Learning Dynamic User Representations and Exploiting Personalization in NLP

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Joan Plepi
aus
Durres, Albanien

Bonn, 03. 2024

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachterin / Betreuerin:	Prof. Dr. Lucie Flek
Gutachter:	Prof. Dr. Paolo Rosso
Tag der Promotion:	25.07.2024
Erscheinungsjahr:	2024

Abstract

Nowadays, with the improvement of hardware and computations capabilities, there has been an increase of language models size and language understanding capabilities. However, these Natural Language Processing (NLP) models often treat language as universally understood, disregarding the socio- and psycholinguistic insights about the impact of speaker and audience characteristics on communication. In the dynamic environment of social media, users not only create connections but also tailor their language to the context, their audience and their affiliation to different sociodemographic group. Hence, in the evolving landscape of language technologies, there is an increasing demand for personalized systems that can mirror a user's individual style. Recognizing this, our thesis posits that a deeper understanding of users' social and semantic networks is essential to enhancing language understanding.

Central to this thesis is the development of methodologies for capturing user's context, and integrating this context into NLP models in order to enhance their performance. Furthermore, there is a diversity of perspectives within different user groups concerning various subjective topics or situations. By integrating user-specific information, our models seek to better interpret how an individual perceive an utterance, and improve the performance of text classification in subjective NLP tasks, where there is a variety of perspectives. Additionally, users interactions with a community, are influenced by the evolving topics of interest and the prevailing views within their groups. This evolving landscape posits the need for dynamic user representations, that can capture evolving aspects of user behavior and social interactions. By leveraging social and semantic graphs, we construct models that effectively encapsulate the changing nature of user behavior over time. Modeling these temporal patterns of users' interactions, can provide insights into how their opinions or behaviors change, in addition to predicting future behaviors. In general, these approaches aim to significantly augment text representation in NLP.

Finally, we propose an evaluation framework across diverse tasks, such as sarcasm detection, misinformation spreading, perspective classification, and personalized language generation, to showcase the effectiveness and versatility of our approaches. Overall, our research makes a significant leap in integrating user-specific information into NLP models for a variety of tasks, paving the way for more nuanced and context-aware language processing.

Acknowledgements

This work would not have been possible without the support and guidance of many individuals. First and foremost, I would like to thank my supervisor Prof. Dr. Lucie Flek, for giving me the opportunity to pursue this research work, and for her constant support, and feedback throughout my PhD studies. Her insightful guidance, helped me to grow both professionally and personally throughout these years. I would also like to thank Prof. Dr. Paolo Rosso, for taking the time to review this thesis, and for the inspiring and interesting discussions we had in the past. I want to also recognize the support that this work has from the German Federal Ministry of Education and Research (BMBF) as a part of the Junior AI Scientists program under the reference 01-S20060.

I want to extend my sincere gratitude to my colleagues with whom I worked closely, Dr. Flora Sakketou and Dr. Charles Welch, for co-authoring a good amount of research papers and providing constructive discussions, and feedback to further improve my research work. I am grateful to all my colleagues at the Conversational AI and Social Analytics (CAISA) lab for our valuable discussions during lunch breaks, ideas exchange, and overall great experience at the lab. Thank you Vahid Sadiri Javadi, Akbar Karimi, Allison Lahnala, Shaina Ashraf, and Fabian Lechner. Additionally, I would like to thank my research assistants and collaborators who contributed in various stages of my research, Henri-Jacques Geiss, Riccardo Cervero, Béla Neuendorf, Jakob Leander Müller, Magdalena Buski, and all the bachelor and master students whom I was pleased to support.

I would also want to thank my friend and colleague Endri Kacupaj for valuable discussions and feedback throughout my research work. Additionally, I want to thank my close friends Mariel and Olsi for keeping me focused and motivated throughout my journey.

Last but not least, I want to thank my family for their encouragement throughout my study. I want to thank my partner, Lueda, for her support during the days and evenings of writing this thesis and for believing in me. Many thanks to my brother and future colleague, Gjergj, for his support, discussions, and feedback on the thesis, and a special thanks to my parents, for continuously supporting and encouraging me throughout my journey.

This thesis is dedicated to my parents, Kozma and Enkelana.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The evolution of Natural Language Processing (NLP) systems has been marked by significant advancements, moving from early statistical methods like term frequency-inverse document frequency (tf-idf) and bag of words (BoW) to more sophisticated techniques. Initially, these methods relied solely on word frequency counts in corpora, disregarding the word order and the surrounding context, leading to semantic information loss. With the boom of deep learning, Mikolov et al. [1, 2] introduced word2vec, a family of unsupervised pre-trained embeddings capable of capturing semantic relationships between words. Despite its breakthroughs, word2vec had its limitations, notably in handling out-of-vocabulary words as well as capturing different meanings of a word based on context (polysemy). These challenges were later addressed by deep contextualized word embedding such as ELMo [3], BERT [4], OpenAI GPT [5].

Despite the recent improvements in word representations, the majority of NLP models treat language as universally understood, assuming that the context is the same for everyone. Social and psycholinguistic research show that the communicated message is influenced by the individual characteristics of the speaker, as well as by their affiliations to different sociodemographic groups [6, 7, 8]. For instance, the example shown in Figure 1.1, ”Going to the festival tonight, will be fun meeting so many new people”, can be interpreted either as sarcastic or non-sarcastic. Without additional context, such as knowing the speaker’s introverted tendencies from past posts, the sarcastic intent might be missed.

In general, social media contains a lot of user-generated content that facilitates the exploration of users. On platforms like Twitter (now X) or Reddit, users tend to share posts that express their views or their individual characteristics. These platforms facilitate interactions within social networks, where users interact and influence each other to varying degrees. These dynamic interactions are often shaped by factors like the type of social network, geographical region, or individual traits, such as the influence exerted by celebrities [9, 10, 11, 12]. Yet, similar to word representations, user representations face various challenges. For instance, users often belong to multiple ’contexts’ or subgroups, each reflecting different facets of their identity and opinions. For example, as illustrated in Figure 1.2, a user’s participation in various subreddits might reveal their interests and ideological leanings, such as being a vegan, politically left-leaning, and a basketball enthusiast. Such diversity in user interactions and affiliations can profoundly affect their viewpoints, like ”Who shall be the next US

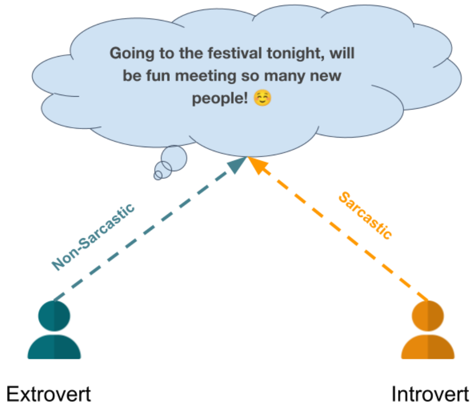


Figure 1.1: An example showing ambiguity of the text without having any additional context about the author.

president?". To this end, one can infer the views based on: a) the user's traits which can be extracted by their posting history (maybe the user is sharing news from left or right-wing biased sources based on their partisanship), and b) interaction or affiliation with a social network, which may be influenced by having a similar political bias. Recognizing the importance of these dynamic user contexts, this thesis focuses on developing models that not only integrate text and user context but also adapt existing models to incorporate pre-trained user information. This approach is crucial in subjective NLP tasks, where understanding the user's context provides a more nuanced interpretation of language.

While individual traits of users on social media tend to be more general and static, their interactions within social networks are dynamic, and deeply influenced by the evolving topics of interest and the prevailing views within their groups. Each interaction or post by a user can be seen as a snapshot, capturing their stance or opinion at a particular moment in time. Interests, opinions, and even the general nature of users can shift over time, influenced by various factors like global events, personal experiences, or changes in social group dynamics. As a result, static user features, which may have been accurate at one point, can quickly become outdated, failing to accurately represent the current state or views of the user. This evolving landscape of social media interactions underscores the necessity of a dynamic approach to model user context. Static models simply cannot keep pace with the rapid changes in user behavior and opinions. To this end, our research proposes the development of a dynamic framework, one that is capable of not just capturing but also predicting the future behavior of users based on their past interactions.

1.2 Problem Statement and Challenges

Nowadays, there is an emerging necessity for NLP systems that are not just accurate but also personalized, reflecting the linguistic nuances and preferences unique to each user. This paradigm shift emphasizes the importance of incorporating the nuanced characteristics of language speakers or writers into NLP models. The main goal of this thesis is to investigate the utilization of user context across various NLP tasks. In order to achieve this goal, we need to consider the following. When does the user context help? Are there any characteristics of the task or the domain of the text where the user context helps more? Moreover, the discourse prevalent in platforms like social media is not

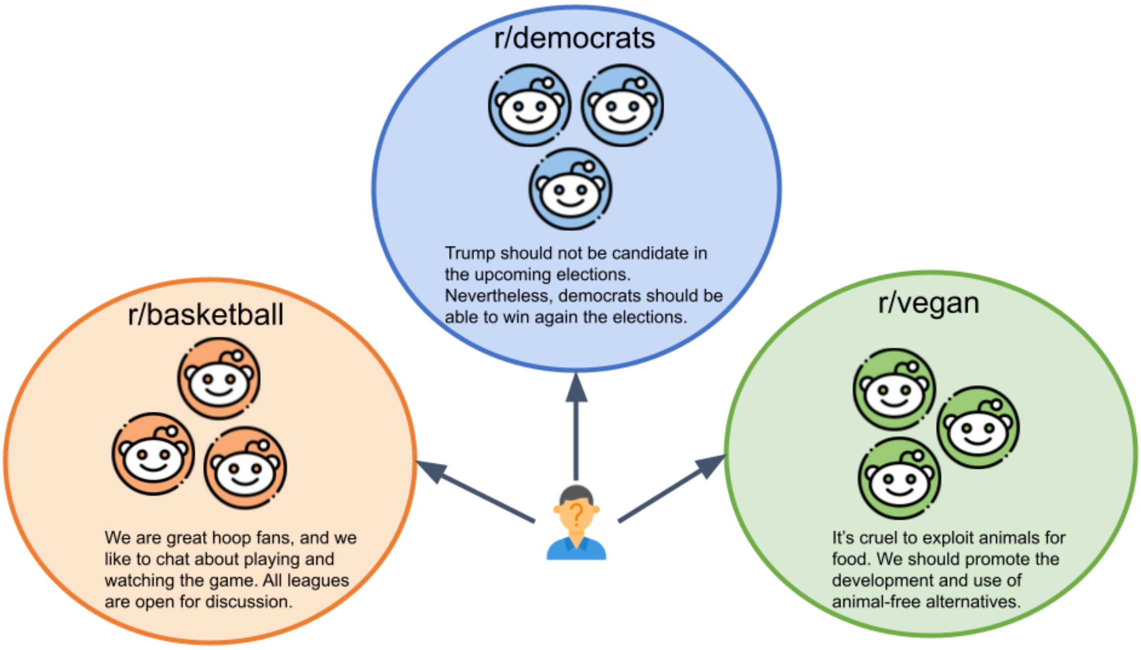


Figure 1.2: A user can interact with multiple communities during their posting history. From these interactions, we can capture the user traits like political preferences, and hobbies in order to understand their interests and behavior for better tailoring NLP systems to their preferences. Additionally, we can model their writing style depending on the community which they are interacting.

dependent only on the individual traits, but also on the audience they are addressing. This raises the following question: What if we augment NLP systems with the audience context in addition to the speaker? However, integrating user context in existing models, increases computational times, due to additional parameters. Therefore, it is important to utilize methods that do not complicate the underlying architectures. This raises the critical question of how we can represent users, in a large-scale context, and how can we filter the most helpful context.

In this section, we define the main problem statement, for this dissertation.

> **Problem Statement**
>
> How can dynamic user context be computed and integrated into Natural Language Processing (NLP) models to enhance their performance across various tasks?

We identify several fundamental challenges to be tackled while working towards our research problem, related to the data, modeling, and evaluation (Figure 1.3).

Challenge 1: Extracting User Data The process of augmenting textual data with user-specific historical comments or posts presents its own set of challenges. First, the task of crawling and gathering such data is inherently time-consuming, primarily due to the large volume of information available on social media platforms. Second, the issue of data incompleteness often arises, typically caused by

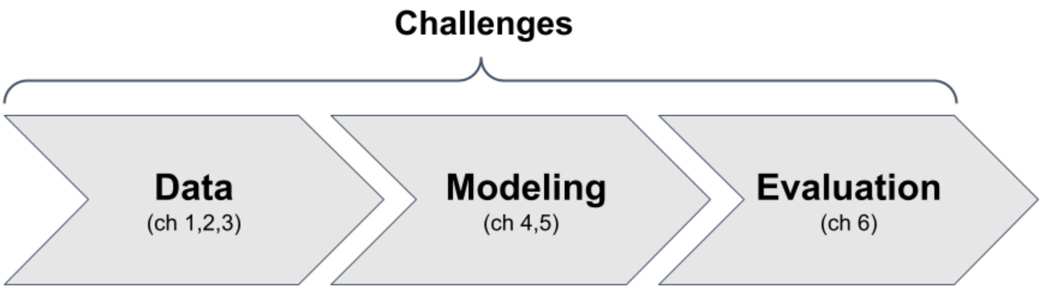


Figure 1.3: Pipeline of the challenges related to our research problem.

users deleting their accounts or removing their content. This results in missing historical data, which can create gaps in the user profile and lead to incomplete or skewed analyses. Additionally, even when historical data is available, it presents other challenges such as relevance and quality. Users generate vast amounts of text over time, but not all of this content is relevant to the specific NLP task. Furthermore, the dynamic nature of user behavior on social media means that historical data may not always be representative of a user's current state. Over time, changes in opinions, interests, or writing style can introduce noise into the dataset, potentially negatively impacting the performance of current models. Consequently, these challenges necessitate the exploration of methods to correctly filter and retrieve the data from the users.

Challenge 2: Social Media Data In our research, as highlighted in the preceding section, social media data serves as the primary source for exploring human interactions. However, leveraging this data presents several challenges. A primary issue is the sparsity and often fragmented nature of social media communications. Comments or messages on these platforms frequently consist of only a few tokens and may not constitute meaningful sentences. This brevity can lead to a significant lack of context. Understanding a specific text often necessitates reading through entire conversation threads or delving into a user's past comments. This requirement poses a challenge in data processing and interpretation, as it demands comprehensive analysis beyond the immediate content of individual posts.

Challenge 3: Data Imbalance A major issue encountered when collecting social media data for various NLP tasks is data imbalance. This challenge is particularly evident in specialized tasks such as sarcasm detection or misinformation spreading. For instance, in a dataset of 100,000 tweets gathered for sarcasm analysis, only a small fraction may exhibit sarcastic content. Similar patterns of imbalance are observed in datasets for hate speech detection and other specific NLP tasks. This imbalance poses a significant challenge, as it can skew model training and affect the accuracy of predictions. Therefore, addressing data imbalance is crucial, both in the data processing phase and during model development. Furthermore, data imbalance is not limited to the content; it also extends to user-level data. The level of activity among social media users varies widely - some users are highly active, contributing large volumes of posts rich in information, relationships, and interactions, while others are much less active. This disparity leads to an uneven distribution of data across users. Consequently, it is essential to develop models capable of effectively handling this variability.

Challenge 4: Integrating User Context Integrating user context into NLP models is influenced by several key factors. Firstly, the relevance and utility of user context vary significantly depending on the specific NLP task at hand. In certain tasks, user information can provide valuable insights and enhance model performance, whereas in others, it may introduce noise and reduce accuracy. Secondly, the domain from which user information is derived plays a crucial role. Different domains yield varying amounts and types of user data, influencing the depth and quality of the context that can be integrated. Finally, the challenge of integrating user context is compounded by the diversity in the architectural design of NLP models. Different architectures may necessitate distinct methods for incorporating user context. Therefore, in this thesis, we focus on exploring a range of methodologies for integrating user context into various NLP tasks and model architectures.

Challenge 5: Modeling Temporal Dynamics Yet another significant challenge is to model the temporal dynamics of user context. The foremost difficulty lies in acquiring data that effectively captures the temporal evolution of user behavior. A key aspect of this challenge is the diversity in behavioral patterns exhibited by similar users. While some users may display relatively consistent behavior over time, others demonstrate more volatile patterns, with their interactions and posts fluctuating significantly. Moreover, explaining these fluctuations in user behavior can be particularly challenging, especially in the absence of direct correlations with global events or trends, such as political elections or the Covid-19 pandemic. In addition to these challenges, there's also the task of integrating these temporal fluctuations with the user's interaction history. This requires models that can not only capture temporal dynamics but also how these dynamics interact with the user's social environment.

Challenge 6: Evaluation of Personalized Generation The evaluation of personalized generation models, particularly in terms of maintaining persona consistency, represents another critical challenge in our research. Traditional evaluation frameworks primarily focus on determining whether the generated text aligns with a predefined set of persona sentences. However, this approach has inherent limitations, especially when considering the complexity of human personas and the nuances of language. One of the primary difficulties arises from the human evaluators' task of judging the consistency of a single piece of generated text with a broader pool of text that defines a persona. This task is particularly challenging as it requires evaluators to make subjective judgments based on limited context. Without the ability to compare across multiple generated texts, it becomes difficult to ascertain if a particular piece of content truly reflects the nuances and characteristics of the intended persona. Therefore, the development of robust and nuanced evaluation methods is essential for accurately assessing the performance of personalized language models.

1.3 Research Questions

To tackle the challenges that we mentioned in the previous section, we formulated four specific research questions. Figure 1.4 illustrates the breakdown of our research questions, and how they connect with the main problem statement.

Research Question 1 (*RQ1*)

Can we enhance text classification tasks by incorporating authors' context?

Text classification tasks traditionally rely on isolated text instances paired with labels, often overlooking the rich contextual background that could enhance model understanding. Particularly in social media, where texts are characteristically brief and sometimes miss sufficient context, incorporating additional user-related information could significantly improve the performance. In this research question, we aim to explore approaches that integrate user information in existing text classification tasks. Current datasets usually focus on utilizing text-based information. To address this, we would enrich these datasets with additional user-centric details. Furthermore, it would be beneficial, to construct such datasets from scratch such that it encompasses not only the textual content and its immediate conversational context but also the historical background and social media interactions of the users involved. After constructing such resources, we have the necessary information to explore our user-based approaches. However, we do not only want to integrate the additional user information. What if we can jointly model both users' contexts, together with relations of text and conversational context? We show how we can utilize graph structure to explicitly model users' social and historical context jointly, capturing complex relations between text and its conversational context.

Research Question 2 (*RQ2*)

How can we model the context of recipients to accurately predict their responses to various discourses?

In diverse communication domains, discourse is not solely shaped by the speaker's characteristics, but is also influenced by the audience's traits. Speakers often adapt their language based on the audience they address or the community within which they are communicating, suggesting that a comprehensive understanding of discourse requires more than just an analysis of speaker traits. Unlike the focus on authors' contexts in RQ1, this research question shifts the focus towards the recipients of communication, investigating the potential of modeling the contexts of audiences to predict how they perceive a specific text. Can we model how certain audiences perceive and react to various content? What if we can capture the audience's traits, can we enhance the models' interpretation of their reactions toward different conversation topics?

Our approach involves investigating personalized models that can represent the audience's views and reactions across a variety of topics, examining how these models perform under different experimental setups, across varied topics, and among diverse demographics. By investigating the impact of incorporating audience perspectives into NLP models, we want to find whether a more nuanced understanding of recipient contexts can lead to enhancement in text understanding.

Research Question 3 (*RQ3*)

Do the dynamic user representations help to capture the temporal behavior of users related to their social networks?

In our previous research questions, we focus on approaches that find a single-user representation.

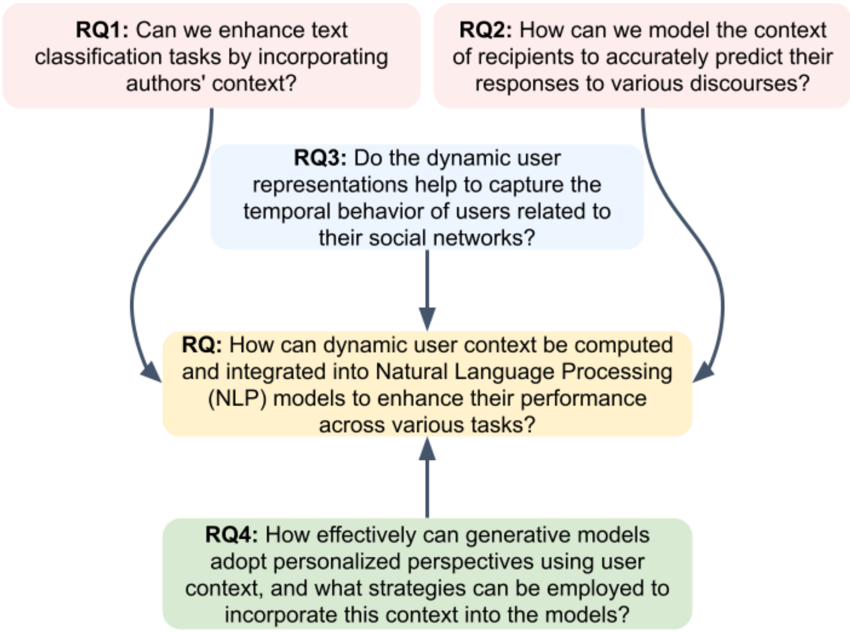


Figure 1.4: Breakdown of our research questions, and their connection to the main problem statement.

Our main assumption was that user context remains consistent throughout their posting history. However, similar to words having multiple meanings across different contexts, a user can have different behavior based on several factors. We hypothesize that users may engage with different subgroups at different times, depending on their shifting interests. In this research question, we focus on constructing dynamic latent user representations based on their semantic and social variations over time. How can we model user representations that can dynamically change based on different social contexts? What if we model different latent representations over a period of time, can we better predict future user behavior based on their interactions?

To examine these temporal dynamics, it is essential to design a framework capable of temporally and jointly modeling these user networks. In order to evaluate the temporal framework, we develop a dataset that concentrates on users and how their interactions evolve over a period of time. We analyze differences between static and dynamic user representation, on a new task focused on users.

Research Question 4 (*RQ4*)

How effectively can generative models adopt personalized perspectives using user context, and what strategies can be employed to incorporate this context into the models?

In the evolving landscape of language technologies, there is an increasing demand for personalized language generation systems that can mirror a user's individual style. Such systems create responses that not only address the user's queries but also resonate with their personal communication style. To facilitate this customization, these models leverage historical interactions and responses to construct a nuanced "persona" of the user. This research question delves into personalized language generation with different types and amounts of context for users. How can generative models adopt personalized perspectives, and what architectures can we utilize to better control language generation? What if we have a large amount of user context, what amount of it is necessary to correctly model the user? How can we better evaluate and compare these models?

To explore these research questions, we construct a realistic dataset with self-reported sentences from users. We analyze the difference of several architecture modifications, that can be utilized for controlled personalized language generation. Additionally, we hypothesize that the utility of different types of historical data may vary depending on the specific architecture employed. Finally, recognizing the limitations of traditional evaluation metrics in capturing the nuances of personalized language generation, we propose a novel human evaluation setting. This evaluation setting is designed to better capture the differences between models in terms of their ability to generate personalized responses that reflect the user's style.

1.4 Thesis Overview

This section highlights the main contributions of this dissertation, concerning the research questions above. For each contribution, we also refer to the scientific publications published during this research period.

1.4.1 Contributions

Our contributions focus on exploring various areas of dynamic user representations in different NLP tasks and experimental setups. These areas include enhancing text classification with user representations, static and temporal analysis of user behavior over time, and modeling different perspectives of users on social media in the context of real-life conflicts.

Contributions for *RQ1*

A deep graph attention-based model to enhance text classification by explicitly modeling users' social and historical context jointly.

Corresponding work: Plepi et al. [13], Plepi et al. [14]

To address research question *RQ1*, we initially expand existing datasets to include users' comments history. Our primary contribution is the development of the first deep graph attention-based model to identify sarcasm on social media by explicitly modeling users' social and historical context jointly, capturing complex relations between a sarcastic tweet and its conversational context. Furthermore, we conduct the first graph-based experiments on the sarcasm perspective detection task. These experiments reveal that our graph network contributes to interpreting the sarcastic intentions of the author more than to predicting the sarcasm perception by others. Finally, we collect a new dataset on Twitter, by extending a semi-supervised method that uses reactive supervision. The dataset contains two types of sarcasm and additional contextual information from conversations and users.

Contributions for *RQ2*

A dataset that focuses on an individual's perspective of conflict situations using Reddit and a comparison of user personalization methods for modeling individual perspectives.

Corresponding work: Welch, Plepi et al. [15], Plepi et al. [16]

In tackling the second research question, *RQ2*, our approach centers on leveraging social media to curate a dataset that contains individual assessments of conflict situations. We propose a novel annotation scheme to annotate a set of 500 conflicts with six aspects of conflict and group them into three different clusters depending on the affinity between people involved in the conflict. Moreover, we address the task of predicting whether someone will perceive the actions of one individual as right or wrong in a given situation. Initially, we provide a discussion regarding the relation between data perspectivism and personalization. Afterward, we propose a novel problem setting, and we explore several user personalization methods for modeling the annotators under different experimental setups. These include averaging the user history, pretraining on authorship attribution tasks, or computing social graph interactions. Finally, given the manual annotations, we perform a comprehensive analysis of how the aspects of conflict and relations between people involved influence the effectiveness of personalization methods in our models. These contributions focus on a deeper understanding of personal perspectives in conflict assessment.

Contributions for *RQ3*

We propose a novel dynamic graph neural network for dynamically capturing users' behavior through time and introduce a user-level factuality and political bias dataset over a period of time.

