. (2020)	89.9	87.8	88.8	0.98

Table 39: F_1-Scores of resolvers on the GAP development set (Bias=F_1 Female / F_1 Male)

System level analysis We first discuss results at the system level. The last row of Table 40 (Model Fit) shows the overall model fit for all systems. The ability of our regression model to outperform majority baselines for the first four systems (Lee et al., 2013; Clark and Manning, 2015; Wiseman et al., 2016; Lee et al., 2017) shows that our analysis can predict mistakes made by these coreference resolvers by only considering a small set of discourse-related features plus Gender. In contrast, Lee et al. (2018) and Joshi et al. (2020) both show an R^2_{VZ} of almost zero, that is, the logistic regression models perform at the level of a majority class baseline – the remaining errors that they systems make are idiosyncratic rather than systematic. These findings tie in well with the overall system performance scores shown in Table 39.

It is striking that Joshi et al. (2020), the best model by a substantial margin, is also the one exhibiting the smallest bias. We see two possible explanations: (a), the model was trained on a large cor-

		Lee et al. (2013)	Clark and Manning (2015)	Wiseman et al. (2016)	Lee et al. (2017)	Lee et al. (2018)	Joshi et al. (2020)
Gender	Coef	-0.473^{***}	-0.308^{**}	-0.314^{**}	-0.271^{**}	-0.215^{*}	-0.084
	DA	0.008	0.004	0.004	0.003	0.002	0.000
C_Freq	Coef	0.004	0.018^{***}	-0.004	-0.003	-0.001	0.001
	DA	0.000	0.005	0.000	0.000	0.000	0.000
I_Freq	Coef	-0.003	-0.003	-0.004	-0.006	-0.003	0.003
	DA	0.000	0.000	0.000	0.001	0.000	0.000
C_Diff	Coef	1.291^{**}	-1.617^{***}	$-0.933^{\cdot}$	-0.337	0.608	-0.065
	DA	0.006	0.002	0.001	0.001	0.001	0.000
I_Diff	Coef	-1.027^{*}	1.444^{***}	-0.086	$-0.740^{\cdot}$	-0.510	-0.053
	DA	0.003	0.002	0.001	0.004	0.001	0.000
C_Single	Coef	0.344^{**}	0.475^{***}	0.775^{***}	0.666^{***}	0.554^{***}	0.171
	DA	0.004	0.008	0.021	0.016	0.010	0.001
I_Single	Coef	-0.053	$-0.166^{\cdot}$	-0.268^{**}	-0.346^{***}	-0.360^{***}	0.036
	DA	0.001	0.001	0.003	0.006	0.006	0.000
C_Same	Coef	-0.603^{***}	-0.456^{***}	-0.561^{***}	-0.564^{***}	-0.336^{*}	-0.007
	DA	0.015	0.002	0.007	0.008	0.004	0.000
I_Same	Coef	0.086	0.366^{**}	0.120	0.318^{**}	0.317^{**}	0.028
	DA	0.000	0.002	0.000	0.003	0.003	0.000
Model Fit	R^2_{VZ}	0.05	0.04	0.05	0.05	0.03	0.01
	Acc	0.61 (0.58)	0.57 (0.55)	0.58 (0.53)	0.59 (0.53)	0.63 (0.63)	0.55 (0.55)

Table 40: Regression-based analysis of coreference resolution systems on GAP dataset. DA: Dominance Analysis, Freq: Frequency, C_: Correct, I_: Incorrect instances.

pus from several domains with different discourse style, which may make it more robust to gender bias (Saunders and Byrne, 2020); (b) in contrast to the older studies, this model is based on contextualized embeddings, which also showed lower bias in Experiment 1. Without re-training the model, we cannot currently distinguish between these two explanations.

Predictor level analysis We now move on to investigate the contribution of each predictor to the systems' predictability. At this level, gender is a statistically significant predictor ($p < 0.05$) for all systems except Joshi et al. (2020). It has a negative sign throughout, indicating worse performance for female entities. This is again in line with the findings reported in Table 39. However, our approach reveals other important patterns which cannot be observed by using traditional analysis methods. First, Clark and Manning (2015) and Wiseman et al. (2016) have the same DA coefficient for gender variable but different R^2_{VZ} values. We interpret this to mean that contribution of gender bias to overall bias in these two systems is not the same, an observation that would not have been possible through traditional bias analysis methods (cf. Table 39).

Second, we see that the coefficient signs of the predictors C_Single and C_Same remain the same across systems: Systems perform better for instances where the correct antecedent is a single word, and it is not in the same sentence with the pronoun. Moreover, dominance analysis shows that these two predictors are among the main contributors to the biased predictions in four systems out of six, the two exceptions being Lee et al. (2013) and Joshi et al. (2020).

Third, the small but consistent positive relative importance values of the C_Diff and I_Diff predictors for half of the systems show that these variables help explain the systems' predictions. In contrast, the low relative importance values of the C_Frequency and I_Frequency predictors indicate that these variables do not affect coreference resolution much.

		I_Same=0	I_Same=1
Lee et al. (2017)	C_Same=0	**0.80**	1.10
	C_Same=1	0.90	0.90
		I_Same=0	I_Same=1
Lee et al. (2018)	C_Same=0	0.90	1.00
	C_Same=1	**0.86**	0.97
		I_Same=0	I_Same=1
Joshi et al. (2020)	C_Same=0	1.02	1.02
	C_Same=1	0.99	**0.94**

Table 41: Bias values for the three best performing systems, with data split into four groups according to C_Same and I_Same (worst bias marked in boldface).

Interpretability These detailed findings indicate that, similar to emotion intensity prediction, the analysis of coreference resolvers can also benefit from not only the controlled bias variable but also from other properties of the input even in datasets which are designed carefully to isolate the effect of the target variable. As stated in Exp. 1, these analyses can also be used to extract interesting examples and subsets.

We illustrate this for the two attributes C_Same and I_Same, i.e., whether the correct and incorrect antecedent are in the same sentence or not. We split the GAP dataset into four reasonably-sized subsets based on the values of these attributes: the subset where both are in the same sentence (C_Same=1 and I_Same=1) includes $\sim$ 900 examples and the other three subsets include $\sim$ 300 examples. Table 41 shows the bias values (defined as above) for the three best performing systems. We observe that these systems vary widely regarding the subset where gender bias is most prominently visible varies across systems: Lee et al. (2017, 2018) both show the worst bias when the incorrect antecedent is not in the current sentence (I_Same=0), but differ in the effect of the position of the correct antecedent (C_Same). In contrast, Joshi et al. (2020) perform almost perfectly when I_Same=0, but struggles most the case when both correct and incorrect antecedent are in the current sentence.

10.5 Conclusion

In this chapter, we have argued that bias analysis, a task of major importance concerning the societal implications of NLP, can benefit from richer statistical methods to detect, quantify and attribute bias. We have proposed to follow other scientific fields in adopting regression analysis which (a) generalizes to multiple bias variables, (b) can quantify the contribution of confounder variables to the observed bias with measures of effect size, and (c) can be used to diagnose system behavior and extract informative datapoints.

Clearly, regression analysis doesn't solve all the problems involved in bias analysis: it presupposes a set of plausible covariates of bias, which can come from a wide variety of sources, including task-specific annotation, task-unspecific input representations, or model architecture (Hovy and Prabhumoye, 2021). Such covariates are typically known through domain expertise or uncovered by exploratory data analysis. Thus, regression analysis complements, but does not replace, traditional methods of bias analysis.

We have demonstrated the usefulness of our approach by analyzing a range of model architectures on a regression task and a classification task, obtaining model-level results that are in line with the existing literature, e.g., BERT-based systems appear to exhibit comparatively little bias (Basta et al., 2019). In addition, adding predictor-level analysis offers a richer understanding of the importance of the bias variables and their interactions with other textual properties. Note that we only considered datasets specifically designed to exhibit the effects of a single bias variable. We believe that the benefits of our analysis framework would be even clearer on more naturalistic datasets such as MSP-Podcast (Gorrostieta et al., 2019) , where pairwise hypothesis tests become even more problematic however we leave this as future work.

Another methodological debate that we hope to contribute to is what constitutes a substantial bias? We have argued that effect sizes offer a statistically sound approach to measuring the amount of variation in the output that can be attributes to a set of input properties. Our study provides a starting point for the community to establish a magnitude for what it considers a "substantial" bias, similar to the often used thresholds for inter-annotator agreement (Cohen, 1968) or general effect sizes in psychology (Cohen, 1988).