Corresponding work: Plepi et al. [17], Plepi et al. [18]

In addressing the third research question *RQ3*, we introduce a dataset for distinguishing the authors who have shared news from unreliable sources in the past, from those who share news from reliable sources. This dataset explores the political bias of users in combination with their misinformation-spreading behavior. Our first step involves conducting classification experiments for identifying misinformation spreaders. To achieve this, we utilize a combination of factors: the social connections between users, their historical posting patterns, and the psycho-linguistic features embedded in their communications.

Further, we provide a comprehensive qualitative and quantitative analysis of the users' temporal semantic and social similarities and investigate the different types of dynamic graph connections. Finally, we develop a dynamic graph neural network framework for (a) predicting the users' future misinformation-spreading behavior, (b) predicting the behavior of unseen users, and (c) predicting misinformation-spreading behavior in a zero-shot scenario. Each of these components contributes significantly to our understanding of users' dynamics over time, offering new insights into the prediction and analysis of their behaviors.

Contributions for *RQ4*

We design two transformer architectures to embed personal context focusing on the encoder and decoder respectively, and explore incorporating a variety of different user context types and amounts.

Corresponding work: Plepi et al. [19] (under review)

In this research question, we study personalized language generation. First, we utilize a realistic dataset, that contains comments and self-disclosure sentences for each user profile. The target-generated texts consist of individual judgments of diverse real-life social situations. We develop several architectures inspired by recent work for persona-based generation, that focus on extending encoder-decoder architectures to integrate auxiliary information. This allows for more personalized outputs, that capture the distinctive styles and perspectives of individual users. Finally, we also introduce a new human evaluation setup. This evaluation setup assesses the performance of our models in generating text that not only aligns with the given situational context but also mirrors the style and judgment typical of the user's self-disclosure sentences. This evaluation setup goes beyond traditional accuracy metrics, delving into the qualitative aspects of text generation and its alignment with personalized user profiles.

1.4.2 Publications

The research presented in this thesis is supported by several publications that have contributed significantly to the scientific foundation of the work. These publications have been important in shaping the methodologies, analyses, and discussions detailed in the subsequent chapters. They serve as key reference points for various figures, tables, and concepts explored throughout the thesis.

- ***Peer Reviewed***

  1. **Joan Plepi** and Lucie Flek. 2021. "Perceived and Intended Sarcasm Detection with Graph Attention Networks." In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4746–4753, Punta Cana, Dominican Republic. Association for Computational Linguistics. DOI: 10.18653/v1/2021.findings-emnlp.408
  2. **Joan Plepi**, Flora Sakketou, Riccardo Cervero, Henri Jacques Geiss, Paolo Rosso, and Lucie Flek. 2022. FACTOID: A New Dataset for Identifying Misinformation Spreaders and Political Bias. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 3231–3241, Marseille, France. European Language Resources Association.
  3. **Joan Plepi**, Flora Sakketou, Henri-Jacques Geiss, and Lucie Flek. 2022. Temporal Graph Analysis of Misinformation Spreaders in Social Media. In Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, pages 89–104, Gyeongju, Republic of Korea. Association for Computational Linguistics.
  4. Charles Welch, **Joan Plepi**, Béla Neuendorf, and Lucie Flek. 2022. Understanding Interpersonal Conflict Types and their Impact on Perception Classification. In Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science

(NLP+CSS), pages 79–88, Abu Dhabi, UAE. Association for Computational Linguistics. 10.18653/v1/2022.nlpcss-1.10

5. **Joan Plepi**, Béla Neuendorf, Lucie Flek, and Charles Welch. 2022. Unifying Data Perspectivism and Personalization: An Application to Social Norms. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7391–7402, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. DOI: 10.18653/v1/2022.emnlp-main.500
6. **Joan Plepi**, Magdalena Buski, and Lucie Flek. 2023. Personalized Intended and Perceived Sarcasm Detection on Twitter. In Proceedings of the 3rd Workshop on Computational Linguistics for the Political and Social Sciences, pages 8–18, Ingolstadt, Germany. Association for Computational Lingustics.
7. Heilig, Niclas, Jan Kirchhoff, Florian Stumpe, **Joan Plepi**, Lucie Flek, and Heiko Paulheim. Refining diagnosis paths for medical diagnosis based on an augmented knowledge graph. CEUR Workshop Proceedings 2022.
8. **Joan Plepi**, Charles Welch, and Lucie Flek. 2024. Perspective Taking through Generating Responses to Conflict Situations. In Findings of the Association for Computational Linguistics ACL 2024, pages 6482–6497, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics. DOI: 10.18653/v1/2024.findings-acl.387

The detailed list of publications completed during the PhD term is available in Appendix 8.3.

1.5 Thesis Outline

The structure of the thesis consists of the following eight chapters:

Chapter 1 – Introduction: This chapter outlines the main motivation behind this dissertation. Afterward, it delves into the main research problem and unpacks challenges related to it. In an effort to tackle this main research problem, we clearly define four specific research questions, each connected to the main problem statement. Moreover, this chapter outlines the key contributions made in addressing these research questions, linking them to the scientific publications that form the core of this dissertation. Finally, we provide an overview of the scientific publications which are done during the tenure of this thesis work. Overall, this chapter sets the stage for the detailed discussions that follow in the subsequent chapters.

Chapter 2 – Background: This chapter serves as a detailed review of the existing literature, focusing on previous methodologies for personalization and the integration of user representations in NLP systems across a variety of tasks. It begins with an overview of the literature that attempts to model individual user characteristics, such as demographics, followed by a discussion of approaches that utilize users' social networks in order to capture linguistic homophily. Moreover, we describe the User2Vec methodology, that we use in our experiments to extract initial user representations. Finally, we conclude by describing our proposed approaches to compute and integrate user representations. This chapter not only lays the theoretical foundation necessary for understanding the upcoming research but also positions our work within the broader context of the personalization for NLP area, highlighting the novelty and relevance of our proposed methodologies.

Chapter 3 – Preliminaries: In this chapter, we provide the preliminaries and core architectures and frameworks, necessary for understanding the context and technical aspects of our research. It begins with a detailed description of the transformer architecture which serves as the main backbone for NLP systems. The chapter then progresses to a detailed discussion of pre-trained models for contextualized word representation including BERT, SBERT, GPT, BART, T5. Finally, it provides an overview of graph neural network architectures used during our work.

Chapter 4 – Post Classification with User Context: In this chapter, we address the first research question (*RQ1*). In our attempt to model the author of the text, we focus on exploring different author personalization methods. We start by proposing a heterogeneous graph neural network, that can jointly leverage an author's context from their historical tweets together with the social information from an author's conversational neighborhood in an interaction graph. The main goal of the framework is to combine both textual and author information in order to contextualize the interpretation of the post. Furthermore, this chapter discusses our efforts to develop and curate resources specifically designed for sarcasm detection tasks. Such texts, usually, lack sufficient context, making it difficult for the systems to detect sarcasm. Hence, these resources are constructed to support our user-centered methodology, encompassing both textual content and the essential contextual information needed to investigate the impact of incorporating authors' contexts into text classification. Finally, we report our results across two datasets for the sarcasm tasks and analyze the effect of incorporating author contextual information for the sarcasm detection task.

Chapter 5 – Perspective Classification with User Context: Following our analysis into the enhancement of text classification tasks through the incorporation of authors' context, this chapter delves into the exploration of the second research question (*RQ2*), by focusing on modeling the audience's traits. In this chapter, we focus on capturing the subjective perspectives of the users, across different social norms. A significant contribution of this chapter is the introduction of diverse initialization methods for user representations and their effective integration into models for downstream tasks. This methodology is crucial for enhancing the personalization aspect of our models. Furthermore, we unveil a novel resource developed during this research, that contains text labels, user information, and social context. This dataset revolves around assessing users' perceptions of social situations, sourced from Reddit. We analyze our approaches, under a variety of experimental setups, providing a comprehensive understanding of the effectiveness and versatility of our approaches in different contexts. The chapter concludes with a thorough and extensive analysis of our personalization methods.

Chapter 6 – Static and Temporal User Classification: This chapter is dedicated to addressing the third research question (*RQ3*), which focuses on the modeling of dynamic user representations. First, we introduce the FACTOID dataset, a user-level factuality and political bias dataset, that contains fine-grained scores about the users' factuality and political bias. Our initial experiments, in this chapter, center on identifying misinformation spreaders by utilizing the static global context of users. Subsequent sections, shift the focus to exploring the temporal setup. Initially, we provide a comprehensive qualitative and quantitative analysis of the users' temporal semantic and social similarities and investigate the different types of dynamic graph connections. Finally, we develop a dynamic graph neural network that can capture past user behaviors, to predict the users' future misinformation spreading behavior.

Chapter 7 – Personalized Natural Language Generation: In this chapter, we address the fourth research question (*RQ4*), which focuses on personalized generation task. A key focus here is determining which user information is most beneficial for enhancing our models. To facilitate this investigation, we introduce the PersonaSocialNorms dataset. This dataset contains self-disclosure statements from the users in the dataset, providing a rich source of personalized data.

We develop several architectures inspired by recent work for a persona-based generation, that focuses on extending encoder-decoder architectures, such that they can include auxiliary user context. Finally, we design a new human evaluation annotation, in order to evaluate the persona consistency of our models.

Chapter 8 – Conclusions and Future Work: Concludes the work of this dissertation. It summarizes the core contributions and key findings during our research work. Furthermore, it identifies gaps that remain in the current understanding and suggests directions for future works that can build upon the foundations laid by this dissertation.

CHAPTER 2

Background

In this chapter, first, we provide an overview of previous work on user representations. We review the literature utilizing user traits, or social networks, for different NLP tasks. Moreover, we describe User2Vec methodology, that we use in our experiments to extract initial user representations. Finally, we describe our proposed approaches to compute and integrate user representations.

2.1 Related Work

In recent years, there have been an emphasis on the importance of having personalized NLP systems [20]. Hovy and Yang [21], further expand on this importance, by formalizing and discussing the notion of social factors. They outline seven key factors: 1) speaker, 2) receiver, 3) social relations, 4) context, 5) social norms, 6) culture and ideology, and 7) communicative goals. In this thesis, we will focus on the initial five social factors. The subsequent sections will present a review of earlier studies that incorporate different user traits to personalize NLP systems. Finally, we will provide an overview of some of our proposed approaches designed to incorporate social factors into NLP systems.

User Traits

Since the early days, personalization has been adapted in areas like information retrieval, or recommendation system, recognizing early the obvious need for personalization in the interpretation of the user queries [22, 23, 24].

More recently, the focus has shifted towards integrating user characteristics into various NLP tasks. In early studies, Hovy 2015 [25], adapted the word2vec method in order to learn conditional word embeddings. The main goal is to generate embeddings that reflect group-specific linguistic differences, by conditioning on the corresponding demographic variables across five different languages. The key results from this work, show that by enhancing embeddings with age or gender information, generally improves the model's performance compared to those that do not consider these demographic factors. The experiments are performed on three text classification tasks like topic identification, sentiment analysis, and author attribute identification. Additionally, Benton et al. [26] focus on social media users and learn embeddings that can be used to cluster users with similar behaviors or predict their attributes. They adopted an unsupervised approach based on Generalized Canonical Correlation Analysis (GCCA) [27]. This approach is a multiview technique that learns a single, low-

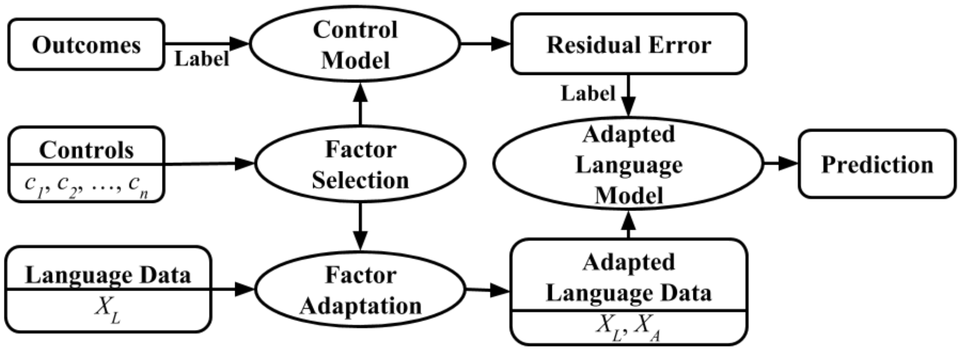


Figure 2.1: Residualized factor adaptation (RFA) method diagram (Source [29])

dimensional vector for each user, that captures the information from their various online interactions. Their experiments show interesting results on applying multiview embeddings on three different prediction tasks user engagement prediction, friend recommendation, and demographic characteristics, highlighting the potential of leveraging user-specific information for enhancing model performance.

Lynn et al. [28], introduce a novel approach, adapting user factors to enhance NLP models based on the author's characteristics of language. They focused on two main types of user factors, 1) inferred demographics and personality traits, and 2) latent language factors that capture variations in language use across different users. Their approach demonstrated improved performance of the models in a variety of tasks like part-of-speech tagging, preposition-phrase attachment, sentiment analysis, and sarcasm detection. Zamani et al. [29], present an approach known as residualized factor adaptation (RFA), for enhancing text-based predictions by incorporating extra-linguistic factors such as demographic or socioeconomic information. RFA approach combines the benefits of two methods: 1) Residualized control and 2) factor adaptation method. Residualized control methods, are similar to ensemble approaches, by constructing two different models one for each type of feature separately, language and extra-linguistic features. In inference, it merges the results from both models to make the final prediction. On the other hand, the factor adaptation method integrates both types of information at the feature level. It produces a new feature space out of language and extra-linguistic features for training a single model. RFA method combines the advantages of both methods, without increasing the complexity of the model. It attempts to combine both simultaneously the linguistic and extra-linguistic features on two levels, the feature level and the model level. An overview of the RFA methodology is illustrated in Figure 2.1.

Moreover, Kolchinski and Potts [30], investigated two novel strategies for representing authors in the context of textual sarcasm detection. They explore a Bayesian approach that directly represents authors' tendencies towards sarcasm, and a dense embedding technique that can learn interactions between the author and the linguistic content. Their experiments are conducted on the sarcasm detection task with data collected from Reddit. Their findings show that augmenting a bidirectional RNN with dense embeddings, improves performance. Furthermore, Welch et al. [31] propose an innovative approach of personalized word embeddings that use demographic-specific word representations by incorporating full or partial demographic information for a user. Such information includes gender, age, location, religion, and word associations. They expand the original word2vec method, by integrating an additional matrix, that encodes the extra demographic values. Their experiments are performed on two tasks for the English language: 1) language modeling and 2) word association. They find that

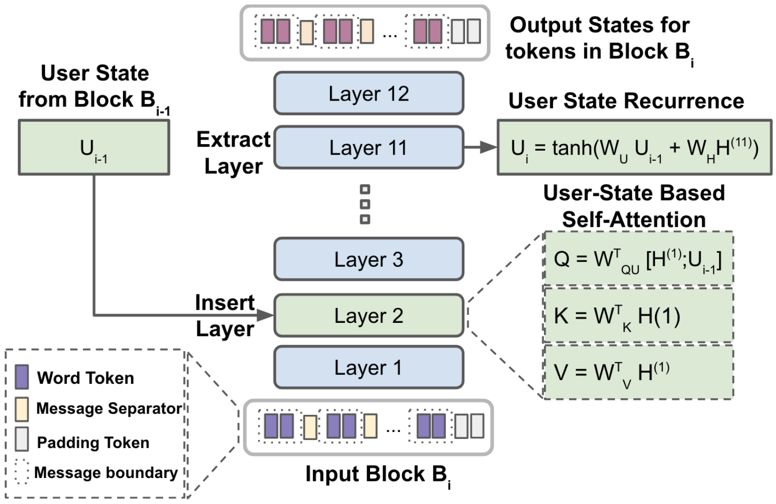


Figure 2.2: HaRT architecture proposed by Soni et al. (Source [32]).

incorporating demographic-aware word representations outperforms generic word representations without demographic information. Contrasting with these methods, which primarily integrate user characteristics into existing NLP frameworks, Soni et al. [32], proposed human language modeling (HuLM). HuLM introduces a hierarchical extension to the language modeling problem, that includes a human-level context to connect sequences of documents (like social media messages) and capture the idea that human linguistic expression is influenced by fluctuating human states. They developed HaRT model, which is a large-scale transformer model for the HULM task, pre-trained on data from around 100,000 social media users. HaRT has shown it's effectiveness not just in improving language modeling accuracy (as measured by perplexity) on social media texts but also in enhancing performance across four downstream applications at both document and user levels, including stance detection, sentiment classification, age estimation, and personality assessment.

Social Networks

While including demographic factors improves the performance of existing models across various tasks, another line of research exploits integrating social relations factors into NLP models. Yang et al. [33] show how to exploit social networks to enhance model robustness to nuances of linguistic features on the sentiment analysis task. The main motivation is based on linguistic homophily, which consists of the tendency of individuals within a social network, to adopt similar language usage patterns. They construct an attention-based neural network architecture, that dynamically divides attention among multiple basis models, based on an author's social network position. Hence, it effectively utilizes the tendency of individuals who are part of the same social networks, and have similar language.

This model is inspired by ensemble learning and employs a social attention mechanism. The final prediction aggregates the outputs of the basis models, by computing a weighted combination, where each author has a unique weighting, based on their position within their social network. Moreover, Mishra et al. [34] developed SNAP-BATNET, which is a deep learning model, designed to detect suicide ideation on social media. This model combines text-based feature extraction with a novel feature stacking approach. They utilize information from social networks in the form of graph embeddings and author profiling using features from historical data. Del Tredici et al. [35] present a model based on Graph Attention Networks [36], aimed at capturing sociolinguistics features from individual interaction across different communities. They perform experiments on three different tasks containing data from Twitter (now X) namely: sentiment analysis, stance detection, and hate speech detection. Investigating the predictive power of social networks, Pan et al. [37] explore the task of predicting a user's occupational class. Pan et al. show that the content information of a user's tweets, their community's profile descriptions, and the user's social network provide relevant information for classifying a user's occupational group. Finally, Mishra et al. [38] explore capturing the structure of online communities together with the linguistic behavior of the users within them on the task of automated abusive language detection on Twitter. Focusing on the detection of abusive language, Mishra et al. [38] explored the importance of capturing both the structure of online communities and the linguistic behavior of their members. Their work highlights the potential of integrating community structure and user-specific linguistic patterns to improve the detection of abusive content on platforms like Twitter.

These studies illustrate the growing recognition of user and social relational factors as a necessary component for enhancing NLP models, offering new perspectives on leveraging these features to better understand and predict language more accurately.

2.2 User Representations

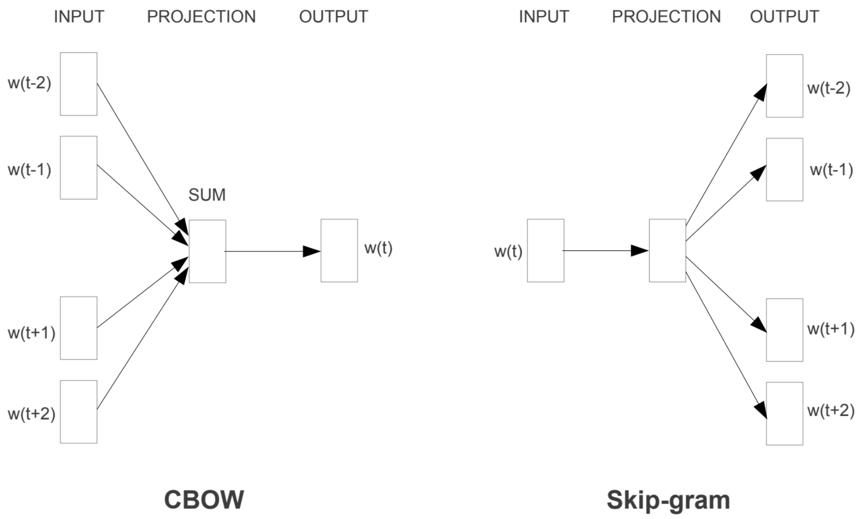


Figure 2.3: CBOW and skip-gram models (Source [1]).

Word2Vec

Word embeddings play an important role in NLP, due to their ability to convert words into vectors. These vectors can effectively capture the semantic and syntactic information of the word in a text. Word representations have enhanced the performance of NLP systems across several downstream tasks including named entity recognition, sentiment analysis, machine translation, question answering, and more. Skip-gram and a continuous bag of word models are the earliest techniques to efficiently convert words to vector representations [1, 2].

The initial architecture, represented by the CBOW model on the left side of Figure 2.3, is essentially a simple feed-forward network. In this model, the inputs are the one-hot encoding of the words in the training set. The one-hot encoding of a word is a binary vector where the position corresponding to the word in the vocabulary is assigned a value of 1, while all other positions are set to 0:

$$\begin{aligned} king &= [0, 1, 0, 0, 0] \\ queen &= [0, 0, 1, 0, 0] \\ woman &= [0, 0, 0, 0, 1]. \end{aligned}$$

The primary objective of the CBOW model is to predict the current word within a given context of surrounding words. During training, the model learns to predict the target words and updates the weights of the model, which correspond to the word representations. The output error is calculated by comparing the one-hot encoding of the target word with the output probabilities generated by the network. Conversely, the skip-gram model, as illustrated in Figure 2.3 on the right, approaches the problem from the opposite perspective. In this model, the target word becomes the input to the network, and the task is to predict the context words surrounding the target word. Formally, given a sequence of words $\langle w_1, \ldots, w_T \rangle$ the skip-gram model aims to maximize the following objective:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c \wedge j \neq 0} logp(w_{t+j}|w_t). \tag{2.1}$$

Unlike the CBOW model, the skip-gram model generates multiple probability distributions (usually equal to the number of context words, denoted as C). Each probability distribution corresponds to predicting one of the context words associated with the target word. Similar to CBOW, skip-gram also learns word vectors from the model's weights, and the learning process is achieved through backpropagation. Both models are effective in learning word representations and can be applied depending on the specific NLP task and context.

Both models have impressive results, however, skip-gram architecture works better in representing semantic information of the words, while CBOW architecture performs better in the syntactic part [1]. On the other hand, skip-gram architecture performs better in less frequent words, while CBOW is faster to train, and has better representation in more frequent words.

2.2.1 User2Vec

Word2vec methods are utilized to convert words to vector representations, whereas user2vec [39] focuses on mapping users to vector representations. Similar to word2vec methods that learn the probability of words co-occurring with each other, user2vec [39] aims to model the probability of an author co-occurring with texts. This approach captures relations between users and the texts that they have written, by optimizing the conditional probability of the texts given the vector representations of the respective authors.

Formally, given a sentence $S = \{w_1, \ldots, w_N\}$, where w_i represent a word drawn from a vocabulary V, the aim is to maximize the following probability:

$$p(S|user_j) = \sum_{w_i \in S} logp(w_i|\mathbf{u}_j) + \sum_{w_i \in S} \sum_{w_k \in C(w_i)} logp(w_i|\mathbf{e}_k) \quad (2.2)$$

where $C(w_i)$ denotes the set of words inside the predefined window around the target word w_i. $\mathbf{e}_k \in \mathbb{R}^d$ denote the embedding of word k and $\mathbf{u}_j \in \mathbb{R}^d$ denote the embedding of user j. In this conditional probability, the occurrence of the word w depends on both author u_j of S and words inside the context window. The conditional probabilities in Equation 2.2 can be estimated with log-linear models of the form:

$$p(w_i|x) = \frac{exp(\mathbf{W}_i \cdot x + b_i)}{\sum_{k=1}^{Y} exp(\mathbf{W}_k \cdot x + b_k)} \quad (2.3)$$

, where x denotes a feature vector, $\mathbf{W}_k$ and b_k, represent the weight matrix and bias for class [1] k respectively. The denominator in Equation 2.3 requires summing over all the words in the vocabulary, hence it is approximated using Hierarchical Softmax [40]. The user vector representations, need to capture individual writing styles in the form of word usage patterns. Hence, $P(w_i|\mathbf{u}_j)$ is approximated by minimizing the following hinge-loss objective:

$$\mathcal{L}(w_i, user_j) = \sum_{w_l \in V, w_l \notin S} max(0, 1 - \mathbf{e}_i \cdot \mathbf{u}_j + \mathbf{e}_l \cdot \mathbf{u}_j) \quad (2.4)$$

where w_l is a negative example that is not included in the text written by user j. This objective aims to train a model that learns user representations that differentiate how different users use words. The main idea is to use negative sampling, in order to approximate the objective function in a binary classification task that discriminates between positive examples and sampled negative examples.

This approach performs well in tasks that learn representations when there is a sufficient amount of training data [41]. However, there might be cases where we have a limited amount of text for each user. This problem is tackled by the authors, by carefully selecting the negative samples. One approach would be to sample an unigram distribution estimated from the posts written by all the users. The main reason is to select the most commonly used words as the negative samples, thereby forcing the representations to capture the differences between the words written by a user and the words that are used most commonly by everyone else.

[1] A class in this context, refers to the target word to be predicted.

2.3 Proposed Approaches

2.3.1 Individual User Modeling

In this work, we enhance text classification tasks by utilizing additional author or audience information. Given a textual input x, and the author [2] a, the goal of the model is to output the personalized target y. Formally, we can model the task as

$$\hat{y} = argmax_{y \in Y} p(y|x, a). \tag{2.5}$$

The models can utilize the history texts of the author a, $\mathcal{H}^a = \{r_1, r_2, \ldots, r_m\}$, where r_i is a past comment of in author's timeline. In this section, we utilize the following approaches to encode author information: a) Priming, b) Author ID, c) Average SentenceBERT for authors (A-SBERT), d) Authorship Attribution (AA) e) Graph Neural Networks (GNN).

Priming

This method, originally used in recurrent neural networks, passes data from a given author through the model to alter the parameters before passing the text to use for prediction [42, 43]. In our work, we also sample author data, but instead append it to the text to classify. For our purpose, we randomly sample a number of tokens and append them as a prefix to the text to classify. For each author a, we randomly sample a number of tokens from their historical texts $\mathcal{H}^a$ until the maximum number of tokens is less than k or corresponds to the number of tokens in their historical texts. We append the sampled text to the beginning of the input text [3]. Given the input sequence of n tokens $\langle w_1, w_2, \ldots, w_n \rangle$, and the sampled sequence from author history $\langle r_1, ..., r_k \rangle$, we model the task as:

$$\hat{y} = argmax_{y \in Y} p(y|w_{1:n}, r_{1:k}). \tag{2.6}$$

Average SentenceBERT for authors (A-SBERT)

Given an author a and their historical texts, $\mathcal{H}^a$. We compute the author representation by averaging the SentenceBERT [44] text embeddings h_i of all $r_i \in \mathcal{H}^a$, resulting in:

$$\widetilde{a} = \frac{1}{|\mathcal{H}^a|} \sum_{k=1}^{|H^a|} h_i. \tag{2.7}$$

As noted in Figure 2.4, we utilize these embeddings as initial representations for our authors in the architecture pipeline.

[2] We are going to use the term author in this section. However, the same approaches can be employed for modeling a recipient of a text given his past history of texts.

[3] We also tried concatenating to the end in preliminary experiments, though we found performance was slightly lower.

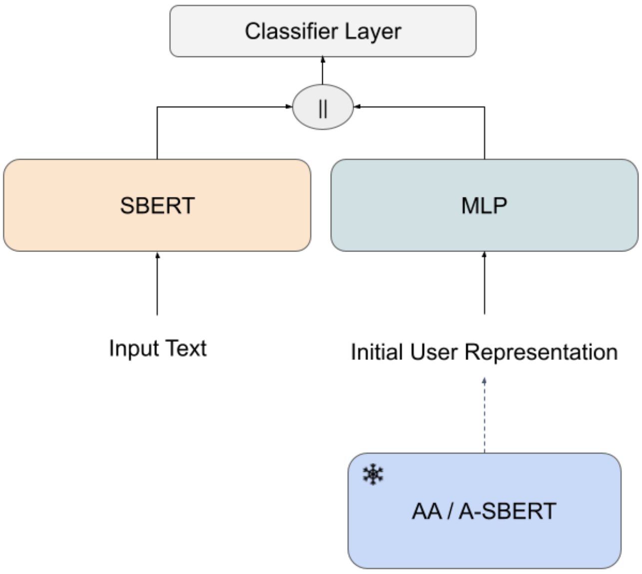


Figure 2.4: In this figure we show how we combine pre-computed author/recipients representation with SBERT. A-SBERT, and AA, are separate encoding methods, to extract initial user representations, utilizing their comments during the history. After computing those, we combine both user and text representations to classify. The encoding layer is frozen during training.

Authorship Attribution (AA)

With this technique, we pre-train a neural network to predict the author of a text. Formally, for a given input text $x = \langle w_1, w2, \ldots w_n \rangle$ we model the classification task as:

$$\hat{a} = argmax_a p(a|w_{1:n}) \tag{2.8}$$

.

In our network we utilize SBERT, to encode the initial representations of the input text. We freeze the SBERT weights, and only use it as a sentence encoder. We forward the SBERT representations h of text x into a two-layer feed-forward network parameterized from weight matrices $W_1 \in R^{\frac{d}{2} \times d}$ and $W_2 \in R^{|\mathcal{H}| \times \frac{d}{2}}$, where d is 768 (dimension of the SentenceBERT text embeddings), and $|\mathcal{H}| \equiv$ number of authors during the training. Then, we forward the output of the last linear layer to a softmax layer to get a distribution over the authors. After training, we use the linear layers to extract a representation of the author.

For each author a, we forward all their historical texts $r_i \in H^a$, where $i = (1, \ldots, m)$ to the trained network, extracting the predictions:

$$Y = \{y_r | r \in H^a\}. \tag{2.9}$$

Next, we initialize a vector $\widetilde{h}^a$ of size $|\mathcal{H}|$, where:

$$\widetilde{h}^a_i =| \{y | y \in Y \wedge y = i\} | \tag{2.10}$$

for $i = (1, ..., |\mathcal{H}|)$, representing the number of times each author is predicted for all texts of user a.

We extend this representation by normalizing the vector so that the sum of all predictions is equal to 1 and thus get another representation - the distribution of authors predicted.

Similar to A-SBERT, Authorship Attribution method is also an encoding method for extracting initial user representations, utilizing their past texts during the history. As depicted in Figure 2.4, we combine both user and text representations, to classify the given input text.

2.3.2 Modeling User's Network Context

Apart from the importance of modeling the author of a given text, certain understanding is needed between the audience and the author [45, 46]. Our goal is to model relations between users and their past texts, interactions between users, and relations between tweets in one conversation. We model these relationships as a graph $\mathcal{G} = (V, E)$. The graph $\mathcal{G}$ can be homogenous, where $V = \{U\}$, or heterogeneous (Figure 2.5) where $V = \{U \cup T\}$ contains two types of nodes - users[4] and texts. In heterogeneous graph, we use three edge types $E = \{\mathrm{e}^U \cup \mathrm{e}^T \cup \mathrm{e}^C\}$, where e^U represents the social interaction between users. This involves quotes, mentions, or replies in the user history. e^T denotes the edges between tweets that are involved in one discussion thread, with all tweets connected with each other, and e^C is the relation between a tweet and its author.

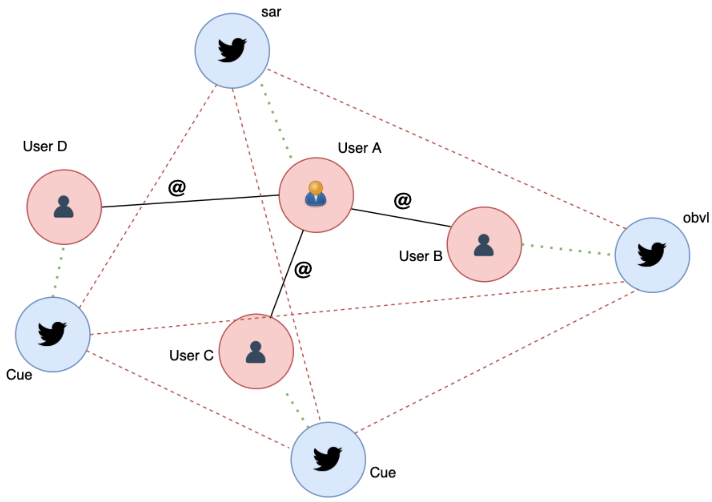


Figure 2.5: An example of a heterogeneous user and tweet social graph extracted from one conversation for the sarcasm task.

Graph Representation Learning We use Graph Attention Networks (GATs, [47]) to exploit the neighborhood of each node to compute the final representations.[5] GAT uses a self-attention mechanism [48, 49] to assign an importance score to the connections that contribute more to the text classification. We initialize the user and the text nodes of the GAT with their corresponding embeddings $\widetilde{\mathbf{a}}_i$ and $\widetilde{\mathbf{t}}_i$. The initial node representation of each node $v \in V$ is linearly transformed by a

[4] We use users and authors interchangeably.

[5] We ran early experiments with Graph Convolutional Networks as well, obtaining inferior and less interpretable results.

weight matrix $\mathbf{W} \in \mathbb{R}^{d' \times d}$ into a vector $\mathbf{h}_v \in \mathbb{R}^{d'}$. Following, the attention weights e_{vn} of each node v are computed as:

$$e_{vn} = att(\mathbf{h}_v \| \mathbf{h}_n) \tag{2.11}$$

where $n \in \mathcal{N}(v)$ is a node in the neighborhood of v and att is the attention mechanism function which is a single-layer feedforward neural network, parameterized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2d'}$ with a LeakyReLU nonlinearity.
The final node representation $\mathbf{h}'_v \in \mathbb{R}^{K \cdot d'}$ is computed as:

$$\mathbf{h}'_v = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{n \in \mathcal{N}(v)} \alpha_{vn}^k \mathbf{W}^k \mathbf{h}_n\right) \tag{2.12}$$

where K is the number of attention heads, σ is the ReLU nonlinear function, $\mathbf{W}^k \in \mathbb{R}^{d' \times d'}$ a weight matrix and $\alpha_{vn}^k = softmax(e_{vn}^k)$ the normalized attention weights from the k-th attention mechanism att^k.

2.4 Summary

In this chapter, we provided a detailed review of the existing literature, focusing on previous methodologies for personalization and the integration of user representations in NLP systems across a variety of tasks. First, we provided a literature review, over works that focus on modeling individual user characteristics, such as demographics or post history. Next, we provided a discussion of approaches that utilize users' social networks in order to capture the tendency of individuals within a social network, to adopt similar language usage patterns. Finally, we defined the text classification task with additional user context, followed by a detailed description of the approaches derived during the work of this thesis. In the next chapter, we will provide an overview of key architectures that serve as a backbone for our methodologies.

CHAPTER 3

Preliminaries

This chapter provides the key concepts that lay the foundations for addressing the research problem defined in Chapter 1. We will provide a detailed description of the transformer architecture that serves as a backbone for various modern NLP systems. Following transformer architecture, we will provide a detailed discussion of pre-trained transformer-based models used in our thesis like BERT, SBERT, GPT, BART, and T5. Lastly, the chapter transitions into a detailed discussion on graph neural network architectures. We provide insights into the workings of Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and Heterogeneous Graph Convolutional Networks (HGCN). Each of these architectures plays a critical role in our research, especially in terms of understanding and leveraging the complex relationships inherent in user data.

3.1 Transformer

The Transformer model [49], illustrated in Figure 3.1, marks a significant advancement in NLP. This model architecture relies on attention to boost the training speed of the model, and significantly improve performance. Unlike its predecessors such as recurrent neural networks (RNN), long short-term memory networks (LSTM) [50], and gate recurrent units (GRU) [51], the Transformer network does not use any recurrence in its architecture, in addition to enabling training acceleration and parallelization of computations.

3.1.1 Encoder-Decoder structure

The Transformer network follows an encoder-decoder structure. The encoder's objective is to transform an input sequence $x = (x_1, \ldots, x_n)$ to a sequence of continuous representations $z = (z_1, \ldots, z_n)$. Given the $\mathbf{z}$ representations, the decoder generates the output sequence $y = (y_1, \ldots, y_m)$

Encoder consists of six identical layers, each comprising two sub-layers. The first sub-layers is a multi-head self-attention mechanism, allowing the model to focus on different parts of the input sequence. The second sub-layer is a fully-connected feed-forward network, applied separately to each position of the first attention sub-layer's output. To enhance training efficiency and facilitate the gradient flow (the vanishing gradient problem), each sub-layer includes residual connections [52], followed by layer normalization [53].

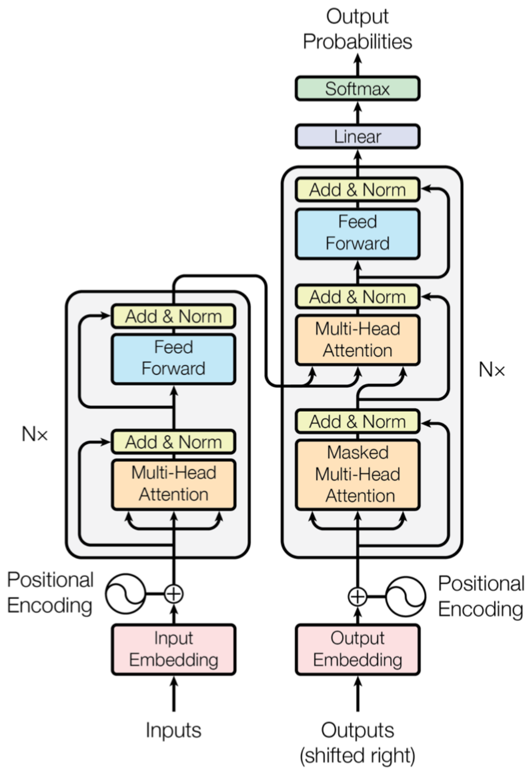


Figure 3.1: The Transformer network architecture (Source [49]).

Decoder, similar to the encoder, is composed of six identical layers. These layers include the same types of sub-layers as the encoder, with an additional sub-layer that performs multi-head cross-attention over the encoder's output. The decoder also uses residual connections and layer normalization for each sub-layer. Additionally, the decoder contains a mechanism to prevent subsequent positions from influencing current positions, which modifies the self-attention sub-layer by applying a masking technique. The masking hides the subsequent values by replacing those with ∞. This ensures that for any position i, the model can only utilize information from positions before i.

3.1.2 Attention in Transformer

In the Transformer model, the encoder layer is pivotal in capturing relevant information from other words within the input sequence to better represent the word currently under consideration. This improvement in representation is achieved through the self-attention sub-layer. Self-attention serves as a mechanism that maps a query and a set of key-value pairs to an output, effectively determining the importance or relevance of other words in the context of the current word.

The scaled dot-product attention is illustrated in the left of Figure 3.2. It begins with the embedding of the current word, which is then multiplied by three matrices that have been learned during the training phase to produce query (Q), key (K), and value (V) vectors. The attention output is calculated using the formula:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{3.1}$$

where Q, K, V represent the set of queries, keys, and values respectively, and d_k is the dimension. The model opts for dot-product attention over additive attention.

To enhance the effectiveness of the attention mechanism, the Transformer model introduces a concept known as multi-head attention. Instead of relying on a single attention function, the model generates multiple sets of queries, keys, and values by projecting them h times through distinct linear projection matrices. This approach allows each "head" to independently process its version of the query, key, and value vectors in parallel. The outputs from all heads are then concatenated and passed through an additional projection matrix to produce the final output. The right side of Figure 3.2 depicts this multi-head attention mechanism. The computation is formalized as follows:

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, ..., head_h)W^O \\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V), \end{aligned} \tag{3.2}$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_u}$ and $W^O \in \mathbb{R}^{hd_u \times d_{model}}$ are the learned parameters of the model. This multi-head attention enables the model to simultaneously capture information from different representational spaces. As a result, the Transformer can more effectively focus on varied aspects of the input data in parallel, improving its ability to understand complex dependencies and capture long-distance relationships within the data.

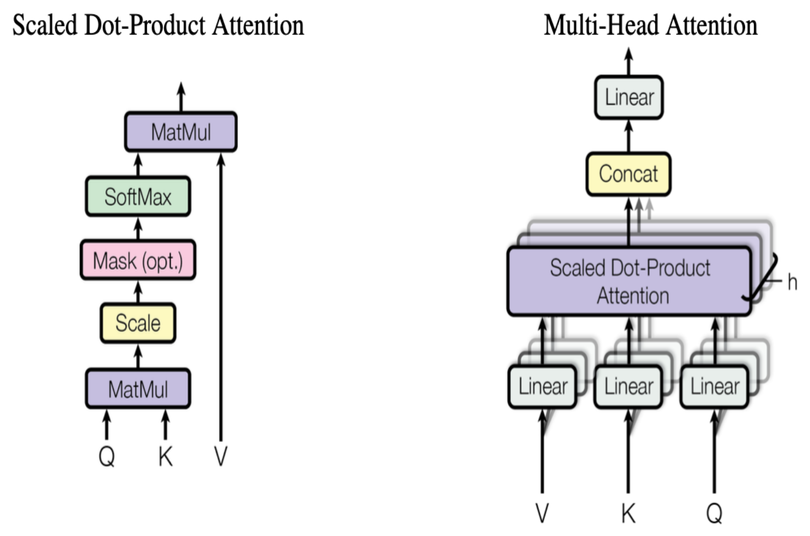


Figure 3.2: Scaled Dot-Product Attention and Multi-Head Attention layers (Source [49])

3.1.3 Position-wise Feed-Forward Networks and Softmax

Another component of the Transformer's architecture is a position-wise feed-forward network, which is present in every layer of both the encoder and decoder. This sublayer consists of two linear transformations with ReLU activation in between, and it is applied to every position independently. The function of the feed-forward network can be expressed as:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2. \tag{3.3}$$

For representing input words as vectors, the model uses pre-trained embeddings within the encoder. On the other hand, the decoder, for each output symbol, produces vector representations of a fixed dimension (d_{model}). A linear projection layer followed by a softmax function is then applied during training to convert these representations into a probability distribution over the output vocabulary.

3.1.4 Positional Encoding

Due to the lack of recurrence in its structure, the Transformer adds some new vectors to each input embedding to incorporate sequence order information. It achieves this by adding positional encoding vectors to the input embeddings, which encode both the absolute and relative positions of tokens in the sequences. These positional encodings use sine and cosine functions of different frequencies to represent the position of each word within the sequence:

$$\begin{aligned} PE(pos, 2i) &= sin(pos/10000^{2i/d_{model}}) \\ PE(pos, 2i+1) &= cos(pos/10000^{2i/d_{model}}), \end{aligned} \tag{3.4}$$

where d_{model} represenet the dimension of the model's embeddings, pos indicates the position, and i is the index of dimension. This use of sinusoidal functions allows the model to effectively capture positional information and ensure that the representation of each word carries a notion of its location within the sequence.

3.2 Transformer-based Architectures

The Transformer model has played a major role in boosting the performance of many natural language processing downstream tasks. The Transformer finds usage in several recent language models like [4, 5, 54].

3.2.1 BERT

The BERT (Bidirectional Encoder Representations from Transformers) model [4] builds upon the Transformer architecture. BERT utilizes a multi-layer bidirectional Transformer encoder, significantly improving the model's understanding of language context and semantics.

For text tokenization, BERT employs the WordPiece method [55] with a 30 000-token vocabulary, marking the sub-word parts with ##. A special token "[CLS]" is placed at the beginning of every input sequence, which is used in classification tasks by representing the entire sequence's context. For inputs consisting of sentence pairs, these are merged into a single sequence separated by another special token "[SEP]".

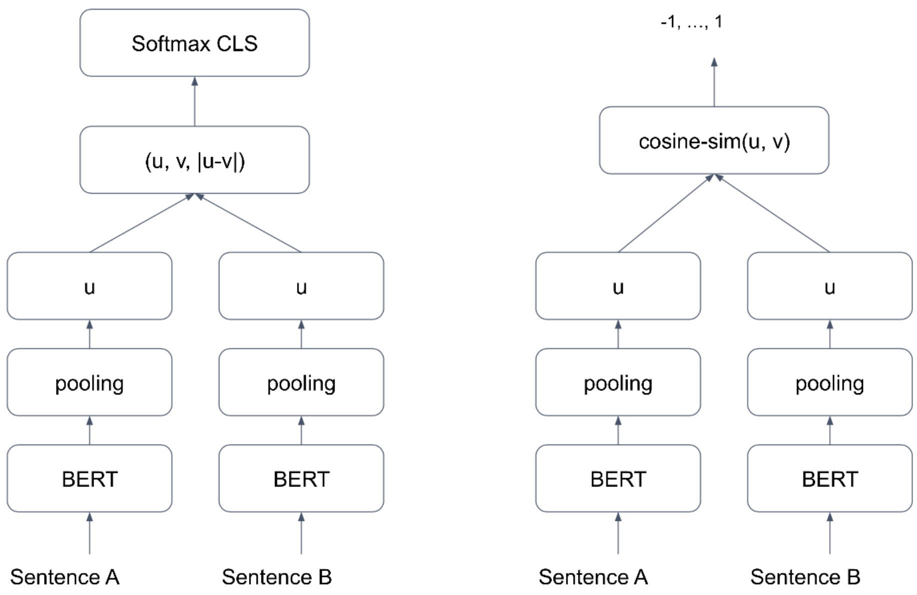


Figure 3.3: On the left, SBERT architecture with classification objective function. The two BERT networks have tied weights (siamese network structure). On the right, SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function (Source [44]).

BERT is pre-trained on a large corpus that combines English Wikipedia (2.5 billion words) with the BooksCorpus (800 million words), utilizing two unsupervised tasks. The first task is masked language modeling (MLM), designed to overcome the limitations of unidirectional models by enabling the training of a model that incorporates context from both directions. To achieve this, BERT randomly masks 15% of tokens in the input and trains the model to predict these masked tokens, a procedure which is commonly referred to as Cloze task [56]. They apply the following procedure for a chosen token:

- 80% of the time the model replaces with $[MASK]$ token
- 10% of the time the model replaces with a random word
- 10% of the time the model keeps the word unchanged

This strategy ensures the model maintains a broad context understanding without relying solely on the masked tokens. However, there would be a slower training convergence and a bigger amount of training iterations due to the reduced focus on only 15% of tokens.

The second pre-training task is next sentence prediction, essential for tasks requiring an understanding of sentence relationships, such as Question Answering and Natural Language Inference. In next sentence prediction, BERT learns to determine whether a sentence B logically follows a sentence A, with B being the actual subsequent sentence 50% of the time (positive label) and a random sentence the other 50% (negative label). This task helps BERT learn effective sentence-level representations, similar to representations learned from objectives in prior works [57] and [58].

3.2.2 Siamese BERT Networks

A significant limitation of the original BERT model is its inability to generate independent sentence embeddings, which complicates the process of obtaining useful sentence embeddings. Common methods used from the researchers to derive sentence embeddings is to average the BERT output layer (known as BERT embeddings) or utilizing the embedding of the first [CLS] token, but these techniques often result in not suitable sentence representations. Addressing this issue, Sentence-BERT (SBERT) was introduced by Reimers and Gurevych [44]. SBERT adapts the structure of siamese and triplet networks to produce semantically meaningful sentence embeddings efficiently.

SBERT extends the BERT and RoBERTa models [59] by incorporating an additional pooling operator on their outputs. The authors experimented with three different pooling strategies: taking the output of the [CLS] token, calculating the mean of all output vectors (`MEAN` strategy), and performing a max-over-time operation on the output vectors (`MAX` strategy). Furthermore, siamese and triplet network [60] architectures were utilized to refine the training process. The primary objective was to train the network weights such that the resulting sentence embeddings could be effectively compared using cosine similarity, making them semantically meaningful. The following structures and objective functions are used:

Classification Objective Function. The sentence representations u, and v respectively, are concatenated with element-wise different $|u - v|$ and multiplied with weight $W_t \in \mathbb{R}^{3n \times k}$ to compute the output:

$$o = softmax(W_t(u, v, |u - v|)) \tag{3.5}$$

where n is the dimension of sentence embeddings and k, is the number of labels. This structure is shown on the left of Figure 3.3.

Regression Objective Function. In this objective function, shown on the right of Figure 3.3, the authors compute the cosine similarity between sentence embeddings u and v. The mean-squared error loss is used as the objective function.

Triplet loss. Triple loss tunes the networks such that the distance between given an anchor sentence a and a positive sentence p is smaller than the distance between a and a negative sentence n. Formally we can minimize the given loss function:

$$max(||s_a - s_p|| - ||s_a - s_n|| + \epsilon, 0) \tag{3.6}$$

where s is the corresponding sentence embedding for a, p or n, ϵ is a given margin, and $|| \cdot ||$ is a distance metric. In their experiments, they use Euclidean distance as a metric and set $\epsilon = 1$.

The dataset used to train SBERT is a combination of SNLI [61] and Multi-Genre NLI [62]. SBERT is fine-tuned using a 3-way softmax classifier objective function, batch size is set to 16, Adam optimizer with learning rate 2e-5, and a linear learning rate warm-up over 10% of the training data. The default pooling strategy is set to `MEAN`.

3.2.3 OpenAI GPT

Language Modeling. The OpenAI GPT (Generative Pre-trained Transformer) model represents a significant advancement in the field of language modeling, aiming to generate highly contextualized word representations. Unlike BERT that utilizes an encoder component, GPT focuses on a decoder-only approach. This section explores the foundational aspects of language modeling, followed by an overview of the GPT model's architecture and training procedure.

A language model (LM), computes the probability distribution of a sequence of words (tokens) within a given vocabulary $\mathbb{V}$. Given a sequence of token $w_1, \ldots, w_m \in \mathbb{V}$, the language model assigns a probability p over the sequence, $p(w_1, \ldots, w_m)$. This probability distribution can be decomposed using the chain rule of probability as follows:

$$p(w_{1:m}) = p(w_1)p(w_2|w_1)p(w_3|w_2, w_1) \ldots p(w_m|p_{1:m-1}) = \prod_{i=1}^{m} p(w_i|w_{1:i-1}) \tag{3.7}$$

where $p(w_i|w_{1:i-1})$ represents the conditional probability of the word w_i given the previous words in the context $w_{1:i-1}$. Autoregressive language models leverage neural networks to compute the conditional probability $p(w_i|w_{1:i-1})$.

The introduction of the Transformer architecture, laid foundations or creating contextualized word representations adaptable to various tasks with minimal adjustments, like BERT. One such approach is also OpenAI-GPT model [5], which perform a semi-supervised approach for universal language understanding, combining unsupervised pre-training, and supervised fine-tuning.

Unsupervised pre-training. In the first part, they utilize a large corpus with unlabeled text, in order to train a language model. Formally, given an unsupervised corpus of tokens $\mathcal{U} = u_1, \ldots, u_n$, they maximize the following log-likelihood using a language modeling objective:

$$L_1(\mathcal{U}) = \sum_i logP(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta) \tag{3.8}$$

where the conditional probability P is modeled using a neural network with parameters Θ, and k is the size of the context window. Their language model consists of a multi-layer Transformer decoder-only variant [49, 63]. As described in section 3.1, this model applies a multi-headed self-attention operation over the input context tokens as follows:

$$\begin{aligned} h_0 &= UW_e + W_p \\ h_l &= transformer\_block(h_{l-1}) \forall l \in [1, n] \\ P(u) &= softmax(h_n W_e^\top) \end{aligned} \tag{3.9}$$

where U is the context vector of tokens, W_e is the token embedding matrix, W_p is the position embedding matrix,n is the number of layers, and $P(u)$ is the output distribution over the target tokens.

Supervised fine-tuning The second part consists of adapting the parameters of the pre-trained model to a specific supervised task. For the supervised task, is assumed a dataset C which consists of sequences of input tokens $X = x^1, \ldots, x^m$, along with a label y. The inputs are passed through the pre-trained model, and the final activations h_l^m are used as input to an extra output linear layer to predict y as follows:

$$P(y|x^1, \ldots, x^m) = softmax(h_l^m W_y) \tag{3.10}$$

where W_y are the parameters of the output linear layer. Furthermore, the following objective function can be used to train the model:

$$L_2(C) = \sum_{(x,y)} logP(y|x^1, \ldots, x^m). \tag{3.11}$$

Additionally, it was found that a combination of language modeling objective with task-specific objective improves the generalization and convergence. Formally, the following objective function can be optimized:

$$L_3(C) = L_2(C) + \lambda * L_1(C) \tag{3.12}$$

where λ is the weight for the language modeling objective.