Lastly, we would like to acknowledge that we believe that the regression based bias analysis approaches presented in this chapter can be applied to previous bias analysis studies we conducted through Chapters 7-9 although we could not perform this due to time constraints. Such an analysis would be useful, for example, to extend the scope of the correlation analysis study we performed in Chapter 9 from two variables to many.

Part IV

Conclusion and Future Work

11 Conclusion and Future Work

This thesis is devoted to investigating challenges of computational social science analysis with NLP methods. Our work makes contributions on two directions, namely system-wise and fairness-wise:

Our first system-wise contribution was to show how analyses in CSS that are complex by nature and traditionally performed manually, can be automated by first decomposing them into several sub-tasks that are conceptually simpler and then developing NLP models to automatically perform each of these sub-tasks. While the idea of breaking down complex tasks into smaller pieces is not new and has been applied in other contexts in CSS, such as social network extraction from historical data (van de Camp, 2016), construction of graphs capturing emotion interactions between fictional characters (Kim and Klinger, 2019) or transformation of raw text to argument graphs (Mirko et al., 2020), we are the first, to the best of our knowledge, to demonstrate this in the context of discourse network analysis, an aspiring analysis approach in political science to understand the structure and temporal dynamics of political debates.

Our second contribution in this aspect was to create models for the first two components of the workflow which are responsible for identification of concept nodes in affiliation networks. We developed and evaluated a range of semantic NLP models comprising RNNs, CNNs, LSTMs and Transformers for the tasks of Political Claim Detection which is part of Argument Mining (Peldszus and Stede, 2013) in a domain specific setting, and Claim Classification, a special type of text classification with shorter input text and hundreds of categories.

Our third contribution was to perform a case study on a manually annotated corpus of the German migration debate, using the workflow and NLP methods that we developed. Through our claim detection experiments, we showed that claim detection can be modeled as a text classification task as well as a sequence labeling task, and in both cases it is possible to detect the political claims in debates automatically with a reasonable performance using transformer based models. Similarly, our claim classification experiments revealed two important insights. First, an attention mechanism is one of the key factors for claim classification models to achieve good performance. Second, there is no single winner model. While in some circumstances, such as where there is enough data for model fine-tuning, BERT-based claim classification model performs best, in other cases it might be overkill and less complex models, such as BiLSTM+ATTN model can yield better performance. This finding is in line with previous research (e.g., Yan et al. (2019); Lai et al. (2021, 2022)), which show that although transformer-based models are able to set new standards and achieve state-of-the-art results across many NLP tasks and datasets, it is sometimes possible that a simpler model can be as effective as or even more effective than them.

Regarding the fairness aspect, our first contribution was to evaluate frequency bias in our claim detection model, which is one of the components of the automatic discourse network construction workflow. Through our analyses, we found that claim detection model makes spurious correlation between actor frequency and positive label, causing the model to recognize claims made by infrequent actors with much worse recall. This finding contributes to the literature by extending the research on the relationships between frequency and performance of the statistical models, which has previously made similar discoveries (i.e. models perform worse on infrequent instances) for different tasks such as Part-of-Speech tagging (Bhatia et al., 2016) and sentiment analysis (Wang et al., 2021a). Following our finding of frequency bias in the claim detection model, we proposed various debiasing methods. Our empirical results reveal that besides reducing bias, a simple masking of names and pronouns can improve classification performance too.

Later, we extended our frequency related analysis to claim classification task. Through our experiments, we found that due to the skewed distribution of fine-grained categories, most of which are infrequently attested (which is not specific to the codebook that we used in our case-study but arises typically in many CSS codebooks) standard claim classification models, even the ones based on state-of-the-art architectures (e.g., transformers), can't perform well, in particular for infrequent categories. As a solution, we proposed to include the domain knowledge into the models, which has been shown to be effective for improving performance for different tasks in CSS and NLP in general, such as crime association analysis (Schroeder et al., 2003), social media analysis (Declerck, 2013), and biomedical data analysis (Wu et al., 2022). Specifically, we utilize domain knowledge by integrating the is-a relations between super- and sub-categories available in the codebooks into the models using lightweight methods. Our evaluation on two datasets showed that our proposed approach leads to better overall performance as well as better algorithmic fairness. This was our second contribution.

Our third contribution was to draw attention to the general question of prerequisites for successful adversarial debiasing, one of the most popular debiasing approach utilized to unlearn the spurious correlations models make between between target classes and other textual attributes of the data. We argued that adversarial debiasing fails when the target attribute is subsumed by the bias attribute, as in that case debiasing the bias attribute would create an empty feature space for target attribute.

Finally, our last contribution in this aspect was a methodological one. After identifying limitations of the statistical methods that are currently applied for bias identification, we proposed an approach based on multivariate regression that can be used as a complementary method for bias identification and analysis. We showed that our method leads to model-level results that are in line with the existing literature (Kiritchenko and Mohammad, 2018; Webster et al., 2018). On top of that, it offers richer understanding of the importance of the bias variables (i.e. practical importance of biases (Sullivan and Feinn, 2012) using effect size measurements) and their interactions with other textual properties as a result of its ability to generalize to multiple bias variables and to take covariates into account.

Future work. Our work has opened a number of new possibilities for future work that are worth exploring. As part of this thesis, we developed machine learning based NLP models for claim detection and classification tasks, enabling (semi-)automatic construction of discourse networks. A natural extension to this work would be to design and developed NLP models to automatize the remaining steps of the workflow as described in Chapter 4. Such an extension would allow researchers to build up the discourse networks in a fully automatic way (thus reducing the human-effort), and work with massive collections of data, but at the same time, this might also make the overall system more vulnerable to issues such as fairness, and robustness that the statistical methods are faced frequently (Grimmer and Stewart, 2013; Papakyriakopoulos, 2020).

One main limitation of the models presented through this thesis for computational construction of the discourse networks is that they are trained in a fully supervised manner. It means that time-consuming and expensive data annotation and codebook generation steps are still required (to obtain labeled data sets for the purpose of training the models), making the process sub-optimal especially from the perspective of domain experts such as political scientists. One way to reduce the cost of data annotation required for model training would be to rely on some human-in-the-loop approaches such as Active Learning in which only instances that would be most helpful to the model are shown to the annotator by the machine (i.e active learner) and the redundant instances are discarded (Druck et al., 2009). Additionally, to tackle the labeled data problem, future work may also consider developing semi-supervised or unsupervised models for these tasks which require less or no labeled data at all.

Such an extension would allow to perform computational analysis on domains and languages, where no labeled data available. On the negative side, however,they tend to suffer from relatively lower performance especially when the number of target categories is in the hundreds (Aggarwal and Zhai, 2012; Baker et al., 2016).

It is not always obvious how the differences in intrinsic evaluation metrics for each part of a complicated program affect the performance of the overall application that uses it (Kovár et al., 2016). Considering this, another possibility for future work related to automatic construction of discourse networks would be to perform model evaluation based on the quality of resulting discourse network. As a concrete example, it could be interesting to assess impact of the lightweight hierarchy encoding methods that we proposed in Chapter 8 to improve fine-grained claim classification on the quality of resulting discourse network. This idea has also been exploited by various NLP researchers in the past. For example, Yuret et al. (2010) compared different parsers based on how much they contribute to the performance of a textual entailment system, and Dzikovska et al. (2012) developed a generic framework to evaluate the dialogue systems used in complex natural language understanding applications.

Furthermore, we also suggest future work to further develop our regression analysis based bias identification system introduced in Chapter 10. A possible improvement in this direction would be to develop richer regression models that analyze interactions among predictors. Such interactions, when properly motivated, can further improve our understanding of the performance data.

Last but not least, we would like to mention that the computational models developed through this thesis would be also useful for other tasks/analysis in Political Science beyond the original purpose of development. For instance, the models we introduced for identification and fine-grained classification of claims can also be used for identifying socio-political events in raw text and classifying them into fine-grained categories (Hürriyetoğlu et al., 2021), which is yet another challenging CSS task that social and political scientists have been working in order to create socio-political event databases.

Bibliography

Achen, C. H. (1982). *Interpreting and using regression*, volume 29 of *Quantitative Applications in the Social Sciences*. Sage.

Aggarwal, C. C. and Zhai, C. (2012). A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer.

Anand, P., Walker, M., Abbott, R., Tree, J. E. F., Bowmani, R., and Minor, M. (2011). Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9.

Aono, M. and Himeno, S. (2018). KDE-AFFECT at SemEval-2018 Task 1: Estimation of affects in tweet by using convolutional neural network for n-gram. In *Proceedings of SemEval*, pages 156–161, New Orleans, LA.