Model Specifications The initial OpenAI-GPT model, consists of a 12-layer decoder-only transformer with 12 masked self-attention heads and dimension state 768. The model is trained using stochastic gradient descent [64], and Adam optimizer [65], with a maximum learning rate of $2.5e-4$. Warmup steps were set to 2000, and cosine learning rate schedule. The model was trained for 100 epochs, using minibatches of 64, and a maximum length of 512. They used bytepair encoding vocabulary (BPE) [66] with 40,000 merges, and a dropout rate of 0.1 for regularization. Instead of ReLU, they utilize GeLU [67] as an activation function, and initialize the parameters from $\mathcal{N}(0, 0.2)$.

The following releases of GPT models, namely GPT-2, GPT-3, GPT-4 [68, 69], follow the same architecture, with a few modifications. However, the biggest upgrade of the new models consists of the amount of data and the size of the models which have scaled to billions of parameters [70]. The bigger models introduce better language understanding with increased capabilities, making it possible to adapt to new tasks without further training but only utilizing few/one/zero-shot learning.

3.2.4 BART

In the previous sections, we saw different variants of Transformers models [49], like BERT (see Section 3.2.1) a bidirectional encoder, or GPT (see Section 3.2.3), a left-to-right decoder. In this section, we describe BART [71], a model that merges the strengths of both bidirectional and auto-regressive approaches seen in models like BERT and GPT.

BART model different from the models we saw previously, follows a standard sequence-to-sequence Transformer architecture, and utilizes GeLU [67] as an activation function instead of ReLU. This model has two key differences with BERT [4]: a) each layer of the decoder performs cross-attention over the final hidden layer of the encoder, and b) BART does not utilize a feed-forward network before word prediction.

Pretraining. Pretraining of BART has two main parts: (1) documents corruption by utilizing various noising functions, and (2) model training to reconstruct the original text. For the first part, they experiment with different noising schemes, as shown in Figure 3.4:

- **Token Masking.** This approach replaces random sampled tokens with `[MASK]` tokens (this approach is similar to BERT in Section 3.2.1).

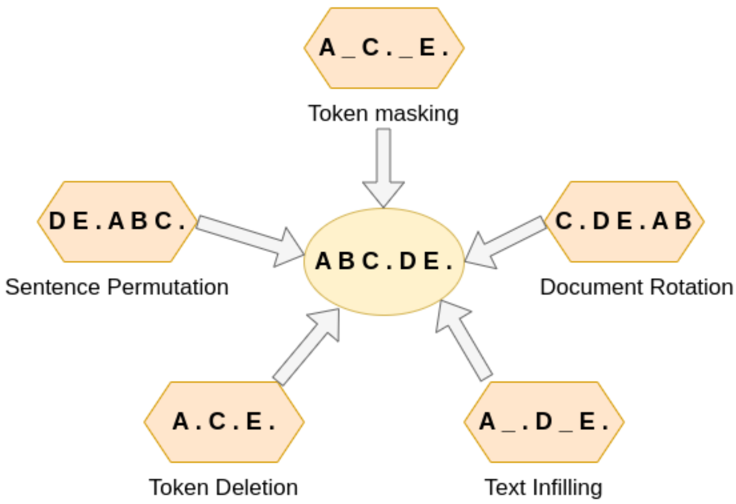


Figure 3.4: Noising schemes used for the input documents (Source [71]).

- **Token Deletion.** Random tokens are removed from the input sequence. In this scheme, the model needs to decide also the position of the missing inputs.
- **Text Infilling.** A number of spans of text are sampled, using span lengths drawn from a Poisson distribution ($\lambda = 3$). Similar to the first approach, the span of text is replaced with a `[MASK]` token. The goal is for the model to learn how many tokens are missing from a span.
- **Sentence Permutation.** After splitting the document into sentences, these sentences are shuffled in random order.
- **Document Rotation.** Given a randomly selected token, the document is rotated so that it begins with that token. The model is trained to identify the start of the document.

Fine-tuning. BART is able to compute representations that are highly adaptable for a variety of downstream tasks. For sequence classification tasks, the authors follow a similar approach to BERT, where they utilize the final hidden state of the final decoder token as an input to a linear classifier. For token classification tasks, the top hidden state of the decoder for each word is fed to the classifier. In addition, BART can also be used for sequence generation tasks due to its autoregressive decoder. In this case, the encoder encodes the input sequence and the decoder generates the outputs autoregressively.

3.2.5 T5

In addition to different architectures explored, Raffel et al. focus on empirical comparison of existing techniques [73], by utilizing a standard transformer model architecture. The core idea behind T5 is to treat every problem as a text-to-text task, and the model is referred to as "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer". The model is pre-trained using Common Crawl, a text scraped from the web, that was pre-processed and cleaned, to yield an $750GB$ amount of cleaned text. To evaluate the T5 model's general language understanding capabilities, they use a wide range of downstream tasks: Sentence

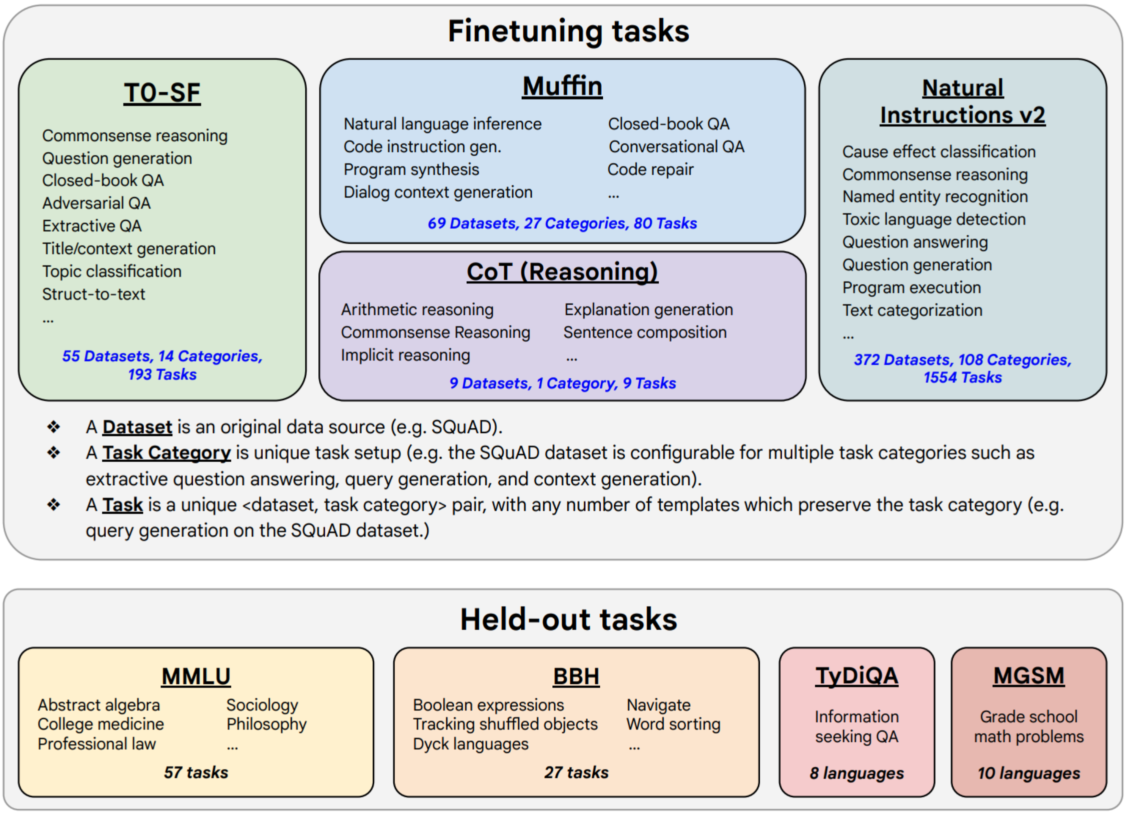


Figure 3.5: Tasks that were used to fine-tune FLANT-T5 (Source [72]).

acceptability judgment, sentiment analysis, paraphrasing, natural language inference, coreference resolution, sentence completion, word sense disambiguation, and question answering. In addition, each task is cast as a ”text-to-text” format, where the model generates an output sequence based on the context provided by the input sequence. The model is trained with maximum likelihood objective using teacher forcing [74] for all tasks.

Their base model is similar in size and configuration to BERT [4], resulting with about 220M parameters in total. Their largest model, scales up 11B parameters. Before fine-tuning, each model is pre-trained for $524,288$ steps on Common Crawl with a batch size of 128 and a maximum sequence length of 512. During pre-training, they use an inverse square root learning rate scheduler, with warm-up steps set to 10^4. They fine-tune their model for $262,144$ steps on all tasks using the same batch size and maximum length.

In this work, they utilize SentencePiece[1] [75], to encode text as WordPiece tokens [66]. This heuristic builds up a vocabulary of $32,000$ wordpieces, and was also trained to include other languages. In addition, the vocabulary is shared across both input and output of the model.

Unsupervised Objective. Different from GPT which utilizes a causal language modeling objective for pre-training, in this work, the authors utilize a masked language modeling objective. This is similar

[1] different from GPT that utilized BytePair encoding.

to BERT denoising objectives [4] and the word dropout regularization technique [61]. The approach drops 15% of randomly sampled tokens of the input sequence. Afterward, each consecutive span of tokens are replaced by a unique sentinel special token. The target is then set to predict all the dropped-out tokens.

FLAN-T5. Recently, it was also introduced FLAN-T5 [72], which enhances T5 capabilities. In this paper, they utilize instruction finetuning with a variety of instruction templates and task mixtures. An overview of finetuning tasks is given in Figure 3.5. The models trained, scale up to 540B parameters.

3.3 Graph Neural Networks

Graph Neural Networks (GNNs) have emerged as a powerful paradigm for learning from data represented as graphs, capturing the complex relationships and structures inherent in various domains. Research on graph data tasks with machine learning has gained increased popularity in the last few years due to the expressive power of graphs. They have been used in different fields such as natural science for physical systems [76, 77], protein-protein interaction networks [78], social science for social networks [79], citation networks [80], knowledge graphs [81], and many others [82]. The capability of GNNs to handle non-Euclidean data structures makes them particularly suited for tasks such as node classification, where nodes are categorized into classes; link prediction, which involves predicting the likelihood of a relationship between two nodes; and clustering, grouping nodes based on similarity or connectivity. Recently, deep learning-based approaches employ Graph Neural Networks (GNNs) [83] that operate on the graph domain to handle those tasks. Several GNN architectures have been proposed to work with graph data and employ different learning modules which we will see below.

3.3.1 Graph Convolutional Networks

Kipf et al. introduced an approach for semi-supervised learning on graph-structured data, and employ a variant of convolution neural networks as a learning module [84]. In their work, they propose a multi-layer Graph Convolutional Network (GCN), with the following layer-wise propagation rule:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{3.13}$$

where $\tilde{A} = A + I_N$ is the adjecency matrix of an undirected graph $\mathcal{G}$ with added self-connections and I_N is the identity matrix. In addition, $\tilde{D} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, $H^{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activations in the lth layer where $H^0 = X$. One can also write the update rule for a single node as:

$$h_i^{l+1} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \tilde{A}_{ij}W^l h_j^l\right) \tag{3.14}$$

where $\mathcal{N}(i)$ denotes the neighbourhood nodes of node i.

3.3.2 Graph Attention Networks

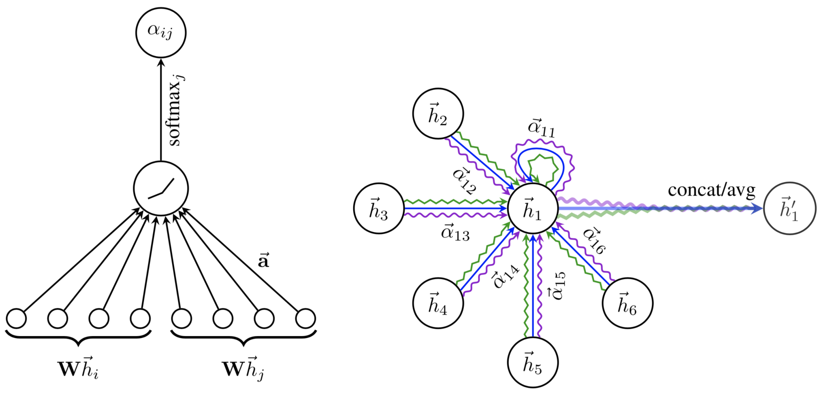


Figure 3.6: (left) The attention mechanism employed by GAT. (right) Multi-head attention in GAT (Source [36]).

A Graph Attention Network (GAT) module (see Figure 3.6) stacks layers in which nodes can attend over their neighborhoods' features. It implicitly allows defining different weights to different nodes in a neighborhood without requiring a costly matrix operation (e.g., inversion) or depending on knowing the graph structure upfront. Formally, a GAT computes the hidden states of each node by attending to its neighbors via a self-attention strategy. For a node v, the hidden state is obtained by:

$$\mathbf{h}_v^{t+1} = \rho\left(\sum_{u \in \mathcal{N}_v} \alpha_{vu} \boldsymbol{W} \mathbf{h}_u^t\right), \; \alpha_{vu} = \frac{exp(LeakyReLU(\mathbf{a}^T[\boldsymbol{W}\mathbf{h}_v \parallel \boldsymbol{W}\mathbf{h}_u]))}{\sum_{k \in \mathcal{N}_v} exp(LeakyReLU(\mathbf{a}^T[\boldsymbol{W}\mathbf{h}_v \parallel \boldsymbol{W}\mathbf{h}_k]))}, \tag{3.15}$$

where ρ is an alternative non-linear function, $\mathcal{N}_v$ is the neighborhood set of node v, $\mathbf{h}_u^t$ is the hidden state of neighborhood node u at time step t, $\boldsymbol{W}$ is the weight matrix associated with the linear transformation applied to each node, and a is the weight vector of a multi-layer perceptron (MLP).

3.3.3 Hyperbolic Graph Convolutional Networks

Liu et al. introduced Hyperbolic Graph Convolutional Networks (HGCNs) [85], to solve the problems with graph operations in non-Euclidian spaces. They generalize the notion of a graph convolutional network, such that the network can become agnostic to the underlying space. They update the propagation rule given in Equation 3.14 as follows:

$$h_i^{l+1} = \sigma\left(exp_{x'}(\sum_{j \in \mathcal{N}(i)} \tilde{A}_{ij} W^l log_{x'}(h_j^l))\right) \tag{3.16}$$

where $x' \in \mathcal{M}$ is a chosen point in the Riemman manifold $\mathcal{M}$.

HGNNs are shown to be more efficient in capturing structural properties of synthetic data than GNNs operating in Euclidean space. Moreover, they show to perform better for chemical properties of molecules, or predicting properties of large-scale networks by making use of the hierarchical structure present in the data.

3.4 Summary

In this chapter, we established the foundational concepts and the core architecture, necessary for understanding the context and technical aspects of our research. Initially, we offer an in-depth description of the transformer architecture, which currently forms the backbone of numerous NLP systems. Additionally, we provided a comprehensive discussion on pre-trained models for contextualized word representation such as BERT, SBERT, GPT, BART, T5. The chapter concludes with an overview of graph neural network architectures used during our work, such as Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and Hyperbolic Graph Convolutional Networks (HGCN). In the following chapter, we will investigate methodologies for incorporating additional authors' context, in order to enhance text classification tasks.

CHAPTER 4

Post Classification with Author Context

Human language is produced to fulfill specific goals in communication [86] and is influenced by various individual and social group characteristics [7, 8]. Hence, we hypothesize that incorporating additional context for individuals might improve text understanding performance for NLP systems.

This chapter delves into our first research question (RQ1):

> Research Question 1 (*RQ1*)
>
> Can we enhance text classification tasks by incorporating authors' context?

To address this research question, we focus on sarcasm detection task, on social media data. Such texts, usually, lack sufficient context, making it difficult for the systems to detect sarcasm. Moreover, social studies suggest that the relationship between the author and the audience can be equally relevant for sarcasm usage and interpretation. Hence, we propose a framework jointly leveraging (1) an author context from their historical tweets together with (2) the social information from an author's conversational neighborhood in an interaction graph, to contextualize the interpretation of the post. This approach aims to enrich the contextual understanding of posts, facilitating a more nuanced detection of sarcasm. Furthermore, this chapter discusses our efforts to develop and curate resources specifically designed for sarcasm detection tasks. These resources are constructed to support our author-centered methodology, encompassing both textual content and the essential contextual information needed to investigate the impact of incorporating authors' contexts into text classification.

The key contributions of this chapter are:

- We present the first graph attention-based model to identify sarcasm on social media by explicitly modeling authors' social and historical context jointly, capturing complex relations between a sarcastic tweet and its conversational context.
- We demonstrate that exploiting these relationships increases performance in the sarcasm detection task, reaching state-of-the-art results on the recent SPIRS dataset [87], which we expand with author history. We examine the impact of different parts of the context, captured by attention weights, in modeling sarcastic utterances.

- We collect a new dataset from Twitter (now X) by extending a semi-supervised method that uses reactive supervision and provides additional contextual information.

- We find that even with author-based models, detecting the sarcastic intentions of the author is easier than identifying the sarcasm perception by others.

This chapter is based on the following publication ([13, 14]):

- **Joan Plepi** and Lucie Flek. 2021. Perceived and Intended Sarcasm Detection with Graph Attention Networks. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4746–4753, Punta Cana, Dominican Republic. Association for Computational Linguistics. DOI: 10.18653/v1/2021.findings-emnlp.408

- **Joan Plepi**, Magdalena Buski, and Lucie Flek. 2023. Personalized Intended and Perceived Sarcasm Detection on Twitter. In Proceedings of the 3rd Workshop on Computational Linguistics for the Political and Social Sciences, pages 8–18, Ingolstadt, Germany. Association for Computational Lingustics.

The rest of the chapter is structured as follows. Section 4.1 introduces the work. In Section 4.2 we describe the related work. Furthermore, in Section 4.3 we introduce our dataset for this task, followed by our experimental setup and results in Section 4.4. Finally, Section 4.6 provides a summary for the chapter.

4.1 Introduction

Sarcasm is a form of non-literal language, in which the intended meaning of the utterance differs from the literal meaning, fulfilling a social function in a discourse [88, 89, 90]. Sarcasm detection poses a challenge for numerous NLP tasks, such as sentiment or stance prediction [91, 92]. Early sarcasm detection systems are based on lexical and syntactic cues [93, 94, 95, 96, 97]. However, sarcasm interpretation requires the surrounding context of utterances, even for humans [98]. More recent works hence incorporate discourse information such as contrast [90, 99, 100, 101], and contextualize the post by using features from user history [102, 103, 104, 105]. The relationship between an author and the audience has been given comparably less attention, despite its relevance for the sarcasm interpretation [106, 107, 108, 109, 102]. For example, since background information about an author's topic of interest can cause different interpretation of an utterance, authors are more likely to use sarcasm among close friends [107].

To address these complexities, our research introduces a novel approach: the development of a heterogeneous graph structure. This structure integrates user and text nodes, connected by a network of edges that capture the relationships between them. Furthermore, we propose a graph neural network framework that learns the representation of this heterogeneous graph. This framework jointly combines user context, derived from their historical tweets, with social information from the user's network. By doing so, it allows for a more nuanced and comprehensive understanding of sarcasm, accounting for both the content of the utterance and the rich social dynamics that support its interpretation.

4.2 Related Work

In this work, we refer to users as social media users. The paragraphs below summarize the related works, in sarcasm detection area. The main focus is on related datasets or tasks, and the methodology used in previous works.

Collection and Labeling of Sarcastic Data

Previous approaches to data collection for automatic sarcasm detection can be divided into two groups: distant supervision and manual annotation [110, 111]. One approach requires annotators to manually label whether a given utterance is sarcastic or not [112], while distant supervision focuses on automatically collecting large datasets of intended sarcasm. The automatic data collection uses specific keywords to query social networks [113, 114, 115, 116]. Nevertheless, the subjectivity and sociocultural dependence of perceived sarcasm [117, 118] often lead to discrepancies between intended and perceived sarcasm. Recent approaches have addressed this issue by generating datasets for automatic sarcasm detection that reflect this discrepancy. For example, the iSarcasm dataset [119] manually collects and labels sarcastic utterances by their authors, instead of relying on third-party annotators. However, this dataset only contains 777 sarcastic tweets and does not include perceived sarcasm. In contrast, the SPIRS dataset [87] utilizes reactive supervision to collect both intended and perceived sarcasm. The dataset consists of 30k tweets and relies on cues from participants in online conversations, therefore using context-aware annotations.

Leveraging User Information for Sarcasm Detection

Several previous works contextualize a sarcastic post by using features from user history - employing past tweets to identify a user's behavioral traits [120], encoding user sentiment priors over different entities [121], or manually crafting user interaction features [102]. Amir et al. introduce the user2vec model [39], applying paragraph2vec [122] over user history. Hazarika et al. propose an alternative user embedding approach, encoding style, and personality features [123].

An emerging line of research makes use of social interactions to derive information about the user - representing each user as a node in a social graph and creating low-dimensional user embeddings induced by neural architecture [124, 125]. Including network information improves performance in detecting online behavior such as cyberbullying [126], abusive language use [127], suicide ideation [34] or fake news detection [128]. To our knowledge, GNNs were not used in the sarcasm classification tasks.

Due to a lack of consistency in sarcasm interpretation by people of different socio-cultural backgrounds, an utterance that is intended as sarcastic by its author might not be perceived as such by diverse audiences (Rockwell and Theriot, [108]. Recently, this has been reflected in sarcasm detection interpretations [123, 87] and a resulting sarcasm perspective classification task.

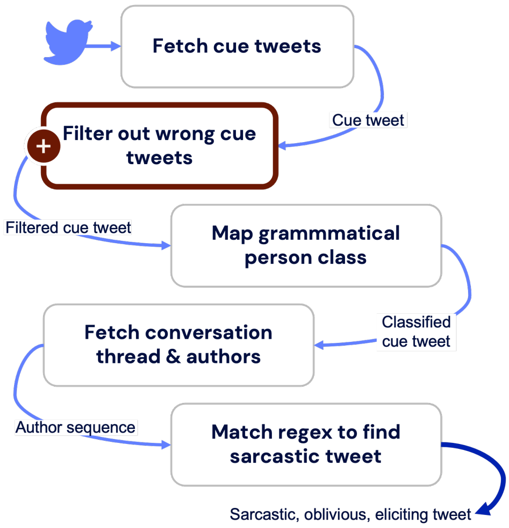


Figure 4.1: 5-step pipeline of enhanced reactive supervision.

4.3 Dataset

4.3.1 Reactive Supervision

For our experiments, we use a recently published SPIRS sarcasm dataset [87]. The dataset is crawled from Twitter, by utilizing a new reactive supervision method. The method utilizes *cue tweets*, conversation replies which point out the sarcastic nature of a previous post. In addition, the dataset also provides *oblivious tweets*, questioning the sarcastic nature of a given example, and *elicit tweets*, being the original start of the conversation. This approach consists of four steps as below:

1. Fetching cue tweets q_n, querying for tweets containing "being sarcastic"

2. Mapping the cue tweets to a grammatical person class (1st, 2nd, 3rd) by examining the personal subject pronoun in the cue tweet

3. For a cue tweet q_i, fetching the corresponding conversation $C^i = \{c_n, ..., c_1\}$, where c_n is the main post, $c_1 = q_i$ and the corresponding tweet author sequence $A^i = \{a_n, a_{n-1}, ..., a_1\}$

4. Applying specific regular expressions on the author sequence to identify the sarcastic tweet. Unmatched sequences are discarded and matched are saved along with the cue tweet, the eliciting (occurring if the sarcastic tweet is a reply and represents tweets that evoked the sarcastic reply), and the oblivious tweets (a reply to the sarcastic tweet that lacks awareness of sarcasm).

The labeled dataset contains in total 15,000 sarcastic tweets (10,000 self-reported and 5000 perceived cues), 15,000 non-sarcastic posts that were collected randomly, 10,000 oblivious, and 9156 elicit tweets. We extended SPIRS with over 10 million past tweets of the authors in the dataset in order to compute the user embeddings.

Regular expressions
r"not being sarcastic"
r"not(\s*[A-Za-z,;'\\\/s@])* \s*sarcastic") r"(sarcastic)\s*(\?)+"
r"wasn't being sarcastic"
r"wasnt being sarcastic"
r"wasn't being sarcastic"
r"was not being sarcastic"
r"weren't being sarcastic"
r"weren't being sarcastic"
r"werent being sarcastic"
r"were not being sarcastic"
r"(sarcastic)\s*(\?)+"
r"sarcastic\sor"
r"hope(\s*[A-Za-z,;'\\s@])*\s*being sarcastic"
r"hope(\s*[A-Za-z,;'\\s@])*\s*being(\s*[A-Za-z,;'\\s@])*\s*sarcastic"
r"hope you're being sarcastic"
r"pray(\s*[A-Za-z,;'\\s@])*\s*being sarcastic"
r"if(\s*[A-Za-z,;'\\s@])*\s*being sarcastic"
r"sarcastic[A-Za-z,;'\\s@]*\s*correct"
r"sarcastic\s*([A-Za-z,;'\\s@]\s)0,2 right"
r"are you being sarcastic"

Table 4.1: Compound regular expression used to filter tweets incorrectly identified as cue tweets.