Atanasova, P., Simonsen, J. G., Lioma, C., and Augenstein, I. (2020). Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

Azen, R. and Traxel, N. (2009). Using dominance analysis to determine predictor importance in logistic regression. *Journal of Educational and Behavioral Statistics*, 34:319 –347.

Baayen, H. (2008). *Analyzing Linguistic Data.* Cambridge University Press.

Bach, T., Hammerschmid, G., and Löffler, L. (2020). More delegation, more political control? politicization of senior-level appointments in 18 european countries. *Public Policy and Administration*, 35(1):3–23.

Baker, S., Kiela, D., and Korhonen, A. (2016). Robust text classification for sparsely labelled data using multi-level embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2333–2343, Osaka, Japan. The COLING 2016 Organizing Committee.

Balaji, T., Annavarapu, C. S. R., and Bablani, A. (2021). Machine learning algorithms for social media analysis: A survey. *Computer Science Review*, 40:100395.

Bansal, R. (2022). A survey on bias and fairness in natural language processing. *arXiv preprint arXiv:2204.09591*.

Barr, D. J., Levy, R., Scheepers, C., and Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3):255–278.

Basta, C., Costa-jussà, M. R., and Casas, N. (2019). Evaluating the underlying gender bias in contextualized word embeddings. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 33–39, Florence, Italy. Association for Computational Linguistics.

Bates, D., Kliegl, R., Vasishth, S., and Baayen, H. (2018). Parsimonious mixed models. ArXiv preprint, `http://arxiv.org/abs/1506.04967`.

Baumgartner, F. R., Green-Pedersen, C., and Jones, B. D. (2006).

Comparative studies of policy agendas. *Journal of European public policy*, 13(7):959–974.

Beinborn, L., Zesch, T., and Gurevych, I. (2014). Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–530.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.

Bhatia, P., Guthrie, R., and Eisenstein, J. (2016). Morphological priors for probabilistic neural word embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 490–500.

Bilbao-Jayo, A. and Almeida, A. (2021). Improving political discourse analysis on twitter with context analysis. *IEEE Access*, 9:104846–104863.

Blokker, N., Blessing, A., Dayanik, E., Kuhn, J., Padó, S., and Lapesa, G. (2021). Between welcome culture and border fence. a dataset on the european refugee crisis in german newspaper reports. *arXiv preprint arXiv:2111.10142*.

Blokker, N., Dayanik, E., Lapesa, G., and Padó, S. (2020). Swimming with the tide? positional claim detection across political text types. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 24–34, Online. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017a). Enriching word vectors with subword information. *Transactions* *of the Association for Computational Linguistics*, 5:135–146.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017b). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Proceedings of NIPS*, pages 4349–4357.

Bordia, S. and Bowman, S. (2019). Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15.

Boydstun, A. E., Card, D., Gross, J., Resnick, P., and Smith, N. A. (2014). Tracking the development of media frames within and across policy issues. *American Political Science Association Annual Meeting.*

Bresnan, J., Cueni, A., Nikitina, T., and Baayen, H. (2007). Predicting the dative alternation. In Bouma, G., Kraemer, I., and Zwarts, J., editors, *Cognitive Foundations of Interpretation*, pages 69–94. Royal Netherlands Academy of Science.

Bsoul, M. A., Qusef, A., and Abu-Soud, S. (2022). Building an optimal dataset for arabic fake news detection. *Procedia Computer Science*, 201:665–672.

Budescu, D. V. (1993). Dominance analysis: a new approach to the problem of relative importance of predictors in multiple regression. *Psychological bulletin*, 114(3):542.

Bui, T. D., Ravi, S., and Ramavajjala, V. (2018). Neural graph learning: Training neural networks using graphs. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 64–71, New York, NY, USA. Association for Computing Machinery.

Burstein, P. (2010). Public opinion, public policy, and democracy. In *Handbook of politics*, pages 63–79. Springer.

Büyüköz, B., Hürriyetoğlu, A., and Özgür, A. (2020). Analyzing ELMo and DistilBERT on socio-political news classification. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 9–18, Marseille, France. European Language Resources Association (ELRA).

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Campagnano, C., Conia, S., and Navigli, R. (2022). Srl4e–semantic role labeling for emotions: A unified evaluation framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4586–4601.

Card, D., Boydstun, A. E., Gross, J. H., Resnik, P., and Smith, N. A. (2015). The media frames corpus: Annotations of frames across issues. 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 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 438–444. The Association for Computer Linguistics.

Cardie, C. and Wilkerson, J. (2008). Text annotation for political science research.

Caucheteux, C. and King, J.-R. (2022). Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1):134.

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit3: Web inventory of transcribed and translated talks. In *Conference of European Association for Machine Translation*, pages 261–268, Trento, Italy.

Chen, X., Wang, Y., Agichtein, E., and Wang, F. (2015). A comparative study of demographic attribute inference in twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Chen, Y., Hou, W., Cheng, X., and Li, S. (2018). Joint learning for emotion classification and emotion cause detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 646–651, Brussels, Belgium. Association for Computational Linguistics.

Cheng, J., Danescu-Niculescu-Mizil, C., and Leskovec, J. (2015). Antisocial behavior in online discussion communities. In *Proceedings of the International AAAI Conference on Web and Social Media*.

Choi, D. and Rhee, W. (2019). Utilizing class information for deep network representation shaping. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Clark, A., Fox, C., and Lappin, S. (2012). *The handbook of computational linguistics and natural language processing*, volume 118. John Wiley & Sons.

Clark, K. and Manning, C. D. (2015). Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China. Association for Computational Linguistics.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition.

Cohen, R. and Ruths, D. (2013). Classifying political orientation on twitter: It's not easy! In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 7, pages 91–99.

Corbett, G. G. (1991). *Gender*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Cotterell, R. and Heigold, G. (2017). Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

Dahlström, C. and Holmgren, M. (2019). The political dynamics of bureaucratic turnover. *British Journal of Political Science*, 49(3):823–836.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. (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, Florence, Italy. Association for Computational Linguistics.

Davidson, T., Bhattacharya, D., and Weber, I. (2019). Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Daxenberger, J., Eger, S., Habernal, I., Stab, C., and Gurevych, I. (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, Copenhagen, Denmark. Association for Computational Linguistics.

De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., and Kalai, A. T. (2019). Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128.

De Wilde, P. (2011). No polity for old politics? a framework for analyzing the politicization of european integration. *Journal of European integration*, 33(5):559–575.

Declerck, T. (2013). Integration of the thesaurus for the social sciences (TheSoz) in an information extraction system. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 90–95, Sofia, Bulgaria. Association for Computational Linguistics.

Del Vicario, M., Vivaldo, G., Bessi, A., Zollo, F., Scala, A., Caldarelli, G., and Quattrociocchi, W. (2016). Echo chambers: Emotional contagion and group polarization on facebook. *Scientific*

Reports, 6(1):37825.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Díaz, M., Johnson, I., Lazar, A., Piper, A. M., and Gergle, D. (2018). Addressing age-related bias in sentiment analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Dieng, A. B., Wang, C., Gao, J., and Paisley, J. (2016). Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*.

Doll, R., Peto, R., Boreham, J., and Sutherland, I. (2004). Mortality in relation to smoking: 50 years' observations on male british doctors. *BMJ*, 328(7455):1519.

Druck, G., Settles, B., and McCallum, A. (2009). Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 81–90, Singapore. Association for Computational Linguistics.

Dzikovska, M. O., Bell, P., Isard, A., and Moore, J. D. (2012). Evaluating language understanding accuracy with respect to objective outcomes in a dialogue system. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 471–481, Avignon, France. Association for Computational Linguistics.

Elazar, Y. and Goldberg, Y. (2018). Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.

Fairclough, N. (2013). Deliberation as genre in the parliamentary debate on university tuition fees. In *Political discourse analysis*, pages 210–244. Routledge.

Fan, C., Yan, H., Du, J., Gui, L., Bing, L., Yang, M., Xu, R., and Mao, R. (2019). A knowledge regularized hierarchical approach for emotion cause analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5614–5624, Hong Kong, China. Association for Computational Linguistics.

Farra, N., Tomeh, N., Rozovskaya, A., and Habash, N. (2014). Generalized character-level spelling error correction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–167, Baltimore, Maryland. Association for Computational Linguistics.

Farzindar, A. and Inkpen, D. (2017). Natural language processing for social media. *Synthesis Lectures on Human Language Technologies*, 10(2):1–195.

Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data.* Cambridge university press.