4.3.2 Enhanced Reactive Supervision

In addition to SPIRS dataset, we introduce a new dataset crawled from Twitter, utilizing an enhanced version of reactive supervision. After manual analysis of random data points in the dataset [87], we found that the proposed approach can mistakenly label certain non-sarcastic tweets as sarcastic. We discovered several cue tweets containing "being sarcastic" which are noisy reactions from the audience, which express doubt, or ask for clarification for example: "*@user I can't tell if you are being sarcastic*". To create a dataset excluding those falsely classified tweets we propose an extension of the reactive supervision method. We add another filter (Figure 4.1), to remove tweets falsely identified as cue tweets using regular expressions, hence improving the quality of the extracted data. The filter contains a series of regular expressions to clear out the false positive cue tweets. We show a list of these regular expressions in Table 4.1. The main target class that was fixed from the regular expressions, was the perceived sarcasm, where the number of false positive rates was significantly reduced. In order to compare both methods, we collected 500 random cue tweets, which we labeled manually into three classes: sarcastic, non-sarcastic, and unknown (the user is asking for clarification, rather than pointing out sarcasm). Given the cue tweet and the conversation, we annotated the examples into three categories: sarcastic, non-sarcastic, and unknown. Fleiss' Kappa inter-annotator agreement between two annotators was almost a perfect agreement, with a kappa value of 0.94. Upon manual inspection and discussion, we found that the cases where the annotators were disagreeing were mainly between classes unknown and sarcastic (possible perceived sarcasm), where the user was expressing doubts if the previous tweet was sarcastic or not. Hence, we were able to resolve the disagreements through a deeper inspection of the conversation thread. In Table 4.2 we show the number of tweets filtered out as sarcastic from both methods and also the false positive rate (we treat unknown and non-sarcastic as a single category). We observed that the number of filtered sarcastic tweets increased while the rate of false positive examples decreased from 46.6% to 5%.

Cue tweet indication	Gold	4-step	5-step
Sarcastic	318	24	109
Non-sarcastic	182	21	6
Total	500	45	115

Table 4.2: Comparison of the 4- and the 5-step data collection pipeline.

We applied our method (Figure 4.1) on a large scale to collect a dataset for sarcasm detection. For the collection of cue tweets, we queried for English tweets containing ”being sarcastic”, which are not retweets and were generated in the period from January until November 2022. For the collection of non-sarcastic tweets, we chose to fetch tweets randomly, querying for English tweets that have been generated from January until November 2022, are not retweets, and don't contain the words ”sarcastic”, ”sarcasm” or the tags ”#sarcasticquote”, ”#sarcasticquotes”, ”#sarcasticmemes”, ”#sarcastic”, ”#sarcasm”. Finally, we gathered 17k English sarcastic tweets and 19k non-sarcastic tweets with corresponding additional conversational contexts such as oblivious or elicit tweets (a tweet that caused the sarcastic reply). In addition, we collected around 89M historical tweets for the users in our dataset in order to extend the dataset with additional author contextual information.

Statistics We collected 100K cue tweets for the new dataset. In Table 4.3 we present examples of the cue tweet for each grammatical person class. Next, we applied the exclusive filter, filtering out 26.6% of the cue tweets. After collecting the threads, and corresponding authors for the remaining cue tweets and matching those author sequences, we end up with 17k English sarcastic tweets, 10k eliciting, and 13k oblivious tweets. In addition, we collected 19k non-sarcastic tweets as well as 11k corresponding eliciting and 4k oblivious tweets. We summarize the new dataset grouped by grammatical person classes and perspectives in Table 4.4, and with the statistics of user history in Table 4.5.

In Table 4.6 we examine the distribution of different author sequence patterns of the sarcastic threads. We observed that 80% of the threads are equal to or smaller than 4 tweets per thread. In addition, it shows the most common author thread pattern per grammatical-person class, indicating

Person	Perspective	Cue Tweet
1st	Intended	@user @user **I was being sarcastic**. That is what they tried to spin after the Nazi speech.
2nd	Perceived	@user I know **you are being sarcastic** btw. I just figure answering honestly is the best policy.
3rd	Perceived	@user @user Do you not see how many repeats there are? **He's being sarcastic**.

Table 4.3: Exemplary cue tweets per grammatical person class.

Pers.	**Perspective**	**Sarcastic**	**Oblivious**	**Eliciting**
1st	Intended	12574	12574	9023
2nd	Perceived	3295	0	519
3rd	Perceived	846	846	120
–	Non-sarc.	18535	4346	10639
Total		**35250**	**17766**	**20301**

Table 4.4: Break down by grammatical person class and perspective of our new dataset.

that sarcastic tweets are often provoked by other authors (see eliciting tweets). Moreover, we notice the patterns used to detect perceived sarcasm, grouped in 2nd and 3rd person perspective cues. These cues capture conversations where other participants detect the presence of sarcasm.

During our analysis of the most common bi-grams in the dataset, we noticed that political or politician-related bi-grams predominated within the perceived sarcasm class (Figure 4.2). This finding reinforces the link between sarcasm and political discourse [129, 116], offering insights into the potential significance of the detection of (perceived) sarcasm in understanding the political stance and the presence of this linguistic phenomenon in online interactions.

Class/Perspective	**# Authors**	**# Historical tweets**
Sarcastic	**15884**	**45244265**
Intended	12245	33328130
Perceived	3686	12257193
Both	47	–
Non-sarcastic	**17340**	**43475563**
Both	**99**	**–**
Total	**33125**	88719828

Table 4.5: Break down of the number of tweet authors by class and perspective.

Historical tweets The 35k sarcastic and non-sarcastic tweets of our new dataset have been composed by 33k different authors. Along with the new dataset, we collected 89M historical tweets for those 32k authors (Table 4.5). The number of historical tweets per author varies between 1 (16 authors have 1 historical tweet) and 500 (upper bound) with an average tweet number of 471.46.

4.4 Experimental Setup

Our experiments are focused on two main tasks: sarcasm detection to predict if a tweet is sarcastic or not, and perspective classification to predict if a sarcastic tweet is intended or perceived. We perform our experiments in two datasets: SPIRS that is collected utilizing reactive supervision, and our new dataset which utilizes enhanced supervision, consisting of 35K tweets. Two type of models are used: a) text-only-based models, and b) author-contextual-based models.

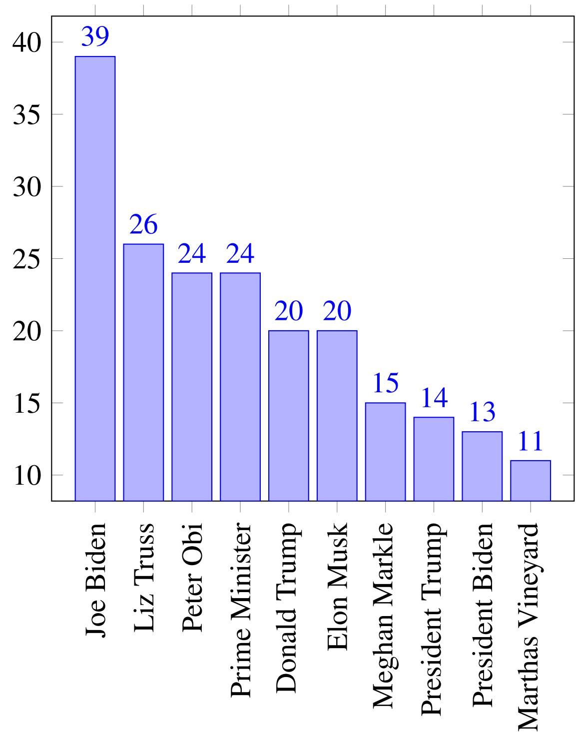


Figure 4.2: Top 10 most common bi-grams in as sarcastic perceived tweets.

4.4.1 Text-only-based models

This model only uses a representation of the textual information in the sarcastic and non-sarcastic tweets as input. For this purpose, we fine-tuned the pre-trained Transformer encoder like SentenceBERT [44] on the binary task of predicting the label sarcastic vs. non-sarcastic or perceived vs. intended, given only the tweet text. In this setup, we are also able to append the conversational context, namely oblivious and elicit tweet [1], in case those exist. We do so by appending the conversational context with the tweet that is to be classified, and we use special tokens to separate those (as in Figure 4.3).

4.4.2 Author-based-models

These models expand the textual features of tweets by adding representations of the authors of tweets as features. For encoding user representations, we used different models, namely: a) Priming, b) Average SentenceBERT for authors (A-SBERT), c) Authorship Attribution (AA) d) Graph Neural Networks (GNN) described in Section 2.3.1.

Priming: This model is the same as the base model, but the input text is different. The SBERT base model is finetuned to the binary task of predicting the verdict, given the situation title, the sampled text from each annotator, and the comment in the interpretation model case.

Average SentenceBERT for authors (A-SBERT): We finetune the SBERT base model and add an additional layer to concatenate the text representations with annotators' representations, using the

[1] Cue tweets are not part of the conversational context.

Person	Pattern	Count	% of person class
1st	*ABAC*	3368	27%
(intended)	*ABA*	2795	22%
	ABAB	1918	15%
	other	4493	36%
Subtotal		12574	
2nd	*AB*	2679	82%
(perceived)	*ABA*	476	14%
	other	140	4%
Subtotal		3295	
3rd	*ABC*	621	73%
(perceived)	*ABCA*	54	6%
	other	171	20%
Subtotal		846	
Total		16715	

Table 4.6: Most common thread pattern by person class. The colors represent red-cue, blue-oblivious, violet-sarcastic, and teal-eliciting tweets. The shown letters correspond to different authors in the thread. Equal letters encode equal authors, and the author sequences are shown in reverse order. The rightmost letter represents the end of the thread (cue tweet) while the leftmost represents the beginning of the thread.

initial annotator representations computed from SBERT for Annotators (see Figure 2.4).

Author Attribution: In this setup, we have the same architecture as averaging embeddings, however, the initial annotator representations are generated using the author attribution model (see Figure 2.4).

User+tweet GAT: Constructs a heterogeneous graph as described in Section 2.3.1, and uses a Graph Attention Network, to learn the relationship between tweet and user nodes (see Figure 4.4).

	Sarcasm Detection			
	SPIRS		E - Supervision	
Model	Accuracy	F1	Accuracy	F1
SBERT	65.9%	66.2%	74.4%	74.5%
Priming	68.2%	68.2%	77.5%	77.7%
Authorship Attribution	70.1%	70.1%	79.3%	79.3%
A-SBERT	69.6%	69.6%	80.1%	80.1%
User+tweet GAT	**80.9%**	**81.1%**	**82.0%**	**82.2%**

Table 4.7: Accuracy and macro F1-scores as percentages for the sarcasm detection task.

4.5 Results and Analysis

Our initial experiments focused on the task of sarcasm detection, and we show the results in Table 4.7. As also seen in previous works [102, 39, 13], author-contextual-based models outperform text-based

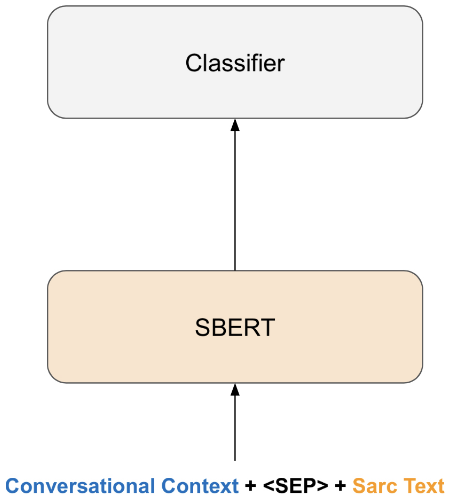


Figure 4.3: For the conversational context, we still use SBERT model as our base model. We only append the conversational context (namely oblivious and elicit tweet) to the original tweet to be classified and separated with special tokens.

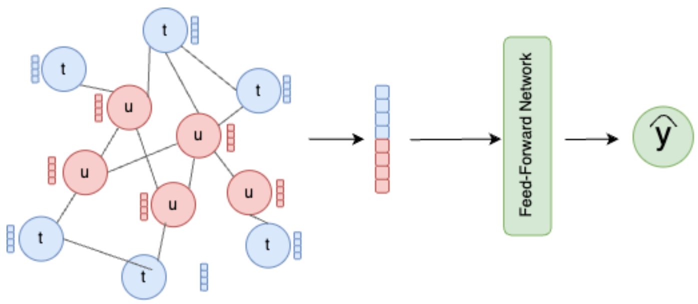


Figure 4.4: The social graph is initialized with user and tweet embeddings, and tuned by GAT to take into account relationships between them. The output representations are then fed into the classification layer.

models. The additional context from the author's representations enriches the text features and enhances its performance on the task of sarcasm detection.

Comparison across datasets. In Table 4.7, we report the results in both datasets, SPIRS and our dataset E-Supervision. We notice that our models are performing better in the E-Supervision dataset, and we believe that the main reason is due to the quality of the data. The decrease of the false positive rate in the data annotation process has improved the data quality, hence improving the model's training and performance.

Author-based-models. All models detecting sarcasm, given the tweet and author embedding, yield higher results. Our results' analysis revealed that the GNN-based model (User+tweet GAT) is our best-performing one in both datasets. Modeling social network interactions as graphs proves to be an effective way to learn better representations for both text and users. However, it seems that the GNN based model has the biggest improvement of 14.9% over the baseline model on the SPIRS dataset. On the other hand, the priming method performs the worst from the author-contextual-based models with only 3.2% improvement from the text-only SBERT model. Furthermore, author attribution performed slightly worse than A-SBERT, mainly due to sparsity in AA representation. Another limitation of AA is its scaling over more authors. Overall, GNN and A-SBERT proved to be the most effective in terms of both performance and computational costs, due to no additional training for computing the author representation.

Conversation context For comparison, we construct three social graphs where: 1) We remove the elicit tweets that triggered the sarcastic comment (GAT - elicit tweets), 2) We remove the oblivious tweets that interpreted the comment as serious (GAT - oblivious tweets), 3) We add the original cue tweets, revealing that the post was sarcastic (GAT + cue tweets). We report the results in Table 4.8. As expected, adding the cue tweets in the social graph leads to an almost perfect F1 score of 94.5% and 93.8% for both datasets. This is attributed to the fact that the GAT network learns to recognize that the cue tweets point out the sarcastic nature of the tweet to be assessed and give high importance to their existence in the graph. Therefore, by introducing the connection between the cue tweet and the tweet to be assessed, the network can easily distinguish between sarcastic and non-sarcastic tweets. Removing oblivious and elicit tweets causes just a small performance drop (2-3%). In the way the SPIRS dataset is annotated, an oblivious tweet typically triggers a cue tweet ("*c mon, dude, it was just sarcasm*"). We hypothesize that even with the cue tweets removed, the model is able to learn the predictive relation between oblivious and sarcastic tweets. This is in line with the original paper (i.e. without user context), where a 3.4% drop in prediction accuracy was observed when the oblivious tweets were removed.

Sarcasm Detection				
	SPIRS		E - Supervision	
Model	**Accuracy**	**F1**	**Accuracy**	**F1**
User+tweet GAT+CC (no cues)	84.7%	84.2%	83.0%	83.5%
User+tweet GAT, no elicit	81.8%	82.0%	80.4%	80.5%
User+tweet GAT, no oblivious	81.5%	81.4%	79.5%	79.6%
User+tweet GAT	80.8%	81.1%	82.0%	82.2%
User+tweet GAT + CC	94.3%	94.5%	93.9%	93.8%

Table 4.8: Sarcasm detection results for different combinations of edges between users, tweets, and conversation context (CC), with and without cue tweets, in order to see the effect of each edge type.

Sarcasm perception. We observe that in the sarcasm detection task, the error rate on perceived sarcasm is higher than in self-reported sarcasm. We therefore test our model on distinguishing between perceived and self-reported sarcasm. Our results for this task are shown in Table 4.9. We notice a lower improvement of at most only 4.0%, of author-contextual-based models over the SBERT model compared to the sarcasm detection task. These results also align with the conclusion in [119], on the perception classification task. In most cases, the perceived sarcasm is misclassified as self-reported, which is present more often (70%) in the data (see Table 4.10). Hence, we believe that modeling the representation of the author is less useful for the classification of perceived sarcasm. To increase the number of tweets classified as perceived, it could be of benefit to additionally model user embeddings

Sarcasm Perception				
	SPIRS		E - Supervision	
Model	**Accuracy**	**F1**	**Accuracy**	**F1**
SBERT	73.2%	67.3%	79.2%	68.5%
Priming	74.1%	69.1%	79.8%	70.9%
Authorship Attribution	76.6%	**71.3%**	**82.2%**	71.3%
A-SBERT	75.3%	70.3%	79.2%	70.6%
User+tweet GAT	76.0%	71.2%	80.8%	**72.2%**

Table 4.9: Accuracy and macro F1-scores as percentages for perspective classification.

for the audience of the tweet, predicting how individual users will react towards the tweet. Improving the quality and quantity of perceived sarcasm remains a challenging task, given its subjective nature that is often influenced by the audience's diverse social and cultural backgrounds, which may influence their interpretation of tweets on a certain topic.

Model	$\mathbf{W_P}$	$\mathbf{W_I}$
SBERT	59.1	7.5
Priming	50.9	9.9
A-SBERT	49.2	11.4
Authorship Attribution	58.5	4.5
User+tweet GAT	51.4	7.8

Table 4.10: False predicted sarcastic perspectives as percentages in relation to gold labels in the E-supervision dataset for all models used. W_P is the percentage of perceived tweets falsely classified as intended; W_I, the percentage of intended tweets falsely classified as perceived. Number of test instances: 3343 tweets.

4.6 Summary

In this chapter, we addressed the first research question, by exploring social networks of user interactions, and contextual information to interpret sarcastic intentions in social media.

Initially, in Section 4.3, we provide an overview of the reactive supervision method that is used to collect SPIRS dataset [87]. Additionally, we provided an extension to this dataset, by including user historical data. Furthermore, we introduced a new dataset crawled from Twitter, E-Supervision, utilizing an enhanced version of the reactive supervision method. This new method aimed to improve the quality of sarcastic data, by lowering the amount of false positives. The dataset consists of around 35K sarcastic and non-sarcastic texts, along with their conversational context such as cue, oblivious, and elicit tweets, and around 89M historical tweets in order to extend the dataset with additional author contextual information.

In Section 4.4, we outlined our experimental setup and described both tasks that we are focused on: sarcasm detection and perspective classification. In our experiments, we employed two types of models, text-only-based models, and our proposed author-based-models like priming, A-SBERT, Authorship Attribution, and User + Tweet GAT.

Our findings indicate a better performance of our models in the E-Supervision dataset compared to the SPIRS dataset, which we attribute to the improved quality of the E-Supervision dataset. We find that all our author-based-models yield higher results compared to text-only-based-models. Our results' analysis revealed that the GNN-based model (User+tweet GAT) is our best-performing model in both datasets, underscoring the importance of modeling social network interactions as graphs, jointly with text, as an effective way to learn better representations for both text and users. However, it was observed that modeling authors were less impactful in the perspective classification task. We noticed a lower improvement in author-based-models over the text-only-based models compared to the sarcasm detection task. This suggests that, for tasks like perspective classification, focusing on modeling the text recipients to understand their perspectives might be more advantageous. In the next chapter, we will explore such an approach, by focusing on modeling the recipients of texts to better capture their perspectives.

CHAPTER 5

Perspective Classification with User Context

Following our analysis of the enhancement of text classification tasks through the incorporation of authors' context, we observed a significant impact on interpreting authors' sarcastic intentions. This finding underscores the importance of context in understanding the nuances of communication. However, our analysis also revealed a limitation: while authors' context aids in identifying sarcasm from the perspective of the author, it offers less insight into how sarcasm is perceived by recipients. This observation leads us to shift our focus towards the recipients of communication, addressing an important aspect of discourse analysis.

This chapter delves into the second research question:

Research Question 2 (*RQ2*)

How can we model the context of recipients to accurately predict their responses to various discourses?

In the domain of natural language processing, tasks such as detecting toxicity, offensiveness, and sarcasm present inherent challenges due to the subjective nature of text interpretation. These challenges are further extended by the importance of grasping social norms to accurately understand people's actions and intentions—a task that is not only important for humans but increasingly for artificial agents as well. Additionally, perceptions of what is socially acceptable behavior is subjective and issues are often divisive [130]. Given the diversity in social-cultural backgrounds, feelings, thoughts, experiences, and perspectives, interpretations of text can vary significantly among individuals [131]. In such subjective tasks, different people will interpret text in different ways. Hence, the additional context for the author who writes it and the audience who reads it becomes important to understand the meaning of an utterance. Therefore, this chapter aims to explore personalized models that capture the audience's views and reactions across a spectrum of topics. We will examine the performance of these models under various experimental conditions, across different topics, and among diverse demographic groups. By focusing on the recipients' context, we aim to delve deeper into the challenges of discourse interpretation, analyzing how differing perspectives on social norms influence the understanding of text.

The key contributions of this chapter are:

- We utilize Reddit community to construct a dataset that contains individual assessments of conflict situations. Additionally, this dataset includes: 1) Clustering descriptions of social situations involving interpersonal conflict 2) a set of 500 conflicts annotated with six aspects of conflict 3) Annotators context, including their past history together with their demographics.
- We address the task of predicting whether someone will perceive the actions of one individual as right or wrong in a given situation over different degrees of conflict.
- A discussion of the relation between data perspectivism and personalization
- We introduce and adapt five different methodologies to encode users' context, and perform a novel comparison and analysis of contemporary over a novel problem setting

This chapter is based on the following publication ([16, 15]):

- Charles Welch, **Joan Plepi**, Béla Neuendorf, and Lucie Flek. 2022. Understanding Interpersonal Conflict Types and their Impact on Perception Classification. In Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS), pages 79–88, Abu Dhabi, UAE. Association for Computational Linguistics. 10.18653/v1/2022.nlpcss-1.10
- **Joan Plepi**, Béla Neuendorf, Lucie Flek, and Charles Welch. 2022. Unifying Data Perspectivism and Personalization: An Application to Social Norms. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7391–7402, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. DOI: 10.18653/v1/2022.emnlp-main.500

The rest of the chapter is structured as follows. Section 5.1 introduces the work, and we describe the related work in Section 5.2. Following related work, we introduce our dataset in Section 5.3. In Sections 5.4 and 5.5, we show our experimental setups for different tasks, together with the results. In Section 5.6, we provide an analysis over our results. Finally, in Section 5.7 we provide a summary for the chapter.

5.1 Introduction

Understanding social norms is critical to understanding people's actions and intents, not only for humans, but also for artificial agents. The inability for artificial agents to take these norms into account may serve as a barrier to their ability to interact with humans [132]. However, perceptions of what is socially acceptable behavior vary and issues are often divisive [130]. It is critical to model these differences both to build higher performing systems and better understand people [20, 133].

In the NLP field, similar to sarcasm perception, we have the subjectivity of annotators in the data annotation field. More specifically, obtaining a single ground truth is not possible or necessary for subjective natural language classification tasks [133]. Each annotator is a person with their feelings, thoughts, experiences, and perspectives [131]. Researchers have been calling for the release of data without an aggregated ground truth, and for evaluation that takes individual perspectives into account [20]. The idea that each annotator has their view of subjective tasks, and even those previously

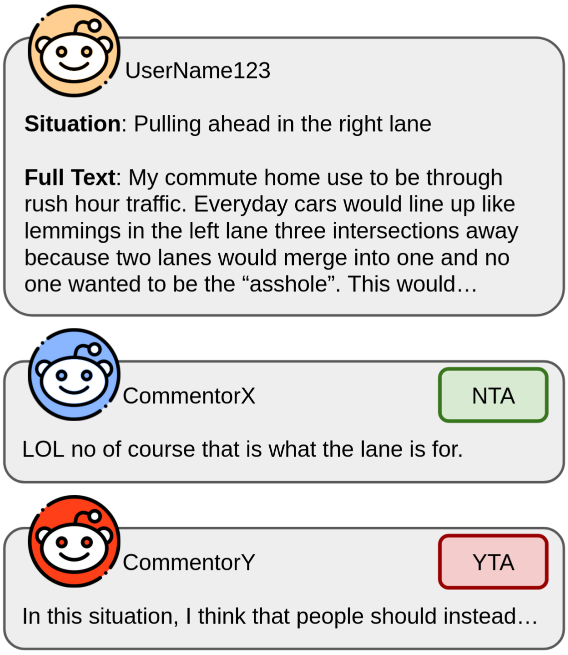


Figure 5.1: Example of a post on Reddit and two comments. The post has the situation, which comes from the post title and the full text of the post (truncated here). Usernames appear next to the icons of the poster and commentators. Each comment has a verdict, which is the label they assign (YTA or NTA).

thought to be objective, was introduced by [134] as *data perspectivism*. A growth in the interest of this viewpoint has led to the 1st Workshop on Perspectivist Approaches to NLP in 2022. Work has examined how to model annotators for subjective tasks and to predict each annotator's label [135, 136]. Modeling annotator perspectives requires the release of corpora that include annotator-level labels rather than aggregated "ground truth" labels. Bender et al. [137] further recommend releasing data statements that describe characteristics, including who is represented in the data and the demographics of annotators. Such information is beneficial for raising awareness of the biases in our data. While some corpora contain this information, like those for humor, emotion recognition, and hateful or offensive language, they contain few annotators and no additional information about them [138, 139, 140].

An additional complication for subjective tasks is the fact that different people will interpret text in different ways. What is deemed toxic or offensive depends on who you ask [141, 142]. There are notable differences in perceived and intended sarcasm [104, 13]. How one perceives the receptiveness of their own text is different from how others see it [143]. For such tasks, predicting the label given by third-party annotators, without knowing much about them, is not very useful. Modeling annotators with personalization methods requires a corpus with many self-reported labels from many annotators and additional contextual information about them.

In this chapter, we use English textual data in the form of posts from the website, Reddit, about social norms from the subreddit `/r/amitheasshole` (AITA). As shown in Figure 5.1, users of this online community post descriptions of situations, often involving interpersonal conflict, and ask other users to judge whether the user acted wrongly in the situation or not. The judgments from these users constitute our labels, and their authors are the set of annotators (and we refer to them as such for the remainder of the paper), which allows us to explore methods to model annotators at a larger scale. First, we explore methods of clustering descriptions of social situations involving interpersonal conflict and perform a human evaluation and analysis. After proposing a novel annotation scheme, we annotate a set of 500 conflicts with six aspects of conflict. Aspects and clusters are then used to provide an analysis of our model performance. Furthermore, we also explore methods of personalization to model these annotators and examine how the effectiveness of our approach varies with the social relation between the poster and others in the described situation. We further provide an analysis of how personalization affects demographic groups and how performance varies across individuals.

5.2 Related Work

5.2.1 Social Norms

Lourie et al. [130] looked at the AITA subreddit to model judgments of social norms. They looked at how to predict the distribution of judgments for a given situation, which indicates how controversial a situation may be. [144] expanded on this study by using their data to extract a corpus of rules-of-thumb for social norms. We examine a new dataset, created from the posts in their data but including the set of comments, which include annotators, their label, and the accompanied comment [15].

Efstathiadis et al. [145] examined the classification of verdicts at both the post and comment levels, finding that posts were more difficult to classify. Botzer et al. [146] also constructed a classifier to predict the verdict given the text from a comment and used it to study the behavior of users in different subreddits. De Candia [147] found that the subreddits where a user has previously posted can help predict how they will assign judgments. The author manually categorized posts into five categories: family, friendships, work, society, and romantic relationships. They found that posts about society, defined as “any situation concerning politics, racism or gender questions,” were the most controversial. Several works have also looked at the demographic factors or framing of posts affect received judgements [148, 147, 146].

5.2.2 Interpersonal Conflict

Distinctions between conflicts can be made based on who is involved. Intrapersonal occurs within oneself, while interpersonal occurs between individuals. Conflict with more people can occur within or across groups or organizations. Much research on the topic has focused on work goals and differentiates between task-related issues and those that result from differences in personality, values, or style [149]. This work has found it useful to distinguish between conflicts concerning interpersonal incompatibilities and those that arise from the content of a task being performed [150]. Further types have been introduced, though meta-analyses have found these types to be highly correlated, and thus researchers have called for improvements to how conflict is conceptualized and measured [151, 152, 153].

Barki et al. [154] surveyed work on interpersonal conflict and noted that studies focused on three common attributes: disagreement, negative emotion, and interference, which correspond to cognitions, emotions, and behaviors respectively. They suggest that these aspects vary across situations and that it is important to specify the target of the conflict. They define interpersonal conflict as “a dynamic process that occurs between interdependent parties as they experience negative emotional reactions to perceived disagreements and interference with the attainment of their goals." As this suggests, *conflict is about perception* [155].

5.2.3 Personalization

Many different approaches and tasks have used some form of personalization. These methods use demographic factors [25], personality traits [28], extra-linguistic information that could include context, or community factors [102], or previously written text. A similarity between personalization and annotator modeling is that the most common approach appears to be using author IDs. These have been used, for instance, in sentiment analysis [156], sarcasm detection [30], and query auto-completion [157].

King and Cook[42] evaluated methods of personalized language modeling, including priming, interpolation, and fine-tuning of n-gram and neural language models. [158] modeled users by predicting their behaviors online. Similarly, one's use of language can be viewed as a behavior. Welch et al. [159] modeled users by learning separate embedding matrices for each user in a shared embedding space. Welch et al. [160] explored how to model users based on their similarity to others. They used the perplexity of personalized models and the predictions of an authorship attribution classifier to generate user representations. In social media in particular, a community graph structure can be used to model relationships between users and their linguistic patterns [33].

5.2.4 Annotator Disagreement

There has been a shift in thinking about annotator disagreement as positive rather than negative [161]. Disagreement between annotators is often resolved through majority voting [162]. In some cases, label averaging can be used [163], or disagreements can be resolved through adjudication [164]. Majority voting, which is most often used, takes away the voice of underrepresented groups in a set of annotators, for instance, older crowd workers [165], and aggregation in general obscures the causes of lower model performance and removes the perspectives of certain sociodemographic groups [166]. On the other hand, [167] uses the annotator's identifiers as features to improve model performance while training. They note that annotator bias is a factor that needs additional thought when creating a dataset.

Fornaciari et al. [136] predict soft-labels for each annotator to model disagreement which mitigates overfitting and improves performance on aggregated labels across tasks, including less subjective tasks like part-of-speech tagging. Davani et al. [168] developed a multi-task model to predict all annotator's judgments, finding that this achieves similar or better performance to models trained on majority vote labels. They note that a model that predicts multiple labels can also be used to measure uncertainty. They experiment with two datasets, which have fewer than a hundred annotators each. This allows them to model all annotators, though they note that training their model on corpora with thousands of annotators, like ours, is not computationally viable.

Most work model annotators use their ID only. Basile et al. [169] has called for extra information about annotators to be taken into account. Some annotation tasks have collected demographic information about annotators, for instance [141], or used the confidence of annotators as extra information [170].

Conflict Aspect	MCC
Disagreement Strength	0.39 → 0.49
Emotion Intensity	0.33 → 0.41
Interference Degree	0.13 → 0.20
Conflict Duration	0.39
Manifestation or Perception	0.10
Number of People	0.40

Table 5.1: Annotator agreement using Matthews correlation coefficient for all six aspects. For non-binary aspects, the improvement after merging labels is shown to the right of the →.

5.3 Dataset

We collected data from Reddit, an online platform with many separate, focused communities called subreddits. In particular, we use data from the AITA subreddit, where members post a description of a social situation involving an interpersonal conflict and ask other members of the subreddit if they think the author of the post is the wrongdoer in the situation or not. Others will respond saying "you're the asshole" (YTA), or "not the asshole" (NTA). As an initial source to crawl the comments, we use the posts from [144]. We crawl the post title together with its full text, and all the comments that contain a verdict (YTA or NTA, extracted with a list of variations). Our dataset contains 21K posts, and 364K verdicts (254K NTA, 110K YTA) written in English. To analyze the types of conflicts, we further group posts into distinct categories as described in the following paragraphs.

5.3.1 Extracting Demographichs

We modify the script from [31], which extracts age and gender using a set of phrases such as "I am a woman" and "I am X years old", to also capture Reddit shorthand. Reddit's users often disclose their age and gender when telling stories or asking for help. This often takes the form of a letter and number in brackets or parentheses (e.g. "[32F]" for age 32 and female), immediately after a first person pronoun. We base this extraction on recent work that has used similar methods [146, 147]. Additionally, we capture gender expressed in a phrase containing an adjective, such as "I am a quiet man". We adjusted the regex to exclude false positives like "I am a manager" or "I am a manly girl".

We then split ages into two groups. The median age of 28 is used to group people into *younger* and *older*. The resulting dataset contains 1,121 younger people (8% of total) and 1,032 older (7.6%). For gender, we find 2,280 are male (16.8%) and 3,392 female (25%). Note that our scripts exclude many people, including those who are non-binary. See our limitations section for more details.

5.3.2 Annotation of Conflict Aspects

Given the history of the typology of conflict, discussed in §5.2, we decided to measure six aspects of conflict; (1) strength of disagreement, (2) intensity of negative emotion, (3) degree of interference, (4) duration of the conflict, (5) manifestation of conflict, and (6) how many people are involved. Aspects 1-3 correspond to the three attributes outlined by [154], but with the view of measuring their intensity. [153]'s suggestions directly inspired aspects 4 and 5 and [152]'s suggestions about groups led to aspect

6.

The authors then annotated a sample of 25 conflicts in order to refine our task. This process made evident how previous conflict scales were not well-equipped for our data. Our conflict situations do not always take place in work settings. The nuance of scales like [150] seemed unnecessary, as conflict is assumed in our setting and as a third party, levels of intensity are less clear (e.g. how to differentiate between degrees of friction, tension, emotional conflict, and personality conflict). Longitudinal aspects also cannot often be directly determined.

With these insights, we refined our annotation questions. We recruited annotators from the crowdsourcing platform Prolific,[1] as well as asking researchers at our university to help annotate as part of their paid working time. There were 14 annotators in total and all were required to have English fluency. All surveys included two attention check questions that provided the same options as the disagreement strength and negative emotion questions, but asked “How should you answer this question? You should answer”, followed by one of the three options. All annotators passed all attention checks. Annotators were asked the following six questions for each Reddit post and additional details on how the labels should be used:

1. How strong is the disagreement or opposition? **Labels:** (Mild, Strong, Intense) with Strong and Intense merged. **Additional details:** You should consider how significant the event seems to the author. For example, a conflict over who should clean the dishes may seem mild, whereas a conflict over divorce may seem intense. However, if the author describes the conflict over dishes as a fight that is causing irreparable damage to the relationship, it may be strong or intense.

2. How intense are the negative emotions? **Labels:** (Mild, Strong, Intense) with Strong and Intense merged. **Additional details:** Use the mild label when emotions are weaker, or it is not clear if they are there at all. Use the strong and intense labels to differentiate between situations where you perceive stronger emotions from the participants.

3. How much is one person interfering with what another wants to or can do? **Labels:** (Not at all, Somewhat, Strongly) with Not at all and Somewhat merged. **Additional Details:** If someone clearly cannot do what they would like and that is the subject of the conflict, then the interference is strong. If there is a disagreement, but parties can still take whichever action they desire, then there is no interference (e.g. telling someone not to do their homework but not stopping them from doing it). If there are alternatives or possibility for some degree of compromise then there is some interference (e.g. a tenant is upset that they cannot pay rent in two parts, landlord gives several alternatives), but if the restricted party is clearly opposed to all options then the interference is still strong (e.g. daughter is not allowed to go to boyfriends house).

4. What is the duration of the conflict? **Labels:** (One-time incident, Longer) **Additional Details:** Additional Details: If someone describes a specific incident that occurred at one point in time then it is a one-time incident (e.g. posting something rude one time on Facebook, not wanting sibling to take over a family vacation with her plans). If the author explicitly states that something is an ongoing conflict over multiple days (or longer), or if it can be reasonably inferred that a conflict spans multiple days (e.g. “every time I talk to my parents we have this problem”), then the conflict is longer term.

[1] https://www.prolific.co/

5. Has the conflict primarily manifested in what someone has said or done or is the conflict primarily perceived by the author? **Labels:** (Manifest, Perceived) **Additional Details:** Additional Details: A conflict can become manifest, for example, in the form of fights, arguments, telling someone something, or taking something, whereas the perception of conflict happens inside someone's head (e.g. someone thinks of themselves as rude/mean/unfair, but we do not know if another party has this same perception because we do not know what they have said or done or if they are aware of or have engaged in the same events as the author). For example, the author feels bad for not texting his parents back quickly. If we have no evidence that this is causing problems between them or that the parents have a problem with this then it is perceived. Sometimes there are small manifestations, but the conflict is still mostly perceived. For instance, the author is blocked on Facebook for not inviting a friend to a party, but the author does not seem to engage with the other person or understand why this is a conflict. In this case, it is primarily perceived by the other person.
6. Who else is directly in conflict with the author? **Labels:** (One person, Multiple people) **Additional Details:** Additional Details: A conflict with multiple people should only count people engaging with or contributing to the conflict. For example, if A tells B to shave their beard and C gets mad at B for doing so, B and C are in conflict but as long as A does not engage, they should not be considered to be part of the conflict and so this would be a one-person conflict.

A subset of 500 posts corresponding to 1,653 comments from the test set were provided to annotators. Matthews correlation coefficient (MCC, [171]) was used to measure agreement between annotators for 100 posts and is shown in Table 5.1. We find moderate to strong agreement for most aspects except for the degree of interference and whether the conflict is primarily manifest or perceived. For the non-binary aspects, we condensed labels (denoted by →) and treated all labels as binary in subsequent analyses. The *strong* and *intense* labels for the negative emotion and disagreement aspects were merged into one *strong* category. The *lesser* and *none* labels for the degree of interference were merged into *mild*. Other labels were already binary and were unchanged. The resulting distribution is shown in Table 5.2.

Disagreement		**Emotion**		**Interference**		**Duration**		**Manifestation**		**Num. People**	
Mild	Strong	Mild	Strong	Weak	Strong	Once	Longer	Perc.	Mani.	One	More
33.0	67.0	35.7	64.3	35.3	64.7	48.3	51.7	33.7	66.3	72.0	28.0

Table 5.2: Label distribution for merged label values resulting from human annotation of 500 posts.

5.3.3 Clustering

Before we acquired any annotated data, we performed an exploratory analysis to determine if there was a natural way of grouping conflicts into different types that would be useful for our analysis. We used two representations to perform clustering: situations and all text from the post (full text). Situations, as referred to in [144], come from the title of a Reddit post and serve as a summary of the situation described in the full post. The posts usually start with "AITA for", which we omit. We cluster posts using Louvain clustering, which maximizes the modularity of our graph [172]. We create a weighted

Cutoff %	**0**	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
Number of Situation Clusters	4	3	4	3	3	4	4	4	4	4
Situation ARI	-	0.44	0.47	0.46	0.93	0.57	0.45	0.49	0.91	0.85
Number of Fulltext Clusters	3	3	3	4	3	5	8	18	49	165
Full Text ARI	-	0.60	0.92	0.91	0.75	0.72	0.74	0.89	0.81	0.65

Table 5.3: The resulting number of clusters using Louvain for different graph representations, and cutoff percentages. ARI denotes the adjusted rand index between the listed cutoff percentage and 10% less.

graph based on each criterion, using situations or full texts as nodes. Their embeddings are obtained with Sentence-BERT (SBERT; [44]), and use the cosine similarity, normalized to $[0, 1]$ between each pair of nodes as weighted edges, resulting in two fully-connected graphs. The graphs are pruned by dropping the N% lowest edge weights determined by the adjusted Rand index between graphs with a 10% difference in the number of dropped edges in order to find a persistent clustering. This yields N=40% for situations and 30% for full texts. First, we determine the amount of edges to drop from our fully connected graphs to drop. This was determined using the adjusted rand index between 10% differences. Further threshold values, ARI, and resulting cluster numbers are provided in 5.3. Although we do use a cutoff of 30% for full texts, which has 4 clusters, one of these clusters contained only 25 posts, so we removed it. We experimented with K-means in preliminary experiments but found that it had a lower agreement with human clusters and clusters seemed less clear.

Manually inspecting the clusters revealed that the groups differ from each other by the social relation of the author to the others in the situation, or how close the author is to others in the situation. For manual verification but also in an effort to explore possible modifications to the groupings, a subset of 100 posts were manually clustered by two of the authors, who intended to form a small number of groups based on the post title and content. While considering other possible groupings both came to the conclusion that it appears most natural to group the posts based on social relations. The events that occur in a conflict, understandably, appear strongly dependent on the relation between participants. Upon manual inspection and discussion between annotators, we find that differences arise from two sources. The first is boundaries between social relations. For instance, one annotator grouped family, romantic relationships, and best friends into one cluster, and put all other friends in a second cluster, while the other annotator put family in one cluster and all romantic relationships and friendships in a second. The second source of disagreement comes from the perception of who is involved in the conflict. For instance, in one post, a person borrows an object from a family member's friend and although the family member is upset, we do not know if the friend is upset. One annotator saw this as a family conflict, while the other saw it as involving someone more distant. The ARI was 0.33 between humans, 0.38 and 0.15 between full text and humans, and 0.31 and 0.13 between humans and situations. We refer to the *Family* cluster and the clusters containing *Close* or more *Distant* individuals in subsequent analyses. Two examples of posts belonging to each of the clusters are shown in Table 5.4. Clusters were obtained using the full text.

Family
Situation: Not watching horror films with my husband **Full Text:** I really don't like horror movies. I dislike gore and loud noise out of nowhere shock tactics especially, but I also have a tendency to get nightmares from movies that don't have those issues. I don't enjoy being scared. Plot holes also stick out like a sure thumb in horror to me. I will try movies on occasion if he really wants me to see them and he says it isn't a gore/shock tactic movie, but it takes a lot of pleading on his part. I almost never enjoy them and generally my reaction is that it was okay/fine, wouldn't watch it again. I watch things I want to see but he wouldn't enjoy separately. I ask him to watch things that I think he will actually like sometimes and he always does. He often watches horror after I go to bed. The things we watch together are things we are both agreeable to. We watch at home. I only wonder if I'm the asshole because it seems common for couples to trade off who picks movies.
Close Relationships
Situation: Not attending my friend's debut **Full Text:** She already placed me on a list where they call people up to give gifts ad stuff without eve asking beforehand if I'll be able to attend. I feel like a real asshole right now because 18th birthdays only happen once in a lifetime ad I wasn't there to celebrate with her when she was expecting me because I needed to attend a birthday for my uncle who was released out of prison. On the other hand, I do feel a bit angry that she listed me before asking. Now everyone has cards with my name on them, ad whoever is attending will expect me to join as well. I feel some conflict. She didn't even tell me the address, she just told me that I'm invited and my name is on the card and I need to give her a gift. She seemed really disappointed days ago when I told her that i could't attend. Stopped talking to me. Didn't even look at me. Tried texting my other friends who were invited but didn't respond. Too busy partying. I have a feeling that people will think of me as a shitty friend and that I'm no good. So, AITA?
Distant Relationships
Situation: Leaving low tips **Full Text:** So there was an event at a bar/club I bought a ticket for online, *pre-paid* - but when I got there, even though I had a ticket, they were unable to let me in due to "max capacity". I mean, normally I don't take it to heart and either wait or find somewhere else, but this was something that I paid for, so I figured it's not fair since I technically paid to be part of that 'capacity'. There were a few others in the same boat as me who they had to do that to who were also frustrated. Eventually I got in, but I was super aggravated because I ended up missing over an hour of the event because of this, and while I was able to eventually enjoy my night I found myself leaving low tips, since I was quite livid (and felt I lost some of my money's worth). Later on I felt kind of bad because I realized it's probably not the bartenders' faults. AITA though?

Table 5.4: Two examples of post situations and full text for each of the three clusters (manually labeled, but automatically clustered using the full text).

Diff.	Disagree $p < 0.002$		Emotion $p < 0.02$		Interference $p < 0.3$		Duration $p < 0.04$		Manifestation $p < 0.04$		Num. People $p < 0.007$	
	M	S	M	S	W	S	Once	Long	Perc.	Mani.	One	More
Acc	89.5	88.3	88.3	84.0	84.7	86.3	82.7	86.5	81.8	86.4	86.1	80.0
Mic F1	70.8	69.5	70.0	69.6	56.4	85.5	68.2	70.7	51.9	73.7	73.1	42.5
Mac F1	78.0	76.4	77.8	76.6	74.5	85.5	71.7	82.0	73.2	78.9	78.5	72.7

Table 5.5: Performance across conflict aspects for our model using the full-text stratification, showing accuracy (Acc) and F1-score. Significance values for differences in model performance between each dyad are shown above, calculated with one-sided unpaired permutation tests. Mild (M), Strong (S), Weak (W).

5.4 Perception Experiments

5.4.1 Hypotheses

After choosing the six conflict types, we developed hypotheses about which values would be associated with conflicts and whose verdicts would be most difficult for our model to predict. We hypothesized that higher emotional intensity would be more difficult, as different people may empathize differently and the classification of emotions is known to be a challenging task in itself. When more people are involved in a conflict, we hypothesized that this would be harder for our model to predict. With more involved parties, coreference resolution becomes more challenging and the interaction of more parties may make interpretation of the situational context more complex. However, we thought that the classifier would perform similarly for both mild and strong disagreements, as we did not see why this aspect by itself would make the task more or less challenging.

We predicted that it would be easier for the model to predict conflicts that occur over a longer duration, that involve more interference, and that are more manifest than perceived. First, longer-duration conflicts may mean that there has been more time to accumulate information about the conflict. In our observations, it also often means that someone is repeating an action. These repeated actions and additional information may give a clearer signal of what facts will lead to a verdict. Similarly, with interference, the action is much clearer when interference is high (e.g. someone taking something away from someone, or preventing people from seeing each other). Lastly, when conflict is manifest, it means that an annotator decided the conflict was more manifest than perceived by the author. When the conflict is more perceived, the reader has to infer more from the text. For example, the author may think they did something wrong (e.g. not moving in with a friend) but the author does not seem to know how the other person feels.

5.4.2 Results

We classify the perception of individuals based on their comments on posts. We concatenate the situation (post title) and comment text after filtering out any labels (e.g. YTA). As our base model, we fine-tune SBERT on the binary task of predicting the perception of the author, given by a verdict (YTA or NTA). We also tried using this model to encode the full text to use as additional features, though we found no difference in performance over using only the comment text and situation, which often succinctly captures the event.

We compare our model to the recent work of Botzer et al. [146] JudgeBERT, which is a BERT-

		Full Text		Situation	
		F1%	Acc%	F1%	Acc%
Botzer et al. [146]	All	72.7	84.9	70.1	83.2
	Family	74.9	86.8	73.3	85.4
	Close	72.2	84.4	67.8	82.2
	Distant	71.2	82.2	68.5	80.8
Our Approach	All	77.2	87.0	77.4	87.2
	Family	79.0	88.3	78.7	88.4
	Close	76.7	86.9	77.4	86.9
	Distant	75.9	85.0	75.6	85.4

Table 5.6: Comparison between Botzer et al. [146] and our approach with accuracy (Acc) and macro F1-score. Results are broken down by cluster (labels from §5.3.3).

base [4] model fine-tuned on our dataset, which is extended with a dropout layer and classification layer. JudgeBERT was evaluated in the work from [146] using a dataset with collections of posts submitted between January 1, 2017, and August 31, 2019, over different subreddits. For the purpose of this work, we re-implemented JudgeBERT in order to evaluate it on our dataset. The main difference between the two models is the encoder layer, where one uses a BERT-base model, and the other one a SBERT model. We train both models for 10 epochs, using the Adam optimizer, learning rate of $1e-4$, and focal loss [173] to cope with class imbalance. We split our dataset into 70-20-10 for training, validation, and test, respectively. We stratify in two ways, for each clustering method.

The results are reported in Table 5.6 for both models and splits. We see that our model significantly outperforms previous work on all data,[2] with a 5-point improvement on full-text F1 (macro averaged over posts, which may have multiple verdicts from different users) and 7 points on situations.

We further break down our results by conflict aspects in Table 5.5. We find significant differences in our model's ability to predict the perception of conflicts between each aspect dyad with the exception of interference, which had a label distribution least similar to the other conditions. We also find that the types of judgements in our sample vary significantly across each aspect of the conflict. The difference in the distribution of NTA and YTA labels between each dyad shown in Table 5.5 is statistically significant using Fisher's exact test. In the difference for disagreement ($p < 0.004$), *Strong* contains an 11% higher ratio of YTA/NTA judgements. For emotion ($p < 0.02$), this difference was 9%. Interference ($p < 0.001$) had the highest difference of 78%, with more YTA judgements when the degree was *Strong*. For duration ($p < 0.001$), *One-time incidents* had a 13% higher ratio. Manifestation of conflict ($p < 0.0003$) showed a 13% higher ratio when conflict was more manifest than perceived. Lastly, when only one person was involved ($p < 0.03$), the ratio of YTA/NTA was 11% higher. All ratios skew toward more NTA, as this is the overall bias of the dataset, and all differences in the ratio are calculated as absolute differences of YTA/NTA between values of an aspect. We correctly hypothesized that situations with more negative emotion would be more difficult for our classifier, though we also found this to be the case for disagreements. Further work is needed to understand the relation between disagreement strength and perception classification. We also correctly hypothesized that conflicts involving more people are more difficult for our classifier and that stronger

[2] Permutation test for full text and situations, $p < 0.0001$.

interference, longer duration, and primarily manifest conflicts were easier to classify, though the improvement for interference was not significant.

5.5 Personalization Experiments

5.5.1 Tasks Formulation

We formalize our task in terms of the textual data points, their authors, annotators, and the annotations they provide. A poster, u, makes a post, p, which is then commented on by an annotator, with ID a, who provides a comment, $c_{a,p}$, and a label, or verdict, $v_{a,p}$. Since we are modeling annotators, u is not important to us, except that $u \neq a$ within the same post. Each post p has many comments c_p, though this is not strictly necessary for our purposes, it does help reveal the subjectivity of the task. Importantly, each annotator, a, has many comments, c_a. In our case, the comment c_{a,p_i} written by annotator a on the i-th post p_i, is linked to a single v_{a,p_i}, though one could gather these from separate sources, and doing so may be necessary for other corpora. The subjective nature of the task and its evaluation comes from the assumption that annotators provide different verdicts for a post.

Work on annotator modeling attempts to estimate the probability of a verdict given the post and annotator, $p(v_{a,p}|a, p)$. This is in contrast to predicting what an individual's language means, $p(v_{a,p}|a, p, c_{a,p})$, which we refer to as a personalized classification task. Importantly, we make this distinction because personalization has historically focused on predicting a label assigned to an individual's text in a particular context (e.g. the sentiment of a review), whereas work on modeling annotators focuses on the label an individual *would assign* in that context.

There is often no information about annotators, or only an ID is known. A few works on annotator modeling include extra information about the annotator, T. In this work, we use a collection of other texts from the annotator (see §2.3.1). To the best of our knowledge, our formulation of T is novel in that it allows the application of previously developed methods for personalization to the task of annotator modeling. Importantly, we are predicting how the annotator will label the post, $p(v_{a,p}|a, p, T)$, not how to interpret their text. For other work that has attempted to interpret verdicts, $p(v_{a,p}|a, c_{a,p})$, refer to §5.2.

5.5.2 Experimental Setup

We experiment with four personalization methods for annotator modeling (see §2.3.1) and two situation text baselines for a secondary task of personalized verdict interpretation.

SBERT (text only interpretation model): As our base model, we finetune SBERT on the binary task of predicting the verdict, given the comment and the situation title.

JudgeBERT (text only interpretation model): We compare our personalized models to Judge-Bert [146], a recent model that was developed to study moral judgements, and reported the highest performance of the models discussed in §5.2. Though our novel task setup does not have an existing baseline to directly compare to, this comparison, which does use the verdict text, serves as a point of reference.

Averaging Embeddings: We finetune the SBERT base model and add an additional layer to concatenate the text representations with annotators representations, using the initial annotator representations computed from SBERT for Annotators.