Friedman, B. and Nissenbaum, H. (1996). Bias in computer systems. *ACM Trans. Inf. Syst.*, 14(3):330–347.

Gao, Y., Eger, S., Kuznetsov, I., Gurevych, I., and Miyao, Y. (2019). Does my rebuttal matter? insights from a major NLP conference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1274–1290, Minneapolis, Minnesota. Association for Computational Linguistics.

Gardner, M. J. (1973). Using the environment to explain and predict mortality. *Journal of the Royal Statistical Society. Series A (General)*, 136(3):421–440.

Garg, N., Schiebinger, L., Jurafsky, D., and Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

Garibo i Orts, Ò. (2019). Multilingual detection of hate speech against immigrants and women in Twitter at SemEval-2019 task 5: Frequency analysis interpolation for hate in speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 460–463, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Gerz, D., Vulić, I., Ponti, E. M., Reichart, R., and Korhonen, A. (2018). On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of EMNLP*, pages 316–327, Brussels, Belgium.

Glavaš, G., Nanni, F., and Ponzetto, S. P. (2017). Cross-lingual classification of topics in political texts. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 42–46, Vancouver, Canada. Association for Computational Linguistics.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of ICML*, pages 513–520, Bellevue, WA.

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309.

Gonen, H. and Goldberg, Y. (2019). Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 60–63, Florence, Italy. Association for Computational Linguistics.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Gorrostieta, C., Lotfian, R., Taylor, K., Brutti, R., and Kane, J. (2019). Gender de-biasing in speech emotion recognition. In *Proceedings of Interspeech*, pages 2823–2827.

Gottron, T. and Lipka, N. (2010). A comparison of language identification approaches on short, query-style texts. In *European Conference on Information Retrieval*, pages 611–614. Springer.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Graves, A., Jaitly, N., and Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE work-*

shop on automatic speech recognition and understanding, pages 273–278. IEEE.

Greenwald, A. G., McGhee, D. E., and Schwartz, J. L. (1998). Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology*, 74(6):1464.

Grimmer, J. and Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3):267–297.

Grömping, U. (2006). Relative importance for linear regression in R: the package `relaimpo`. *Journal of statistical software*, 17(1):1–27.

Gross, J. J., Carstensen, L. L., Pasupathi, M., Tsai, J., Götestam Skorpen, C., and Hsu, A. Y. (1997). Emotion and aging: Experience, expression, and control. *Psychology and aging*, 12(4):590.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Hajat, A., Hsia, C., and O'Neill, M. S. (2015). Socioeconomic disparities and air pollution exposure: a global review. *Current Environmental Health Reports*, 2(4):440–450.

Hall Maudslay, R., Gonen, H., Cotterell, R., and Teufel, S. (2019). It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. In *Proceedings of the 2019 Confer-*

ence on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5267–5275, Hong Kong, China. Association for Computational Linguistics.

Hammarström, H. (2007). A fine-grained model for language identification. In *Proceedings of iNEWS-07 Workshop at SIGIR 2007*, pages 14–20.

Han, X., Dasgupta, S., and Ghosh, J. (2021). Simultaneously reconciled quantile forecasting of hierarchically related time series. In *International Conference on Artificial Intelligence and Statistics*, pages 190–198. PMLR.

Hanson, S. and Pratt, L. (1988). Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, 1.

Harris, J. W. (1991). The exponence of gender in spanish. *Linguistic Inquiry*, 22(1):27–62.

Haunss, S., Dietz, M., and Nullmeier, F. (2013). Der ausstieg aus der atomenergie. diskursnetzwerkanalyse als beitrag zur erklärung einer radikalen politikwende. *Zeitschrift für Diskursforschung*, 1:288–316.

Haunss, S., Kuhn, J., Padó, S., Blessing, A., Blokker, N., Dayanik, E., and Lapesa, G. (2020). Integrating manual and automatic annotation for the creation of discourse network data sets. *Politics and Governance*, 8:326–339.

He, Z., Wang, Z., Wei, W., Feng, S., Mao, X., and Jiang, S. (2020). A survey on recent advances in sequence labeling from deep learning models. *arXiv preprint arXiv:2011.06727*.

Heindorf, S., Scholten, Y., Engels, G., and Potthast, M. (2019).

Debiasing vandalism detection models at wikidata. In *The World Wide Web Conference*, pages 670–680.

Heinisch, P. and Cimiano, P. (2021). A multi-task approach to argument frame classification at variable granularity levels. *it-Information Technology*, 63(1):59–72.

Helbling, M. and Tresch, A. (2011). Measuring party positions and issue salience from media coverage: Discussing and cross-validating new indicators. *Electoral Studies*, 30(1):174–183.

Hemphill, L., Russell, A., and Schöpke-Gonzalez, A. M. (2021). What drives u.s. congressional members' policy attention on twitter? *Policy & Internet*, 13(2):233–256.

Heywood, A. (2015). *Key concepts in politics and international relations*. Macmillan International Higher Education.

Hillard, D., Purpura, S., and Wilkerson, J. (2008). Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics*, 4(4):31–46.

Hinton, G., Vinyals, O., Dean, J., et al. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Hong, T., Choi, C., and Shin, J. (2018). Cnn-based malicious user detection in social networks. *Concurrency and Computation: Practice and Experience*, 30(2):e4163.

Hopkins, D. J. and King, G. (2010). A method of automated non-parametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247.

Hovy, D. (2015). Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Pa-*

pers), pages 752–762, Beijing, China. Association for Computational Linguistics.

Hovy, D. and Prabhumoye, S. (2021). Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8):e12432.

Hovy, D. and Spruit, S. L. (2016). The social impact of natural language processing. In *Proceedings of ACL*, pages 591–598, Berlin, Germany.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer.

Hürriyetoğlu, A., Tanev, H., Zavarella, V., Piskorski, J., Yeniterzi, R., Yuret, D., and Villavicencio, A. (2021). Challenges and applications of automated extraction of socio-political events from text (case 2021): Workshop and shared task report. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 1–9.

Hürriyetoğlu, A., Yörük, E., Yüret, D., Yoltar, Ç., Gürel, B., Duruşan, F., Mutlu, O., and Akdemir, A. (2019). Overview of clef 2019 lab protestnews: Extracting protests from news in a cross-context setting. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 425–432. Springer.

Hürriyetoğlu, A., Zavarella, V., Tanev, H., Yörük, E., Safaya, A., and Mutlu, O. (2020). Automated extraction of socio-political events from news (AESPEN): Workshop and shared task report. In *Proceedings of the Workshop on Automated Extraction of Socio-*

political Events from News 2020, pages 1–6, Marseille, France. European Language Resources Association (ELRA).

Islam, J., Mercer, R. E., and Xiao, L. (2019). Multi-channel convolutional neural network for Twitter emotion and sentiment recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1355–1365, Minneapolis, Minnesota. Association for Computational Linguistics.

Jaeger, T. F. (2008). Categorical data analysis: Away from anovas (transformation or not) and towards logit mixed models. *Journal of Memory and Language*, 59(4):434–446. Special Issue: Emerging Data Analysis.

James, H. and Alvarez-Melis, D. (2019). Probabilistic bias mitigation in word embeddings. *arXiv preprint arXiv:1910.14497*.

Jin, X., Barbieri, F., Kennedy, B., Mostafazadeh Davani, A., Neves, L., and Ren, X. (2021). On transferability of bias mitigation effects in language model fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3770–3783, Online. Association for Computational Linguistics.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Kanakaraddi, S. G. and Nandyal, S. S. (2018). Survey on parts of speech tagger techniques. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–6.

Kaneko, M. and Bollegala, D. (2021). Debiasing pre-trained contextualised embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1256–1266, Online. Association for Computational Linguistics.

Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68.

Karan, M., Šnajder, J., Širinić, D., and Glavaš, G. (2016). Analysis of policy agendas: Lessons learned from automatic topic classification of Croatian political texts. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 12–21, Berlin, Germany. Association for Computational Linguistics.

Kersting, N. (2005). The quality of political discourse: Can e-discussion be deliberative. In *PSA conference in Leeds*.

Kim, E. and Klinger, R. (2019). Frowning frodo, wincing leia, and a seriously great friendship: Learning to classify emotional relationships of fictional characters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 647–653.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Kiritchenko, S. and Mohammad, S. (2018). Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of STARSEM*, pages 43–53, New Orleans, LA.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT.

Koopmans, R. (2002). Codebook for the analysis of political mobilisation and communication in european public spheres. *Codebook from the Project: The Transformation of Political Mobilization and Communication in European Public Spheres. 5th Framework Program of the European Commission. Europub. com.*

Koopmans, R. and Statham, P. (1999). Political claims analysis: Integrating protest event and political discourse approaches. *Mobilization: An International Quarterly*, 4.