Priming: This model is the same as the base model, but the input text is different. The SBERT base model is finetuned on the binary task of predicting the verdict, given the situation title, the sampled text from each annotator, and the comment in the interpretation model case.

Author Attribution: In this setup, we have the same architecture as averaging embeddings, however, the initial annotator representations are generated using the author attribution model.

Graph Attention Network: In addition to the SBERT fine-tuning over the comment and the situation title, we train a GAT model to learn the annotator representations.

Author ID: This model appends only an author-specific ID to each input. This approach is similar to the common ID-only personalization and annotator modeling approaches discussed in §5.2.

We train our models for 10 epochs, with the Adam optimizer, using initial learning rate $1e-4$, and focal loss [173] to cope with class imbalance. As our base SBERT model, we use DistilRoBERTa [174], with a dimension of 768 and a maximum length of 512. For the priming method, we sample $m = 100$. Moreover, we set $d = 768$ in the author attribution model, and train three different networks depending on the number of authors for the corresponding training split. The model is trained for 100 epochs, with the Adam optimizer, using the initial learning rate $1e-5$. Our experiments are run on a single NVIDIA A100 40GB GPU with an average running time (training + inference) of around one hour.

5.5.3 Three Splits

We split the data in three ways. The first is randomly splitting verdicts into train, validation, and test. This involves two confounds; the same situations and the same authors can occur in multiple splits. Our dataset contains authors who comment on many situations, providing a verdict. A graph containing nodes corresponding to authors and posts and edges representing annotators who comment on a post is fully connected. It is therefore not possible to remove both confounds at once without removing edges, reducing the data size, which introduces a new confound. Instead, we examine two additional splits, each controlling for one of the two confounds. The situation and author splits have disjoint sets of situations and authors respectively, across train, validation, and test.

	No Disjoint		**Situations**		**Authors**	
Annotator Model	**Accuracy**	**F1**	**Accuracy**	**F1**	**Accuracy**	**F1**
Averaging Embeddings	**86.1**	**83.3**	66.5	56.2	**86.0**	**83.2**
Priming	83.9	80.6	**69.6**	52.9	70.2	41.2
Authorship Attribution	85.5	82.4	68.4	**56.1**	85.2	82.3
Graph Attention Network	86.0	83.0	67.8	54.4	85.6	82.7
Author ID	85.1	82.1	66.7	55.0	84.3	81.1

Table 5.7: Accuracy and macro F1 scores as percentages for each split method. Bolded numbers are the best results for each column and significantly outperform the next best model ($p < 0.0002$ for situations, $p < 0.002$ for authors, and $p < 0.0004$ for the no disjoint set, with paired permutation test).

5.5.4 Results

Our main findings are in Table 5.7. We find that the performance of models is similar when there are no disjoint sets across splits as when splitting by authors, with the exception of priming, which greatly suffers from not having the same authors to train on. Generalizing to new situations proved the most difficult, suggesting that having experience with interpreting specific situations is more helpful than having experience interpreting specific authors.

When comparing to a majority baseline, the accuracy is 70% for both the author and no-disjoint splits, and 71% for situations. The macro F1 baseline is 50%. In all cases, we outperform the majority baseline, except for situation split accuracy. Although accuracy on the situation split is low overall, the macro F1 is still higher than the baseline. In preliminary experiments, we also tried training models with the full situation text (i.e. the full Reddit post), and found accuracy was slightly higher but F1 was lower.

We are able to compare personalization methods for this challenging task and find that priming has the highest accuracy in the situation split while averaging embeddings and the authorship attribution approach consistently high accuracy and F1 scores. The low performance of priming is similar in the author split, which is close to the baseline, suggesting that priming often does not provide a useful signal to the model. In addition, we notice that using only the author ID as an additional token in the text, is still better than priming, which shows that using randomly sampled text from the authors might sometimes be misleading. Averaging embeddings proved to be very effective considering the simplicity of the method compared to authorship attribution and the graph attention network. In contrast to Welch et al. [160], we found that authorship attribution representations can scale to a large number of users by learning a projection layer to reduce it to a similar size as the text encoding. Contrary to [42], who found that priming outperformed other methods in relatively low data settings (like ours), we find that it underperforms other methods at the verdict-level. Moreover, adding a GAT offers lower improvement than averaged annotator embeddings. This is contrary to our previous work [13], where adding the GAT layer yielded improvements. This may be due to different social media data and interactions (the sarcastic dataset used Twitter data).

	No Disjoint		**Situations**		**Authors**	
Interpretation Model	**Accuracy**	**F1**	**Accuracy**	**F1**	**Accuracy**	**F1**
Text Only (SBERT)	91.7	90.0	83.0	79.0	91.2	89.5
Botzer et al. [146]	89.2	87.0	78.4	71.6	84.6	82.2
Averaging Embeddings	91.7	90.0	83.4	79.5	**91.7**	**90.0**
Priming	90.9	89.0	80.6	75.7	89.7	87.4
Authorship Attribution	**91.9**	**90.1**	**84.2**	**79.7**	68.8	64.3
Graph Attention Network	91.5	89.8	83.3	78.7	91.4	89.7
Author ID	91.5	89.7	84.1	79.5	91.2	89.4

Table 5.8: Accuracy and macro F1 scores as percentages for each split method. Situations and authors are disjoint across splits for the latter two respective column pairs, whereas the first is split by neither, meaning some authors and situations (but not verdicts) overlap across splits. The best models are bolded and in the situation and author splits are significantly better than the SBERT baseline ($p < 0.0003$, paired permutation test), though the result without disjoint splits was not significant.

Our results using the verdict text are shown in Table 5.8. Although this is a separate task, with a goal of interpreting an author's verdict rather than predicting it, it does provide additional insight. We find that our models greatly outperform previous work and future work should consider SBERT as a baseline without personalized features. We also find that the personalized methods outperform the text only baseline except for the priming method. Authorship attribution often performs best, though averaging embeddings outperforms other methods on the authors split.

5.6 Analysis & Discussion

5.6.1 Perception

Overall, our model outperforms previous work for our full data and for each cluster. As noted in §5.2, it is important to understand the subject of the conflict, though in our work we found that this was highly coupled with the type of relation between participants. Future work may consider ways of separating these concepts.

If one considers the Family cluster as the closest social relationship, we find an indirect relationship between the closeness of participants in a conflict and the difficulty in classifying perceptions of that conflict.

The closeness of relation to conflict participants, the strength of negative emotions and opposition, the duration of the conflict, manifestation, and the number of people involved, all impact our classifier's ability to classify people's perception of social norms. These findings pertain to the understanding of conflict, behavior, and personal narratives, but may prove useful for other tasks such as argumentation, framing detection, and understanding offensive speech.

Annotator Model	Distant	Close	Family
Averaging Embeddings	59.2	62.1	65.7
Priming	61.6	64.6	67.9
Authorship Attribution	59.7	62.1	64.5
Graph Attention Network	58.6	61.2	66.3
Author ID	58.9	61.2	64.5

Table 5.9: Macro F1 scores for performance on the situation split, showing that when the relationship between people in conflict is distant (e.g. co-workers, strangers), personalization does not help (50% baseline), but the closer the relationship (e.g. friends, family), the more personalization helps.

5.6.2 Personalization

Performance across tasks. We further analyzed the performance of our methods with respect to the clusters from §5.3.3. We use the situation split with no verdicts, as this most clearly demonstrates performance. The macro F1 scores are shown in Table 5.9. We see a direct correlation between the closeness of the relationship between parties in conflict and the effectiveness of personalization. This means that for relationships such as between family members or friends, personalization methods can better learn how people will judge actions in these situations. However, when relations are more distant, such as those between co-workers or strangers, personalization methods are not as capable of helping to predict judgements. This is a key insight that raises questions for future work on judgements of social norms, but more generally suggests that the effectiveness of personalized models should be considered in terms of the properties of the classification task.

Who personalization helps. To be able to better understand the impact across individuals, we plotted the distribution of accuracies for the situation split with no verdicts in Figure 5.2. We see that

Annotator Model	Gender Male	Gender Female	Gender Unknown	Age Younger	Age Older	Age Unknown	All
Averaging Embeddings	65.8	65.2	66.7	65.4	65.9	66.3	66.3
Priming	70.4	69.7	70.1	68.6	70.4	70.2	69.0
Authorship Attribution	68.3	67.1	68.7	68.0	68.1	68.2	68.2
Graph Attention Network	67.5	67.6	68.1	67.1	67.5	68.0	67.0
Author ID	66.0	66.3	66.0	66.2	66.6	66.0	67.1

Table 5.10: Breakdown of annotator-level accuracies for each personalization method in the situation split. We show performance independently for age and gender, using three values for each.

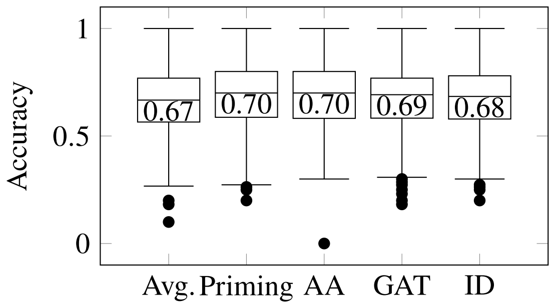


Figure 5.2: Distribution of accuracy for the no-verdict situation split across users with at least 10 verdicts in the test set ($N = 715$). Median values are in the boxes. Abbreviations: AA=Authorship Attribution, Avg.=Averaging Embeddings, GAT=Graph Attention Network, ID=Author ID.

for our models, the variance is similar. We find that performance is much higher for some annotators than others.

To further examine differences we looked at the annotator-level performance across demographics extracted as in §5.3.1. We use three values for each, including a value for *unknown*. The results in Table 5.10 show the accuracies for each demographic do not vary much from the overall scores for each model. Interestingly, we find that those with unknown demographics tend to have slightly higher performance. This group of individuals who are less likely to share demographic information may have something in common that is beneficial for modeling judgements, though this remains to be explored. Overall, our models show relatively fair performance across demographics, not strongly performing for one group over another, even with an uneven distribution of genders. Accordingly, when the only identifiers used for the annotators, are their IDs, the performance across demographics is at 66%.

Annotator Data Volume. When looking at annotator-level accuracy, one may wonder if it helps to have more data for an annotator during training. We tested this with the situation split using Pearson's correlation coefficient. We calculated annotator-level accuracy for each annotator and grouped them by the amount of available data points during training (up to 463). Then, we calculated the correlation between the amount of data and the mean accuracy across annotators that had that amount of training data.

Across methods on the situation split we find $r = 0.19 - 0.22$ ($p < 0.05$) with slight variance across methods. When we looked more specifically at the three tasks, we found that for the Distant task, correlation is weaker, $r = 0.14 - 0.19$ ($p < 0.08$), and we find no correlation in the Close task, $r = -0.02 - 0.11$ ($p < 0.6$). With the Family task we find the strongest correlation with $r = 0.18 - 0.25$ ($p < 0.02$). Having more data per annotator helps with Family, but does not help as much with the more distant clusters. Interestingly, having more data seems to help more with predicting verdicts of the Distant task than it does the Close task.

5.7 Summary

In this chapter, we addressed the second research question and focused on modeling the context of recipients to accurately model their responses to various discourses.

In Section 5.3, we introduced our dataset, alongside with our manual annotations for conflict aspects, user demographics, and clustering of social situations. Subsequently, we provide two sets of experiments, the first one focusing on perception classification and conflict aspects, and the second one employing personalization. In the first experiments, we found significant differences in our model's ability to predict the perception of conflicts between each aspect dyad with the exception of interference, which had a label distribution least similar to the other conditions. We correctly hypothesized that situations with more negative emotion would be more difficult for our classifier, though we also found this to be the case for disagreements. Furthermore, we also correctly hypothesized that conflicts involving more people are more difficult for our classifier, and that stronger interference, longer duration, and primarily manifest conflicts were easier to classify, though the improvement for interference was not significant.

Following our experiments on perception classification, we formalized our personalization task, in terms of the textual data points, their authors, annotators, and the annotations they provide. We investigated the performance of our proposed personalized approaches such as A-SBERT, Priming, Author Attribution, Graph Attention Network, and Author ID across three different dataset split. Each split focusing on different dimension of our dataset. We found that the performance of models is similar when there are no disjoint sets across splits as when splitting by authors, with the exception of priming, which greatly suffers from not having the same authors to train on. Overall, we found that averaging embeddings provided a strong and relatively simple approach, though each model has its strengths, the authorship attribution and graph attention networks were consistently high performing across splits, while for the situation split, annotator-level accuracy was highest with the priming approach. These methods outperform the common approach of representing authors with a single ID. As a secondary result, we found that personalized methods significantly outperformed previous work and text-only baselines on the task of interpreting verdicts.

Moreover, our analysis across tasks, showed a direct correlation between the closeness of the relationship between parties in conflict and the effectiveness of personalization. This key insight shows that for relationships such as between family members or friends, personalization methods can better learn how people will judge actions in these situations. We further showed performance across demographics, showing that our methods appear unbiased in this regard, despite being trained with more data from females than males. We revealed a correlation between the amount of data from an annotator during training, and it's impact on personalization, showing that more data generally, but not always, helps. We hope that our formalization of this task provides a path for future work in this direction, and our insights for the task of predicting judgements of social norms provide meaningful first steps.

In the next chapter, we shift our focus from static user representations towards dynamic user representations.

CHAPTER 6

Static and Temporal User Classification

In the preceding chapters, we have explored different personalization techniques in order to integrate user context into text classification systems. Our approaches modeled both the sender and recipients of the communication, addressing important aspects of discourse analysis. These representations were static, assuming user context remains consistent throughout their posting history. However, users' semantic and social interactions aim to change over time. For instance, during electoral campaigns, a user might shift his usual social interactions with specific groups towards political communities. Modeling these temporal patterns of users' interactions can provide insights into how their opinions or behaviors change, in addition to predicting future behaviors. Nevertheless, static user representations fail to capture the dynamic nature of online behavior and interactions. To address this gap, in this chapter, we focus on temporal user-to-user interactions. We treat user context as dynamic that changes over time over different social and semantic contexts. This chapter delves into the following research question:

> Research Question 3 (*RQ3*)
>
> Do the dynamic user representations help to capture the temporal behavior of users related to their social networks?

This inquiry marks a shift from the static user contexts discussed in earlier chapters, focusing instead on the potential of dynamic representations to more accurately reflect users' evolving behaviors. Unlike the previous focus on text-centric tasks, this chapter introduces a novel task centered on the users themselves, aiming to capture the temporal dynamics of their online behavior. For this purpose, we present a new dataset that consists of the posting activity of users over an 18-month period, providing a rich temporal dimension for analysis.

Proactively identifying misinformation spreaders is an important step toward mitigating the impact of fake news on our society. Hence, in this chapter, we focus on utilizing time-evolving representations, in order to identify potential misinformation spreaders. We propose to design a framework capable of temporally and jointly modeling these user networks, in order to compute dynamic user representations.

The key contributions of this chapter are:

- We introduce FACTOID[1]: a user-level **FACT**uality and p**O**litical b**I**as **D**ataset, that contains a set of 4,150 news-spreading users with 3.3M Reddit posts in discussions on contemporary political topics, covering the time period from January 2020 to April 2021 on individual user level.
- We conduct classification experiments for identifying misinformation spreaders by utilizing the social connections between the users along with their posting history representations and psycho-linguistic features.
- We provide a comprehensive qualitative and quantitative analysis of the users' temporal semantic and social similarities and investigate the different types of dynamic graph connections.
- We develop a dynamic graph neural network framework for (a) predicting the users' future misinformation spreading behavior, (b) predicting the behavior of unseen users, and (c) predicting misinformation spreading behavior in a zero-shot scenario.

This chapter is based on the following publication ([17, 18]):

- **Joan Plepi**, Flora Sakketou, Riccardo Cervero, Henri Jacques Geiss, Paolo Rosso, and Lucie Flek. 2022. FACTOID: A New Dataset for Identifying Misinformation Spreaders and Political Bias. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 3231–3241, Marseille, France. European Language Resources Association.
- **Joan Plepi**, Flora Sakketou, Henri-Jacques Geiss, and Lucie Flek. 2022. Temporal Graph Analysis of Misinformation Spreaders in Social Media. In Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, pages 89–104, Gyeongju, Republic of Korea. Association for Computational Linguistics.

The rest of the chapter is structured as follows. Section 6.1 introduces the work, and we describe the related work in Section 6.2. We introduce the FACTOID dataset in Section 6.3. In Sections 6.4 and 6.5, we show our experiments, results and analysis respectively for static and temporal setups. Finally, in Section 6.6 we provide a summary for the chapter.

6.1 Introduction

As the popularity of social media platforms continuously grows, so does the dissemination of online disinformation. Many deep learning systems have been therefore developed to detect false or biased news [175, 176, 177, 178]. While fake news detection is a big step to mitigate the impact of misinformation on our society [179, 180], it is not sufficient, since limiting the diffusion of false information and avoiding its catastrophic effects is extremely challenging, especially once it has been shared on the Web [181, 182]. Research shows that fact corrections frequently fail in reducing people's misconception of the truth, and occasionally they even have a "backfiring" effect where people's misconception is reinforced [183, 184, 185, 186]. It is essential to address this issue at its origin - to efficiently and rapidly identify accounts and users which are likely to propagate posts from the handles of unreliable news sources. While there are numerous datasets focusing on this issue at a post-level,

[1] https://github.com/caisa-lab/FACTOID-dataset

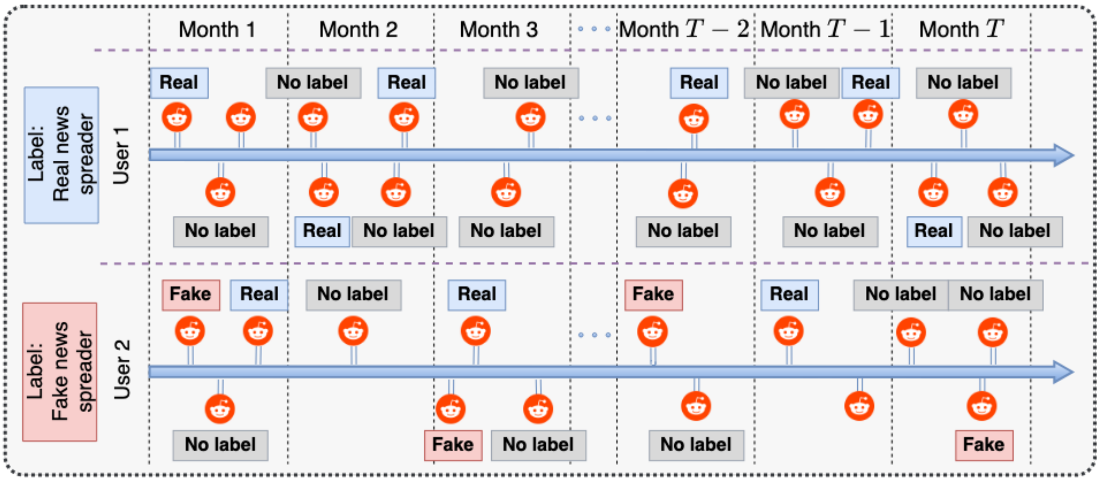


Figure 6.1: Examples of the user classes.

only very few of those are allowed to approach this matter on a user level, since, in most cases, fake news posts are not associated with their individual authors.

Moreover, existing datasets designed for identifying misinformation spreaders only include binary labels for the users. However, the reality is not black and white, therefore a credibility score associated with each user is more realistic. In addition, since partisan polarization constitutes one of the primary drivers of political fake news sharing [187], it is becoming all the more vital to explore the political bias of users in combination with their misinformation-spreading behavior. To the best of our knowledge, there is no existing dataset that combines both of these dimensions on a user level with fine-grained scores. To this end, we introduce a dataset for distinguishing the authors who have shared news from unreliable sources in the past, from those who share news from reliable sources, covering the posting activity of the users before and after the 2020 US presidential elections. We use the terms *misinformation spreaders* and *real news spreaders*, respectively. Apart from the binary labels, we assign a credibility score to each user based on the factuality of the news sources they share, and a political bias score based on the level of partisanship of the news sources they share.

In addition, the impact of time on fake news prediction has made the task even more challenging, as the content-based differences of news sources change due to the highly dynamic nature of the news topics [188]. Most of the fake news detection methods that use static features need to be continuously updated with new annotated data to stay relevant [189]. We argue that this hypothesis can be generalized for detecting misinformation spreaders. Similarly to feature-based methods, existing graph modeling approaches are not specifically designed for learning the time-evolving similarities of the users' interactions. Addressing these limitations of existing research, we propose an approach accounting for the temporal dynamics of user-to-user relationships instead. We introduce a model that extracts features from users' content similarities and social interactions and models the temporal evolution of these connections in order to identify misinformation spreaders.

6.2 Related Work

Datasets

User profiling approaches have been investigated for various tasks, such as author profiling [190], bot detection [191, 192, 193], gender detection [194], among others. However, fake news spreader detection is an under-explored research direction. There are some datasets approaching this matter from different angles, for example, attempts have been made to analyze the user's reactions to fake news [195] or to analyze users who debunk fake news [196]. However, there are only a few publicly available datasets suitable for the task.

Shu et al. [197] constructed a dataset by assessing the users' trust level on fake news. More recently, the PAN 2020 competition [198] brought the problem of misinformation spreaders identification to the fore. The dataset of the competition contained 500 users with 100 posts each, for two languages. Giachanou et al. [199] and Mu and Aletras [200] created a dataset containing misinformation and real news spreaders by collecting users that posted articles that have been debunked as fake and built their user history based on their previous posts. We draw inspiration from the method of curation of these datasets and use a similar semi-supervised method to obtain a description of the authors and their context. However, the proposed dataset is distinctive in three aspects: it contains fine-grained labels about (a) the users' credibility and (b) political bias, and (c) it preserves the structure of the threads. Additionally, while the aforementioned datasets utilized Twitter as a source, we utilize Reddit which does not have a word limit on the posts, making the task all the more challenging.

Static Approaches to User Detection

Our dataset preserves the structure of the threads, facilitating the exploration of the users' social activity by modeling it in a graph. The recent advances in graph representation learning [201] in various domains provide a promising, under-explored research direction in the context of fake news spreader detection. More specifically, Graph Attention Networks (GAT) [47] have achieved state-of-the-art-results in various natural language processing tasks [13, 202, 203, 204]. However, this method has not been explored on user graphs in the context of fake news spreader detection. Research has shown that users tend to interact with like-minded individuals [205]. Therefore, we wish to leverage this attribute in order to obtain better user representations through graphs.

Traditional feature-based user modeling methods analyze the users' linguistic patterns in order to infer psycho-linguistic features [206, 207]. These works extract evidence of mental processes through the Linguistic Inquiry and Word Count (LIWC) software in order to tackle the problem of identifying deceptive authors. Certain psycho-linguistic characteristics are assumed to underlie the vulnerability to fake information, therefore the LIWC tool has often been used to investigate the phenomenon of misinformation from both document-level [208, 209] and user-level perspectives [199, 210]. Interestingly, this method has been used in comparison and in conjunction with innovative graph-based architectures [204]. Therefore, we believe that leveraging these psycho-linguistic features and their combination together with the users' social interactions can contribute in order to obtain a strong, competitive baseline.

In addition to the range of mental processes inferred by the LIWC method, the extraction of personality characteristics and disorders lends itself effectively to the execution of the disparate tasks of author profiling, and notably for detecting misinformation spreaders. The Big Five Model

(BFM) has already given promising results in this respect, especially on corpora from microblogging platforms: Twitter in particular [211, 212, 213], but also Reddit [214]. Moreover, in recent literature, the combination of the LIWC and BFM methodology has also been tested [199]. In [210], the authors successfully exploit the concatenation of the two features for detecting fake news spreaders on the dataset proposed for PAN 2020. In this case, the application is bilingual: the two methods are applied to English and Spanish tweets.

Temporal Approaches to User Detection

We argue that the users' characteristics and interactions change dynamically over time due to the dynamic nature of the news cycle, therefore temporal graphs are more suitable to model the evolution of the user-to-user relationships [201]. Our hypothesis, inspired by [205], is that both the social and content similarity patterns of misinformation spreaders differ from those of other users.

The concept of temporal graphs has been around for some years [215, 216, 217] with numerous applications [218, 219, 220]. The most relevant to our work is the model proposed by Sawhney et al. [221], leveraging signals from financial data, social media, and inter-stock relationships via a graph neural network in a hierarchical temporal fashion. We draw inspiration from these approaches and propose a dynamic temporal graph for misinformation spreader detection.

6.3 FACTOID

6.3.1 Terminology

The term *misinformation* in this paper is used specifically in the context of politics as an umbrella term that covers many aspects: (a) *misinformation*: any news that is false or misleading but is not intended as such, (b) *disinformation*: any false or misleading information that is spread with the specific intent of deception, (c) *hyperpartisan news*: news that might not be entirely false, but they are phrased in a way that satisfies a specific political agenda and (d) *satirical news*: any false content that has a humorous intent.

6.3.2 Data Collection

Reddit[2] is an inexpensive source of high-quality data [222]. On Reddit, registered users tend to submit posts with richer content than Twitter, thus we are able to gather enough context for each user. Having enough users with rich contextual density is particularly beneficial for similarity assessment, which makes it the primary choice as the source for collecting disinformation spreaders and real news spreaders post histories.