Koppel, M., Argamon, S., and Shimoni, A. R. (2004). Automatically categorizing written texts by author gender. *Computing Reviews*, 45(1):43.

Kovár, V., Jakubıcek, M., and Horák, A. (2016). On evaluation of natural language processing tasks. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*, pages 540–545.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4):150.

Kucukyilmaz, T., Cambazoglu, B. B., Aykanat, C., and Can, F. (2006). Chat mining for gender prediction. In *International conference on advances in information systems*, pages 274–283. Springer.

Kumar, S., Wintner, S., Smith, N. A., and Tsvetkov, Y. (2019). Topics to avoid: Demoting latent confounds in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4153–4163, Hong Kong, China. Association for Computational Linguistics.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lai, M., Tambuscio, M., Patti, V., Ruffo, G., and Rosso, P. (2019). Stance polarity in political debates: A diachronic perspective of network homophily and conversations on twitter. *Data & Knowledge Engineering*, 124:101738.

Lai, T., Ji, H., and Zhai, C. (2021). BERT might be overkill: A tiny but effective biomedical entity linker based on residual convolutional neural networks. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1631–1639, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lai, T. M., Bui, T., and Kim, D. S. (2022). End-to-end neural coreference resolution revisited: A simple yet effective baseline. In

ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8147–8151. IEEE.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Lapesa, G., Blessing, A., Blokker, N., Dayanik, E., Haunss, S., Kuhn, J., and Padó, S. (2020). DEbateNet-mig15:tracing the 2015 immigration debate in Germany over time. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 919–927, Marseille, France. European Language Resources Association.

Lauscher, A. and Glavaš, G. (2019). Are we consistently biased? multidimensional analysis of biases in distributional word vectors. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 85–91, Minneapolis, Minnesota. Association for Computational Linguistics.

Lauscher, A., Takieddin, R., Ponzetto, S. P., and Glavaš, G. (2020). AraWEAT: Multidimensional analysis of biases in Arabic word embeddings. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 192–199, Barcelona, Spain (Online). Association for Computational Linguistics.

Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabasi, A.-L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., et al. (2009). Social science. computational social science. *Science (New York, NY)*, 323(5915):721–723.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Lee, J. Y. and Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520, San Diego, California. Association for Computational Linguistics.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Leifeld, P. (2009). Die untersuchung von diskursnetzwerken mit dem discourse network analyzer (dna). In Schneider, V., Janning, F., Leifeld, P., and Malang, T., editors, *Politiknetzwerke. Modelle, Anwendungen und Visualisierungen*, chapter B, pages 391–404. VS Verlag für Sozialwissenschaften, Wiesbaden.

Leifeld, P. (2017). Discourse network analysis. *The Oxford handbook of political networks*, pages 301–326.

Leifeld, P. and Haunss, S. (2011). Political discourse networks and the conflict over software patents in europe. *European Journal of*

Political Research, 51:382 – 409.

Leifeld, P. and Haunss, S. (2012). Political Discourse Networks and the Conflict over Software Patents in Europe. *European Journal of Political Research*, 51(3):382–409.

Li, S., Xia, R., Zong, C., and Huang, C.-R. (2009). A framework of feature selection methods for text categorization. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 692–700, Suntec, Singapore. Association for Computational Linguistics.

Li, W. and Dickinson, M. (2017). Gender prediction for Chinese social media data. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 438–445, Varna, Bulgaria. INCOMA Ltd.

Li, Y., Baldwin, T., and Cohn, T. (2018). Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30, Melbourne, Australia. Association for Computational Linguistics.

Liebeck, M., Esau, K., and Conrad, S. (2016). What to do with an airport? mining arguments in the German online participation project tempelhofer feld. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 144–153, Berlin, Germany. Association for Computational Linguistics.

Lindeman, R. H., Merenda, P. F., and Gold, R. Z. (1980). *Introduction to Bivariate and Multivariate Analysis.* Scott Foresman, Glenview, IL, USA.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis* *lectures on human language technologies*, 5(1):1–167.

Liu, C., Guo, C., Dakota, D., Rajagopalan, S., Li, W., Kübler, S., and Yu, N. (2014). "my curiosity was satisfied, but not in a good way": Predicting user ratings for online recipes. In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, pages 12–21, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Liu, L., Shang, J., Ren, X., Xu, F. F., Gui, H., Peng, J., and Han, J. (2018). Empower sequence labeling with task-aware neural language model. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 5253–5260. AAAI Press.

Liu, Q., Kusner, M. J., and Blunsom, P. (2020). A survey on contextual embeddings. *arXiv preprint arXiv:2003.07278*.

Liu, Y., Li, P., and Hu, X. (2022). Combining context-relevant features with multi-stage attention network for short text classification. *Computer Speech & Language*, 71:101268.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lynn, V., Son, Y., Kulkarni, V., Balasubramanian, N., and Schwartz, H. A. (2017). Human centered NLP with user-factor adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1146–1155, Copenhagen, Denmark. Association for Computational Linguistics.

Majithia, S., Arslan, F., Lubal, S., Jimenez, D., Arora, P., Caraballo, J., and Li, C. (2019). ClaimPortal: Integrated monitoring, searching, checking, and analytics of factual claims on Twitter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 153–158, Florence, Italy. Association for Computational Linguistics.

Malmasi, S. and Zampieri, M. (2016). Maza at semeval-2016 task 11: Detecting lexical complexity using a decision stump meta-classifier. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 991–995.

Manzini, T., Chong, L. Y., Black, A. W., and Tsvetkov, Y. (2019). Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 615–621.

Masala, M., Iacob, R. C. A., Uban, A. S., Cidota, M., Velicu, H., Rebedea, T., and Popescu, M. (2021). jurBERT: A Romanian BERT model for legal judgement prediction. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 86–94, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matero, M., Idnani, A., Son, Y., Giorgi, S., Vu, H., Zamani, M., Limbachiya, P., Guntuku, S. C., and Schwartz, H. A. (2019). Suicide risk assessment with multi-level dual-context language and BERT. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 39–44, Minneapolis, Minnesota. Association for Computational Linguistics.

McDonald, R., Pereira, F., Ribarov, K., and Hajic, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In

Proceedings of human language technology conference and conference on empirical methods in natural language processing, pages 523–530.

McHardy, R., Adel, H., and Klinger, R. (2019). Adversarial training for satire detection: Controlling for confounding variables. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 660–665, Minneapolis, Minnesota. Association for Computational Linguistics.

McNamee, R. (2005). Regression modelling and other methods to control confounding. *Occupational and Environmental Medicine*, 62(7):500–506.

Mehrabi, N., Gowda, T., Morstatter, F., Peng, N., and Galstyan, A. (2019). Man is to person as woman is to location: Measuring gender bias in named entity recognition. *arXiv preprint arXiv:1910.10872*.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6).

Mekala, S., Bulusu, V. V., and Reddy, R. (2018). A survey on authorship attribution approaches. *Int. J. Comput. Eng. Res.(IJCER)*, 8(8).

Mendelberg, T., Karpowitz, C. F., and Oliphant, J. B. (2014). Gender inequality in deliberation: Unpacking the black box of interaction. *Perspectives on Politics*, 12(1):18–44.

Mendelsohn, J., Budak, C., and Jurgens, D. (2021). Modeling framing in immigration discourse on social media. In *Proceedings of* *the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2219–2263, Online. Association for Computational Linguistics.

Merrill, W., Goldberg, Y., Schwartz, R., and Smith, N. A. (2021). Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (Workshop Track)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119, Lake Tahoe, Nevada. Curran Associates Inc.

Milner, W. T., Poe, S. C., and Leblang, D. (1999). Security rights, subsistence rights, and liberties: A theoretical survey of the empirical landscape. *Hum. Rts. Q.*, 21:403.

Mirko, L., Sahitaj, P., Kallenberg, S., Coors, C., Dumani, L., Schenkel, R., and Bergmann, R. (2020). Towards an argument mining pipeline transforming texts to argument graphs. *Computational Models of Argument: Proceedings of COMMA 2020*, 326:263.

Mohammad, S., Bravo-Marquez, F., Salameh, M., and Kiritchenko, S. (2018). Semeval-2018 task 1: Affect in tweets. In *Proceedings of SEMEVAL*, pages 1–17, New Orleans, LA.

Mohammad, S., Zhu, X., and Martin, J. (2014). Semantic role labeling of emotions in tweets. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 32–41, Baltimore, Maryland. Association for Computational Linguistics.

Mohammad, S. M. and Bravo-Marquez, F. (2017). Emotion intensities in tweets. In *Proceedings of the sixth joint conference on lexical and computational semantics (*Sem)*, Vancouver, Canada.

Møller, J. and Skaaning, S.-E. (2013). Autocracies, democracies, and the violation of civil liberties. *Democratization*, 20(1):82–106.

Moradi, R., Berangi, R., and Minaei, B. (2020). A survey of regularization strategies for deep models. *Artificial Intelligence Review*, 53(6):3947–3986.

Morgan-Lopez, A. A., Kim, A. E., Chew, R. F., and Ruddle, P. (2017). Predicting age groups of twitter users based on language and metadata features. *PloS one*, 12(8):e0183537.

Naderi, N. and Hirst, G. (2017). Classifying frames at the sentence level in news articles. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 536–542, Varna, Bulgaria. INCOMA Ltd.

Naik, A. and Rangwala, H. (2015). A ranking-based approach for hierarchical classification. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10.

Newman, M. L., Groom, C. J., Handelman, L. D., and Pennebaker, J. W. (2008). Gender differences in language use: An analysis of 14,000 text samples. *Discourse processes*, 45(3):211–236.

Nguyen, N. and Guo, Y. (2007). Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, pages 681–688.

Nullmeier, F., Lhotta, R., Biegoń, D., Gronau, J., Haunss, S., Hurrelmann, A., Krell-Laluhová, Z., Lenke, F., Nonhoff, M., Pritzlaff, T., and et al. (2015). *Project B1: Legitimating States, International Regimes, and Economic Orders. Codebook – Final Version.*

Ostrowski, W., Arora, A., Atanasova, P., and Augenstein, I. (2021). Multi-hop fact checking of political claims. In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3892–3898. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Oztig, L. I. and Donduran, M. (2020). Failed coups, political survival, and civil liberties restrictions in nondemocratic regimes. *The Social Science Journal*, pages 1–15.

Padó, S., Blessing, A., Blokker, N., Dayanik, E., Haunss, S., and Kuhn, J. (2019). Who sides with whom? towards computational construction of discourse networks for political debates. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2841–2847, Florence, Italy. Association for Computational Linguistics.

Padrão, P., Lunet, N., Santos, A. C., and Barros, H. (2007). Smoking, alcohol, and dietary choices: evidence from the portuguese national health survey. *BMC Public Health*, 7(1):1–9.

Papakyriakopoulos, O. (2020). *Political machines: machine learning for understanding the politics of social machines.* PhD thesis, Technische Universität München.

Papay, S., Klinger, R., and Padó, S. (2020a). Dissecting span identification tasks with performance prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pages 4881–4895, Online. Association for Computational Linguistics.

Papay, S., Klinger, R., and Padó, S. (2020b). Dissecting span identification tasks with performance prediction. In *Proceedings of EMNLP*, page 4881–4895, Online.

Parapar, J., Martín-Rodilla, P., Losada, D. E., and Crestani, F. (2021). Overview of erisk at clef 2021: Early risk prediction on the internet (extended overview). *CLEF (Working Notes)*, pages 864–887.

Pardo, F. M. R., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., and Stein, B. (2016). Overview of the 4th author profiling task at PAN 2016: Cross-genre evaluations. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016*, volume 1609 of *CEUR Workshop Proceedings*, pages 750–784. CEUR-WS.org.

Peldszus, A. and Stede, M. (2013). From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.

Peng, H., Mou, L., Li, G., Chen, Y., Lu, Y., and Jin, Z. (2015). A comparative study on regularization strategies for embedding-based neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2106–2111, Lisbon, Portugal. Association for Computational Linguistics.

Pennacchiotti, M. and Popescu, A.-M. (2011). Democrats, republicans and starbucks afficionados: user classification in twitter. In *Proceedings of the 17th ACM SIGKDD international conference* *on Knowledge discovery and data mining*, pages 430–438.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Plank, B. and Hovy, D. (2015). Personality traits on Twitter—or—How to get 1,500 personality tests in a week. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–98, Lisboa, Portugal. Association for Computational Linguistics.

Plank, B., Hovy, D., and Søgaard, A. (2014). Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland. Association for Computational Linguistics.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Durme, B. V. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Punyakanok, V., Roth, D., Yih, W.-t., and Zimak, D. (2004). Semantic role labeling via integer linear programming inference. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1346–1352, Geneva, Switzerland. COLING.

Qian, Y., Muaz, U., Zhang, B., and Hyun, J. W. (2019). Reducing gender bias in word-level language models with a gender-equalizing loss function. *arXiv preprint arXiv:1905.12801*.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Rabinovich, E., Patel, R. N., Mirkin, S., Specia, L., and Wintner, S. (2017). Personalized machine translation: Preserving original author traits. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1074–1084, Valencia, Spain. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Ramshaw, L. A. and Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Riedel, S. and Clarke, J. (2006). Incremental integer linear programming for non-projective dependency parsing. In *Proceedings* *of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 129–137, Sydney, Australia. Association for Computational Linguistics.

Rinscheid, A., Eberlein, B., Emmenegger, P., and Schneider, V. (2020). Why do junctures become critical? political discourse, agency, and joint belief shifts in comparative perspective. *Regulation & Governance*, 14(4):653–673.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Roller, S. and Erk, K. (2016). PIC a different word: A simple model for lexical substitution in context. In *Proceedings of NAACL/HLT*, pages 1121–1126, San Diego, California.

Roychowdhury, S., Diligenti, M., and Gori, M. (2021). Regularizing deep networks with prior knowledge: A constraint-based approach. *Knowledge-Based Systems*, 222:106989.

Rozado, D. (2020). Wide range screening of algorithmic bias in word embedding models using large sentiment lexicons reveals underreported bias types. *PloS one*, 15(4):e0231189.

Rubin, D. B. (1973). Matching to remove bias in observational studies. *Biometrics*, pages 159–183.

Rudinger, R., Naradowsky, J., Leonard, B., and Van Durme, B. (2018). Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 8–14, New Orleans, LA.

Sabatier, P. A. and Jenkins-Smith, H. C. (1993). *Policy change and learning: An advocacy coalition approach.* Westview press.

Saldaña, J. (2009). The coding manual for qualitative researchers. *Lontoo: SAGE Publications Ltd*, 3.

Salmerón, R., García, C. B., and García, J. (2018). Variance inflation factor and condition number in multiple linear regression. *Journal of Statistical Computation and Simulation*, 88(12):2365–2384.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sap, M., Card, D., Gabriel, S., Choi, Y., and Smith, N. A. (2019). The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Saunders, D. and Byrne, B. (2020). Reducing gender bias in neural machine translation as a domain adaptation problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736, Online. Association for Computational Linguistics.

Schiller, B., Daxenberger, J., and Gurevych, I. (2020). Aspect-controlled neural argument generation. *arXiv preprint arXiv:2005.00084*.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Schrijver, A. (1984). *Linear and Integer Programming.* John Wiley & Sons, New York.

Schroeder, J., Xu, J., and Chen, H. (2003). Crimelink explorer: Using domain knowledge to facilitate automated crime association analysis. In *International Conference on Intelligence and Security Informatics*, pages 168–180. Springer.

Schuster, M. and Nakajima, K. (2012). Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Schwemmer, C. and Jungkunz, S. (2019). Whose ideas are worth spreading? the representation of women and ethnic groups in ted talks. *Political Research Exchange*, 1(1):1–23.

Sezerer, E., Polatbilek, O., Sevgili, Ö., and Tekir, S. (2018). Gender prediction from tweets with convolutional neural networks: Notebook for pan at clef 2018. In *19th Working Notes of CLEF Conference and Labs of the Evaluation Forum, CLEF 2018*. CEUR Workshop Proceedings.

Shardlow, M., Gerber, L., and Nawaz, R. (2022). One emoji, many meanings: A corpus for the prediction and disambiguation of emoji sense. *Expert Systems with Applications*, 198:116862.

Shimaoka, S., Stenetorp, P., Inui, K., and Riedel, S. (2017). Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain. Association for Computational Linguistics.

Slapin, J. B. and Proksch, S.-O. (2008). A scaling model for estimating time-series party positions from texts. *American Journal of Political Science*, 52(3):705–722.

Smelser, N. J., Baltes, P. B., et al. (2001). *International encyclopedia* *of the social & behavioral sciences*, volume 11. Elsevier Amsterdam.

Smith, T. J. and McKenna, C. M. (2013). A comparison of logistic regression pseudo r2 indices. *Multiple Linear Regression Viewpoints*, 39(2):17–26.

Snijders, T. and Bosker, R. (2012). *Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling.* Sage Publishers, London, 2nd edition.

Snow, R., O'connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263.

Sobkowicz, P., Kaschesky, M., and Bouchard, G. (2012). Opinion mining in social media: Modeling, simulating, and forecasting political opinions in the web. *Government Information Quarterly*, 29(4):470–479. Social Media in Government - Selections from the 12th Annual International Conference on Digital Government Research (dg.o2011).

Song, Y., Lee, C.-J., and Xia, F. (2017). Learning word representations with regularization from prior knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152, Vancouver, Canada. Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Stanovsky, G., Smith, N. A., and Zettlemoyer, L. (2019). Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Stretcu, O., Viswanathan, K., Movshovitz-Attias, D., Platanios, E., Ravi, S., and Tomkins, A. (2019). Graph agreement models for semi-supervised learning. *Advances in Neural Information Processing Systems*, 32.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1.

Subramanian, S. (2020). *Natural Language Processing for Improving Transparency in Representative Democracy*. PhD thesis, UNIVERSITY OF MELBOURNE.

Subramanian, S., Cohn, T., and Baldwin, T. (2018). Hierarchical structured model for fine-to-coarse manifesto text analysis. 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 1964–1974, New Orleans, Louisiana. Association for Computational Linguistics.

Sullivan, G. M. and Feinn, R. (2012). Using effect size – or why the p value is not enough. *Journal of graduate medical education*, 4(3):279–82.

Swinger, N., De-Arteaga, M., Heffernan IV, N. T., Leiserson, M. D., and Kalai, A. T. (2019). What are the biases in my word embedding? In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 305–311.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Temple, L., Grasso, M. T., Buraczynska, B., Karampampas, S., and English, P. (2016). Neoliberal narrative in times of economic crisis: A political claims analysis of the uk press, 2007-14. *Politics & Policy*, 44(3):553–576.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Terechshenko, Z., Linder, F., Padmakumar, V., Liu, M., Nagler, J., Tucker, J. A., and Bonneau, R. (2020). A comparison of methods in political science text classification: Transfer learning language models for politics. *Available at SSRN 3724644*.

Thirukovalluru, R., Monath, N., Shridhar, K., Zaheer, M., Sachan, M., and McCallum, A. (2021). Scaling within document coreference to long texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3921–3931, Online. Association for Computational Linguistics.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Tjønneland, A., Grønbæk, M., Stripp, C., and Overvad, K. (1999).

Wine intake and diet in a random sample of 48763 danish men and women. *The American journal of clinical nutrition*, 69(1):49–54.

Toulmin, S. E. (2003). *The uses of argument.* Cambridge university press.

Tu, L., Lalwani, G., Gella, S., and He, H. (2020). An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Tuggener, D., von Däniken, P., Peetz, T., and Cieliebak, M. (2020). LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1235–1241, Marseille, France. European Language Resources Association.

van de Camp, M. (2016). *A link to the past: Constructing historical social networks from unstructured data.* PhD thesis, Tilburg University. Series: TiCC Ph.D. Series Volume: 44.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Verhoeven, B., Daelemans, W., and Plank, B. (2016). TwiSty: A multilingual twitter stylometry corpus for gender and personality profiling. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1632–1637, Portorož, Slovenia. European Language Resources Association (ELRA).

Volkens, A., Burst, T., Krause, W., Lehmann, P., Matthieß, T., Merz, N., Regel, S., Weßels, B., and Zehnter, L. (2020). *The*

Manifesto Project Dataset - Codebook.

Volkens, A., Krause, W., Lehmann, P., Matthieß, T., Merz, N., Regel, S., and Weßels, B. (2019). The Manifesto data collection, version 2019b.

Volkova, S., Wilson, T., and Yarowsky, D. (2013). Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1815–1827, Seattle, Washington, USA. Association for Computational Linguistics.

Wang, B. (2018). Disconnected recurrent neural networks for text categorization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2311–2320.

Wang, J. and Lu, W. (2020). Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Wang, M. and Zhou, X. (2018). Yuan at SemEval-2018 Task 1: Tweets emotion intensity prediction using ensemble recurrent neural network. In *Proceedings of SEMEVAL*, pages 205–209, New Orleans, LA.

Wang, T., Yang, D., and Wang, X. (2021a). Identifying and mitigating spurious correlations for improving robustness in nlp models. *arXiv preprint arXiv:2110.07736*.

Wang, X., Xiong, Y., Qian, X., Wei, Y., Li, L., and Wang, M. (2021b). Lightseq2: Accelerated training for transformer-based models on gpus. *arXiv preprint arXiv:2110.05722*.

Wang, Z., Ng, P., Ma, X., Nallapati, R., and Xiang, B. (2019). Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Webster, K., Recasens, M., Axelrod, V., and Baldridge, J. (2018). Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Wu, C., Wu, F., Liu, J., Yuan, Z., Wu, S., and Huang, Y. (2018). THU_NGN at SemEval-2018 Task 1: Fine-grained tweet sentiment intensity analysis with attention CNN-LSTM. In *Proceedings of SEMEVAL*, pages 186–192, New Orleans, Louisiana.

Wu, S. and Dredze, M. (2020). Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Wu, X., Tao, Z., Jiang, B., Wu, T., Wang, X., and Chen, H. (2022). Domain knowledge-enhanced variable selection for biomedical data analysis. *Information Sciences*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xia, L., Luo, D., Zhang, C., and Wu, Z. (2019). A survey of topic models in text classification. In *2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 244–250. IEEE.

Yadav, V. and Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yan, D. and Guo, S. (2019). Leveraging contextual sentences for text classification by using a neural attention model. *Computational intelligence and neuroscience*, 2019.

Yan, H., Deng, B., Li, X., and Qiu, X. (2019). Tener: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*.

Yang, B. and Cardie, C. (2013). Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria. Association for Computational Linguistics.

Yang, L., Yu, J., Zhang, C., and Na, J.-C. (2021). Fine-grained sentiment analysis of political tweets with entity-aware multimodal network. In *International Conference on Information*, pages 411–420. Springer.

Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., and Lin, J. (2019). End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Yang, Z., Huang, Y., Jiang, Y., Sun, Y., Zhang, Y.-J., and Luo, P. (2018). Clinical assistant diagnosis for electronic medical record based on convolutional neural network. *Scientific reports*, 8(1):1–9.

Yoo, W., Mayberry, R., Bae, S., Singh, K., He, Q. P., and Lillard Jr, J. W. (2014). A study of effects of multicollinearity in the multivariable analysis. *International journal of applied science and technology*, 4(5):9.

Yuret, D., Han, A., and Turgut, Z. (2010). SemEval-2010 task 12: Parser evaluation using textual entailments. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 51–56, Uppsala, Sweden. Association for Computational Linguistics.

Zamponi, L. and Bosi, L. (2016). Which crisis? european crisis and national contexts in public discourse. *Politics & Policy*, 44(3):400–426.

Zhang, B. H., Lemoine, B., and Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017).

Understanding deep learning requires rethinking generalization (2016). *arXiv preprint arXiv:1611.03530.*

Zhang, H., Song, Y., and Song, Y. (2019). Incorporating context and external knowledge for pronoun coreference resolution. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 872–881, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhang, M., Zhang, Y., and Fu, G. (2016). Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhao, J., Mukherjee, S., Hosseini, S., Chang, K.-W., and Hassan Awadallah, A. (2020). Gender bias in multilingual embeddings and cross-lingual transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2896–2907, Online. Association for Computational Linguistics.

Zhao, J., Wang, T., Yatskar, M., Cotterell, R., Ordonez, V., and Chang, K.-W. (2019). Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 629–634, Minneapolis, Minnesota.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2017). Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the*

2017 Conference on Empirical Methods in Natural Language Processing, pages 2979–2989, Copenhagen, Denmark. Association for Computational Linguistics.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2018a). Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 15–20, New Orleans, LA.

Zhao, J., Zhou, Y., Li, Z., Wang, W., and Chang, K.-W. (2018b). Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496*.

Zhou, D., Wang, J., Jiang, B., Guo, H., and Li, Y. (2018). Multi-task multi-view learning based on cooperative multi-objective optimization. *IEEE Access*, 6:19465–19477.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015a). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Zhu, Z., Li, Z., Wylde, D., Failor, M., and Hrischenko, G. (2015b). Logistic regression for insured mortality experience studies. *North American Actuarial Journal*, 19(4):241–255.

Zmigrod, R., Mielke, S. J., Wallach, H., and Cotterell, R. (2019). Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

Zürn, M. (2014). The politicization of world politics and its effects: Eight propositions. *European political science review*, 6(1):47–71.

A DebateNet Dataset - Codebook

In this chapter, we provide a shortened version of the DebateNet dataset codebook in English that contains definitions, guidelines and fine-grained categories. Besides these, the full codebook also contains a commentary section providing examples to the annotators. The full codebook can be found at `https://clarin09.ims.uni-stuttgart.de/debatenet/MARDY_Codebook_Mig_english.pdf`

A DebateNet Dataset - Codebook

Codebook Migration

Annotation manual of the MARDY research project on the refugee and migration discourse in Germany

Sebastian Haunss[1], **Jonas Kuhn**[2], **Sebastian Padó**[2], **André Blessing**[2], **Gabriella Lapesa**[2], **Nico Blokker**[1], and **Erenay Dayanik**[2]

[1]SOCIUM, University of Bremen, Germany
[2]IMS, University of Stuttgart, Germany
{blokker, sebastian.haunss}@uni-bremen.de
{andre.blessing, erenay.dayanik, jonas.kuhn, gabriella.lapesa, sebastian.pado}@ims.uni-stuttgart.de

23. November 2020

1 Definitions

1.1 Claim

A *claim* is an instance of strategic action in the public and consists of an expression of a political opinion by any form of physical or verbal action (verbal expression, explanation, descision, demonstration, court decision, etc.), independently of the role of the actor (governments, social movements, NGOs, individuals, anonymous actors, etc.).

1.2 Definition Frame

Sometimes claims are justified by a so-called *frame*: in our project, a frame is defined as any justification given for a claim. Therefore, a frame cannot stand alone but is always to be annotated with a claim.

1.3 General Examples

Claim: *Horst Seehofer demands an upper limit.*

Frame: Horst Seehofer demands an upper limit, *in order to reduce the numbers of refugees.*

no claim (vague): Horst Seehofer could imagine an upper limit for refugees under certain conditions.

no claim (assessment): Human rights are important.

2 Annotation guidelines

A claim consists of 4 attributes:

- actor
- claim-category
- polarity (valence)
- date (default is the day before)

Unit of annotation is always the entire sentence.

Example 1: Angela Merkel argued last sunday against an upper limit.

A sentence can contain multiple claims and actors.

Example 2: Merkel reaffirmed her rejection of the upper limit demanded by Seehofer and pleaded instead for a solution based on solidarity within the EU.

Also possible are combination of claim-categories to capture more complex demands:

Example 3: In a repatriation decree, Boris Pistorius has stipulated, among other things, that young people who are one year away from completing their training will not be deported.

Albeit, individual claim-categories are to be preferred. The same applies to the priority of specific over general claim-categories.

Special categories are 900 [irrelevant] and 999 [flag for re-evaluation].

A DebateNet Dataset - Codebook

3 Claims

Table 1: Claim Categories

100 *Controlling Migration*	200 *Residency*	300 *Integration*	400 *Domestic security*
101 controlled migration	201 Emergency accommodation/first admission	301 integration offers	401 violence against migrants
102 ceiling/upper limit	202 refugee accommodation	302 language courses	402 refugee protection
104 isolation/immigration stop	203 centralised accommodation	303 forced integration	403 civil protection
105 border controls	204 creation of living space	304 mutual integration	404 refugee crime
106 border defence	205 forced occupancy	305 integration contracts	405 counterterrorism measures
107 fence	206 private accommodation	306 Diversity through immigration	406 ban mile
108 immigration law	207 deportations	307 family reunion	407 human trafficking
109 Fight against people-smugglers	209 residence obligation	308 integration centers	408 deprivation of liberty
110 asylum law	210 subsidiary protection	309 Care (medical, financial, ...)	
111 sea rescue	211 right of abode	310 Cost sharing for refugees	
112 differentiation by group	212 contributions in kind	312 cultural awareness	
113 visa liberalisation	213 church asylum	313 foreign fincancing of schools/curches	
114 (Canadian) points system	214 naturalization	314 access to educational services	
115 resettlement program	215 transit areas	315 access to social benefits	
190 Current migration policy	216 dual citizenship		
	217 municipal voting rights		
	218 voluntary return		
199 General	299 General	399 General	499 General

500 *Foreign Policy*	600 *Economy + Labour Market*	700 *Society*	800 *Procedures*
501 EU solution (quotas for refugees)	601 Labour market integration	701 Populism + actionism	801 Rule of law
502 international solution	602 combating shortage of skilled labour	702 human rights	802 federal responsibility
503 combating causes of flight	603 easier/faster access	703 xenophobia	803 equitable load distribution
504 safe country of origin	604 "guest workers"	704 society overstrained	804 staff increase
505 Asylum procedure in countries of origin	605 Minimum wage for refugees	705 refugees welcome	805 additional financing
507 Cooperation with transit countries	606 Refugees as cheap labourers	706 Recognition of fundamental rights	806 case-by-case assessment
508 military intervention	607 refugee activation	707 Separation of migration/refugee term	807 Reducing bureaucracy
509 Dublin regulation	608 Recording educational attainment	708 societal mobilization	808 Process optimization (cooperation)
	609 Taxes	709 right-wing radicalism	809 enforceability of laws
	610 cost-benefit analysis	710 left-wing radicalism	810 planning reliability
	611 migrant quota	711 Islam	811 criminal prosecution of xenophobia
		712 public debate	812 Fast / accelerated procedure
		713 Christian values	813 Transparent procedures
		714 "Leitkultur"	814 Protection of minors
		715 open society	815 Protection of women
		716 headscarf ban	816 taking concerns seriously
		717 Islamism	817 priority check
		790 scientific findings	818 protection from right-wing violence
			819 privatize prosecution of xenophobia
599 General	699 General	799 General	899 General

B Dataset Details

In this chapter, we provide threshold values used for determining frequency bands and subcategory-frequency band assignments in Chapter 8. In the first experiment of Chapter 8, we split the fine-grained categories in DebateNet into three equal-sized frequency bands using following threshold values: high-frequency (265$\geq$f$\geq$ 67), mid-frequency (65$\geq$f$\geq$ 40) and low-frequency (20$\geq$f$\geq$ 39). Table 42 shows category frequency band assignments.

In the second experiment which is performed in Manifesto corpus, we again split the categories into three equal-sized frequency bands. Table 43 shows threshold values for each language and frequency band in the Manifesto dataset and we publish category-frequency band assignments at `https://github.com/repo4supp/data_splits`.

Frequency Band	Label					
LOW	111	199	201	209	213	214
	406	408	499	502	505	508
	602	603	605	701	706	707
	708	801	802	807	811	814
MID	106	107	109	204	211	212
	215	301	302	303	307	401
	402	405	503	509	601	699
	702	711	715	803	804	808
HIGH	101	102	104	105	108	110
	190	202	203	207	299	309
	399	501	504	507	703	705
	709	712	799	805	812	899

Table 42: Frequency band assigments for the subcategories in DebateNet.

Lang	Freq.	25% Threshold	50% Threshold	100% Threshold
Fi	Low	1 ≥f≥ 12	1 ≥f≥ 23	2≥f≥ 52
	Mid	14≥f≥ 55	24≥f≥ 110	53≥f≥ 215
	High	57≥f≥ 417	111≥f≥ 867	221≥f≥ 1666
De	Low	3 ≥f≥ 56	5 ≥f≥ 98	6≥f≥ 201
	Mid	59≥f≥ 196	99≥f≥ 391	202≥f≥ 764
	High	204≥f≥ 951	401 ≥f≥ 1866	785 ≥f≥ 3655
Hu	Low	1 ≥f≥ 31	1 ≥f≥ 63	2≥f≥ 124
	Mid	37≥f≥ 147	69≥f≥ 276	133 ≥f≥ 560
	High	168 ≥f≥ 772	357 ≥f≥ 1541	697 ≥f≥ 3046
Tr	Low	1 ≥f≥ 33	1 ≥f≥ 67	1≥f≥ 130
	Mid	34≥f≥ 166	68≥f≥ 316	137≥f≥ 628
	High	187 ≥f≥ 937	380 ≥f≥ 1862	739 ≥f≥ 3720
En	Low	2 ≥f≥ 22	4 ≥f≥ 42	4≥f≥ 91
	Mid	23≥f≥ 84	49≥f≥ 180	97≥f≥ 356
	High	101 ≥f≥ 536	188 ≥f≥ 1122	368≥f≥ 2315

Table 43: Frequency band threshold values used in Manifesto corpus.