The data crawling was performed in a user-centric and iterative fashion. To begin with, we manually compiled a list of 65 subreddits regarding controversial political topics that were commonly discussed before the elections, such as general politics or the US presidential race, the SARS-CoV-2 pandemic, women's and men's rights, climate change, vaccines, abortion, gun control, 5G in general. For each of those subreddits, the most recent threads were crawled and inserted into a database. On this data, we performed the first iteration of the URL domain-based disinformation and real news spreader extraction to generate a list of Reddit user accounts with equal amounts of users for either class. We

[2] https://www.reddit.com

Subreddit	**# unlabeled**	**# real**	**# fake**
General political debate			
r/politics (no bias)	2.399.254	81.261	3.869
r/Conservative (right)	346.042	5.165	2.784
r/conservatives (right)	24.310	526	453
r/Republican (right)	17.797	500	256
r/ConservativesOnly (right)	9.431	57	62
r/democrats (left)	11.747	338	41
Other (mostly left)	72.135	2.355	81
SARS-CoV-2			
r/NoNewNormal (anti)	72.411	1.941	1.387
r/LockdownSkepticism (no bias)	62.480	1.441	275
r/NoLockdownsNoMasks (anti)	1.887	82	61
r/Coronavirus (no bias)	92.163	2.753	54
Other (mostly no bias)	21.697	606	53
Women's and men's rights			
r/MensRights (men)	57.654	1.636	501
r/Egalitarianism (non-specific)	83	4	42
r/antifeminists (men)	1.138	44	15
Other (mostly women)	1.399	47	11
Climate change			
r/climateskeptics (questioning)	38.606	756	856
r/climatechange (science)	7.858	622	153
r/GlobalClimateChange (science)	26	2	0
r/climate (science)	120	12	0
Vaccines			
r/DebateVaccines (no bias)	32.635	1.624	637
r/DebateVaccine (no bias)	2.707	57	22
r/TrueantiVaccination (anti)	3.428	48	18
Other (mixed anti and pro)	7.255	225	16
Abortion			
r/prolife (anti)	7.109	167	82
r/Abortiondebate (no bias)	7.590	84	22
Other (mostly pro)	5.228	84	4
Guns			
r/progun (pro)	10.774	453	61
r/Firearms (pro)	12.728	200	33
r/GunsAreCool (pro)	4.930	233	27
r/gunpolitics (no bias)	1.967	61	11
r/guncontrol (anti)	1.062	206	10
Other (mostly pro)	9.744	338	6
5G			
r/5GDebate (no bias)	2.192	19	6

Table 6.1: This table shows the names of the subreddits that belong to each topic and the corresponding number of unlabeled, real, and fake news posts. The rows named "Other" contain the subreddits with a low number of fake news posts for each topic.

then collected the complete histories of all the users in said list, thus gathering all threads in which they participated in the list of political subreddits. All of those threads were inserted into the database from which, again, a now larger list of misinformation and real news spreaders can be extracted. This process was iterated until the dataset reached its current form.

We show the subreddits included in the resulting dataset and the corresponding number of unlabeled, real, and fake news posts they contain in Table 6.1. In the parenthesis, we note the stance that each subreddit supports in its description. For each topic, the subreddits with a very low number of fake news posts, are grouped in the rows named "Other". In this table, the topics are shown based on a descending number of total fake news posts, the same stands for the subreddits that belong to them. For each topic, we opted for an equal distribution of political partisanship and stances, by selecting the same number of the most popular subreddits for each stance and for the same time period.

As we can see, the largest portion of unlabeled, fake, and real news posts are from the subreddit *r/politics* which is a subreddit with no specific political agenda for discussing the news regarding US politics. We can see that the conservative party seems to be posting more frequently based on the number of unlabeled posts. In addition, all topics have a skewed distribution of stances.

6.3.3 Media Domain Lists

Likewise to the work of [223], the website *mediabiasfactcheck.com*[3] was used as the main source for annotated news outlet domains. It was deemed a suitable resource for the study at hand as it offers annotations for two dimensions: the *factuality level* and the *political bias* of a large proportion of highly frequented online news media.

Since we also opted for a binary label for the disinformation spreaders, we created a mapping for those labels. To be considered a disinformation domain, the *mediabiasfactcheck* label has to be below or at *Mixed* factuality level or labeled as satire, while the real news domains have to be at least *Mostly factual* and between *Right-Center* and *Left-Center* political bias.

As for the credibility of the assigned annotations, the maintainers of *mediabiasfactcheck.com* state that they "are looking at political bias, how factual the information is, and links to credible, verifiable sources" [224]. In the description of their methodology, they also describe that they base the labels on reviews of at least 10 headlines and 5 news stories [224].

As a further resource to extend the list of disinformation media sources, an "index of fake-news, clickbait, and hate sites" [225] by the *Columbia Journalism Review*[4] was consulted. Its curators state that it was created by merging pre-existing fake news domain lists from various sources and then checking their actual invalidity with the fact-checking platforms PolitiFact and Snopes [225]. Finally, to ensure the quality of all annotations, we cross-matched the labels of the common domains by consulting both Snopes and Media Bias/Fact Check.

In total, in this way, we aggregated 1577 disinformation and 571 real news domains for our ground truth and post-level annotations.

6.3.4 Binary Annotation.

The users were annotated as *misinformation spreaders* and *real news spreaders* based on the posted web links in their history. More precisely, we first extracted news links from the users' posts using

[3] https://mediabiasfactcheck.com
[4] https://www.cjr.org

regular expression matching. To decide whether the extracted link was counted as misinformation or real news, its domain was matched with the two lists of domains of online news outlets, each corresponding to one class. Users were then labeled as *misinformation spreaders* if they had at least two detected misinformation links in their post history, while for being *real news spreaders* they had to have no shared links from the misinformation list and at least one link posted from the factual news list.

6.3.5 Fine-grained labels.

In addition to the binary separation of users into misinformation spreaders and real news spreaders, each user was annotated with the following factors by averaging over a float mapping of the labels from *mediabiasfactcheck.com*, for a more fine-grained annotation.

Factuality degree (fd). This factor represents the average level of factuality of each author, and is also in the range of [-3, +3] with each label corresponding to different scales; very low ($s_{vl} = -3$), low ($s_{lf} = -2$), mixed ($s_{mx} = -1$), mostly factual ($s_{mf} = +1$), high ($s_{hf} = +2$), very high ($s_{vh} = +3$). Similarly, the factuality factor of each author is computed as follows:

$$fd = \frac{\sum_{\ell} s_{\ell} \cdot N_{\ell}}{\sum_{\ell} N_{\ell}}$$

where N_{ℓ} in the number of posts labeled as $\ell \in [vl, lf, mx, mf, hf, vh]$

Political bias (pb). This factor represents the level of partisanship and is a number in the range of [-3, +3] where each of the labels corresponds to different scales (s_{ℓ}); extreme left ($s_{el} = -3$), left ($s_{l} = -2$), center left ($s_{cl} = -1$), least biased ($s_{lb} = 0$), center right ($s_{cr} = +1$), right ($s_{r} = +2$), and extreme right ($s_{er} = +3$). The political bias of each author is computed as:

$$pb = \frac{\sum_{\ell} s_{\ell} \cdot N_{\ell}}{\sum_{\ell} N_{\ell}}$$

where N_{ℓ} in the number of posts labeled as $\ell \in [el, l, cl, lb, cr, r, er]$

Science belief (sb). This factor quantifies the level of belief in science and is a number in the range of [-1, +1] where each of the labels corresponds to different scales (s_{ℓ}); conspiracy theory article ($s_{c} = -1$), science-based article ($s_{s} = 1$). Similarly, the science factor of each author is computed as follows:

$$sb = \frac{\sum_{\ell} s_{\ell} \cdot N_{\ell}}{\sum_{\ell \in fl}}$$

where N_{ℓ} in the number of posts labeled as $\ell \in [s, c]$

Satire degree (sd). This factor represents the level of satire in the fake news posts. The higher this factor is, the less intentional the misinformation spreading. It is in the range of [0, 1] and is computed

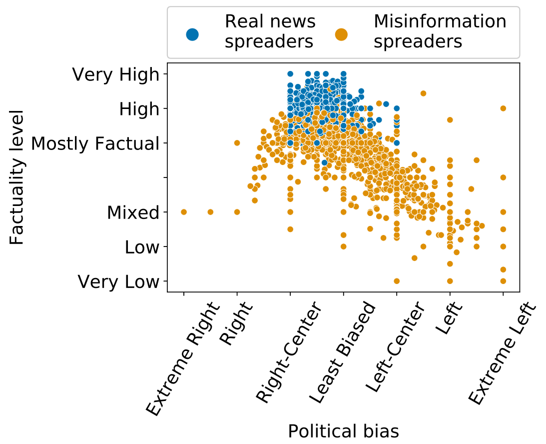


Figure 6.2: Factuality factor over political bias of each user.

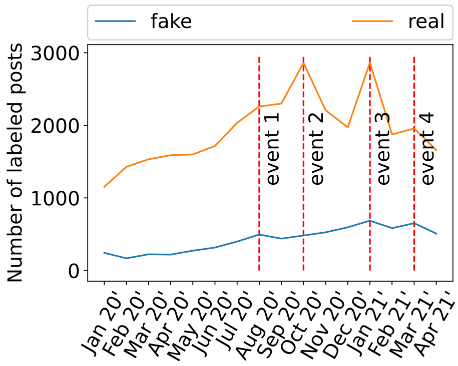


Figure 6.3: Number of fake news posts and real news posts associated with the political events from Table 6.2

as the number of satire posts N_s divided by the number of fake news posts N_{fn}:

$$sd = \frac{N_s}{N_{fn}}$$

Discussion. Current datasets for fake news spreaders detection characterize a user as a fake news spreader based on whether they posted more than n number of posts, with n being an arbitrary number around two or three. By introducing these fine-grained labels we pose some interesting questions to the research community. How many times should a user post about fake news in order to be considered as a fake news spreader? Should it also depend on what kind of fake news post they posted (e.g. a post from a pseudoscience source vs post from a source that has a mixed factuality reporting shouldn't have the same gravity). While satirical news is fake, the intent is usually humorous, however, the dissemination of such news could be equally harmful. Should users who post from these sources also be considered as fake news spreaders? Should we consider a threshold of factuality degree instead of counting fake news posts to separate fake news posters and real news spreaders?

6.3.6 Dataset Statistics

The dataset comprises a total of 3.354.450 posts authored by 4,150 users with a class distribution of 74:26 of real news and fake news spreaders respectively, collected from January 2020 to April 2021. Misinformation spreaders had an average of 1240 posts, with this count being at 654 for the real news spreaders. In total, 2% of the posts contained links to real news media, while 0.3% pointed to domains from the misinformation list.

Using the post-level annotations from Section 6.3.5, the political biases of the users can be looked at: 41.17% of the users that have left-wing political bias are misinformation spreaders, while 58.82% of them are real news spreaders. 91.58% of the users that have right-wing political bias are fake news spreaders, while only 8.41% of them are real news spreaders. Figure 6.2 depicts the factuality factor over the political bias of each user. While there is an apparent correlation (Pearson correlation of -0.45) between the political bias and factuality of the users, it is important to note that this effect is not an isolated case or a problem that arises from the process of collecting our data, in fact, this phenomenon has been observed by many researchers [226] who show that there is indeed a high correlation between the perceived bias of a publisher and the trustworthiness of news content. In addition, [227] showed that US conservatives are uniquely susceptible to misinformation regarding political events, and generally political extremes (both the left and the right) are substantially susceptible to conspiracy beliefs. Note that from Table 6.1, we can see a higher posting activity from the right-wing party compared to the left-wing, which leads us to the conclusion that right-wing supporters might be more active in social platforms compared to left-wing supporters.

Date	Event Description
Feb 5	Trump is acquitted on the charges of abuse of power and obstruction of Congress.
Jul 11	Mail-in votes are encouraged.
Jul 30	Donald Trump threatens to postpone the election if it appears mail-in votes might go against him. (We regard this as if this had happened in August, since the effects of this political event would be still discussed during that month)
Aug 11	Joe Biden chooses Senator Kamala Harris (D-CA) as his running mate (event 1)
Nov 3	2020 United States elections (event 2)
Jan 6	US Capitol is attacked by supporters of Trump (event 3)
Feb 24	Johnson & Johnson's vaccine candidate receives emergency use authorization from the FDA (event 4)

Table 6.2: Major political events coinciding with the peaks observed in the number of fake and real news posts from Figure 6.3

The timestamps and thread structure of all stored posts are preserved in the dataset, in order to encourage a more comprehensive analysis of the users and their posting behavior. Figure 6.3 shows the number of fake news and real news posted per month. We also provide a list of pivotal political events[5] that happened during this time period in Table 6.2. We can see that these events coincide with the increase in the number of fake news and real news posts. We can see an obvious increase in real news right until the US elections and a sudden increase during the attack on the Capitol. This is logical since the elections were scheduled and discussed months before they happened while the attack was an event that developed over a few days. A smoother curve is observed for the fake news, where the numbers do seem to fluctuate in the same manner during these events, but not to the same degree.

6.3.7 Additional Analysis

This subsection delves into the analysis of linguistic features, automatic topic detection, and the examination of selected topics' impact on the dataset. The dataset has a temporal focus on the 2020 US elections, however, due to the user-centric and connected post-history scraping of users, the total period covered is much greater. Figure 6.4 illustrates a moving average of post-publishing times partitioned by topic groups. Despite truncating the x-Axis in 2018, the dataset includes posts dating

[5] https://en.wikipedia.org/wiki/2020_in_United_States _politics_and_government,
https://en.wikipedia.org/wiki/2021_in_the_United_States

back to 2014. These plots, highlight a clear peak around the period of US elections. Furthermore, we observe that all topics in the dataset have a similar distribution in time. Additionally, we note that *SARS-CoV-2* had its first posts in early 2020 which is very reasonable considering the start of the pandemic.

The dataset also exhibits considerable variation in the number of posts annotated per user, ranging from individuals with a single post to those with over a thousand. Specifically, 14.8% of all users contributed fewer than three annotated posts. Additionally, Figure 6.5 reveals a significant number of misinformation spreaders among those with a minimal number of labeled contributions.

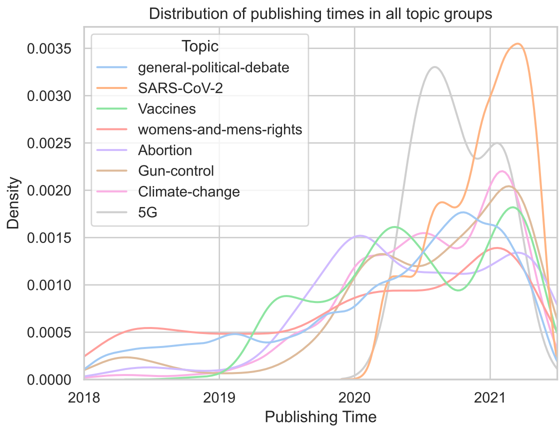


Figure 6.4: Distribution of post publishing times in topic groups.

In Table 6.1 we notice that some topic groups are heavily imbalanced. Figure 6.6 illustrates the distribution inside the topic group 'Guns'. We can notice that over 90% of the posts originate from pro-gun communities. Furthermore, we can spot that the subreddit 'r/liberalgunowners' consists of mostly real news whereas posts from 'r/progun' have a higher percentage of misinformation. Other topic groups e.g. 'Abortion' show a different distribution.

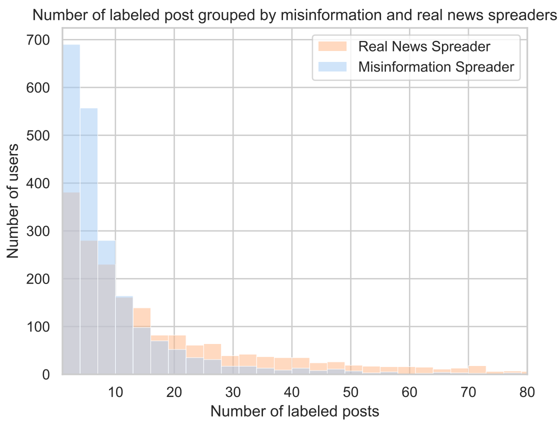


Figure 6.5: Distribution of annotations per user.

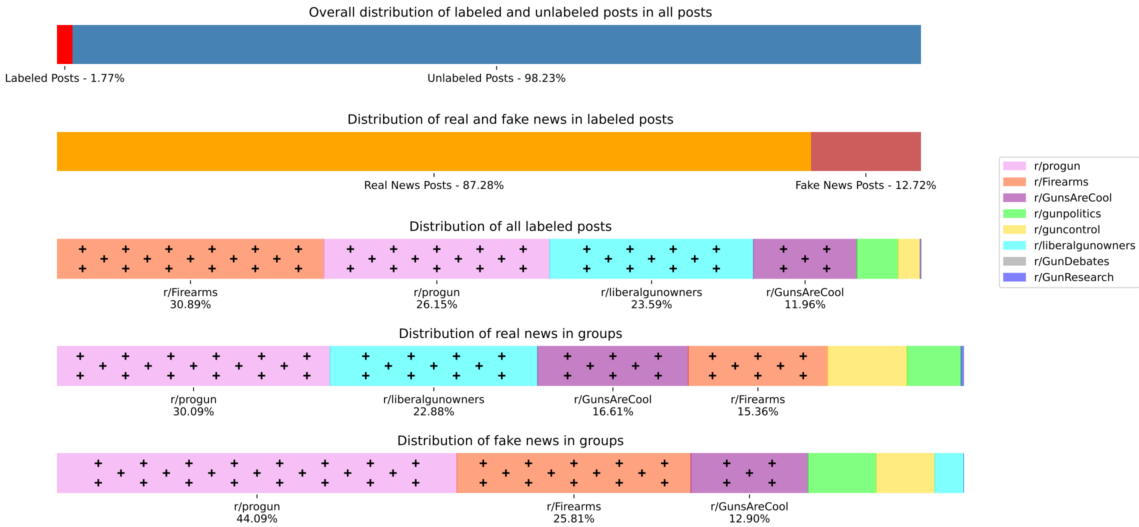


Figure 6.6: Distributions inside the topic group 'Guns'

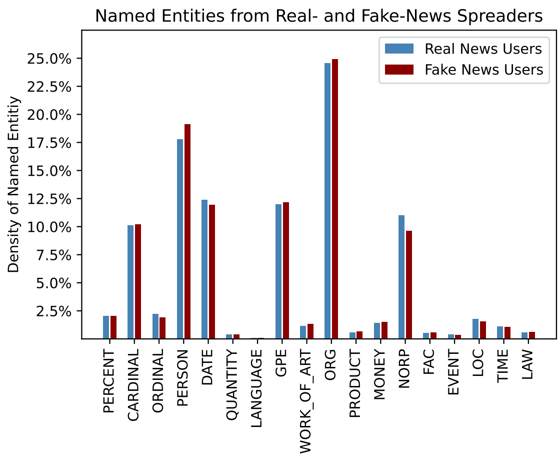


Figure 6.7: Named Entities in Real Misinformation Posts.

Named Entity Recognition

In our initial textual analysis, we conducted Named Entity Recognition (NER) on all labeled posts within the dataset. Anticipating minimal disparities between the entities mentioned by spreaders of real news versus misinformation, this analysis aimed to serve as a preliminary check for any significant anomalies in the data annotations. Figure 6.7 illustrates the densities of named entities identified, utilizing spaCy[6].

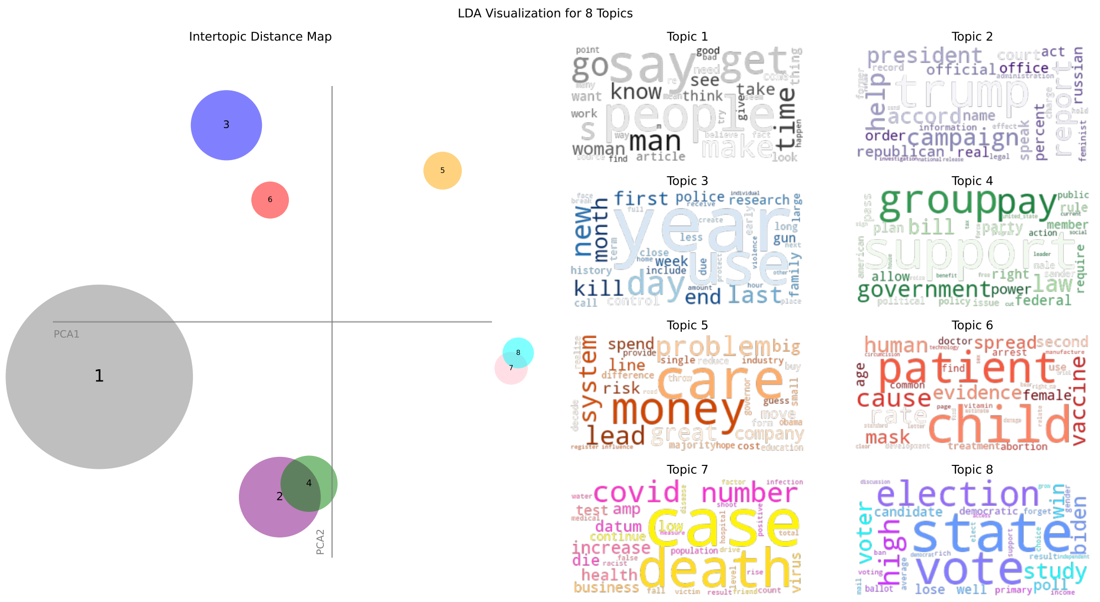


Figure 6.8: Topics found by LDA,

Latent Dirichlet Allocation

The subreddit groups utilized in FACTOID dataset collection, may not represent the actual topics that the users talk about. In order to analyze which topics are most relevant inside the dataset we employ Latent Dirichlet allocation (LDA). The computation was done with the python package Gensim [7]. Upon analyzing the dataset for nine topics, we noticed that the topic with the number 9 is a subset of topic 7. This observation led to the conclusion that eight groups will be enough to understand the major topic groups. In Figure 6.8 we show the topic position in 2D space for eight topic groups, based on the two principal PCA features, along with word clouds based on the keywords identified by LDA.

Given that a major portion of the posts falls under the 'General political debate' we can also observe that topics 1, 2, 4, and 8 have similarities to topics associated with this group. In contrast, minor topic groups like '5G' or 'Climate Change' do not seem to be present. Moreover, topic 3 could be mapped to 'Guns' while topics 6 and 7 seem to be related to 'SARS-CoV-2' and topic 5 to 'Women's and men's rights'.

Topics in Embeddings

We aim to explore the impact of different embedding approaches on the influences of post-clustering. Our analysis revealed that embeddings derived from SBERT model, are particularly sensitive to the subject of a post, suggesting that models based on heavily imbalanced topic groups might prioritize learning about the topics themselves rather than the structure of misinformation. In Figure 6.9, we illustrate the clusters using two different embeddings, the top plot illustrates the

[6] spaCy - Industrial-Strength Natural Language Processing https://spacy.io/
[7] https://radimrehurek.com/gensim/

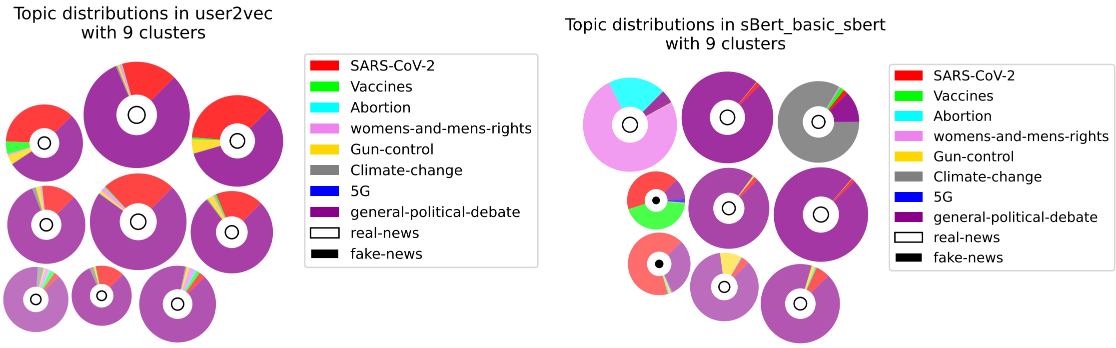


Figure 6.9: Topic clusters in different post embeddings. User2Vec (Top) and SBERT fine-tuned to the FACTOID Dataset (bottom), both set to 9 clusters.

user2vec approach to extract embeddings while the bottom plot utilizes a non-fine-tuned SBERT embedding. Each circle represents one cluster and the distribution of topics in that cluster, while the size of a circle is proportional to the cluster size. Lastly, the inner circle is filled if the cluster consists of mostly misinformation posts. In addition, the shade is proportional to the inter-cluster similarity, meaning the alpha value is equal to one if all posts are either misinformation or real news and 0.5 (minimum value set by me) if the cluster has a 50:50 split of real news and misinformation.

In the user2vec plot, we can observe that all clusters carry posts from all topic groups. Furthermore, the proportion of topics is almost the same in each cluster and equal to the overall proportion in the dataset. On the contrary, the SBERT plot contains clusters with different distributions. For instance, we can identify an Abortion/woman's and men's-rights cluster in the top left corner, that are probable to have overlaps in discussed topics. Additionally, we find a Climate-Change cluster in the top right. The left middle and bottom clusters are both mostly SARS-CoV-2 and Vaccine clusters, which is relevant given the timeframe of the dataset. These observations underscore the importance of topic balance when employing a SBERT approach in future works.

Beyond cluster composition on various topic distributions, our focus extends to evaluating cluster quality from a misinformation detection perspective. Heatmaps, with embeddings along the y-axis and cluster numbers ranging from two to eleven on the x-axis, facilitate this assessment. The quality of each cluster is measured based on their homogeneity regarding misinformation, given from the following formula:

$$\frac{2 \cdot \max\{\#RN, \#MISINF\}}{\#cluster} - 1. \tag{6.1}$$

$\#RN$ is the number of real news users and $\#MISINF$ is the number of misinformation users. This ensures that every cluster is assigned a score between zero and one, where zero means that the cluster carries the same number of misinformation and real-news spreaders, and one means that the cluster is either consisting completely of misinformation or real news spreaders.

The clustering was performed using k-means. In general, the SBERT-based clusters outperform other methods. Furthermore, we observe that embeddings on a user level have a better quality when the embedding model is fine-tuned on the post level. Moreover, fine-tuning on other misinformation datasets such as Fever [228], results in an improvement in embedding quality when compared to the base SBERT model.

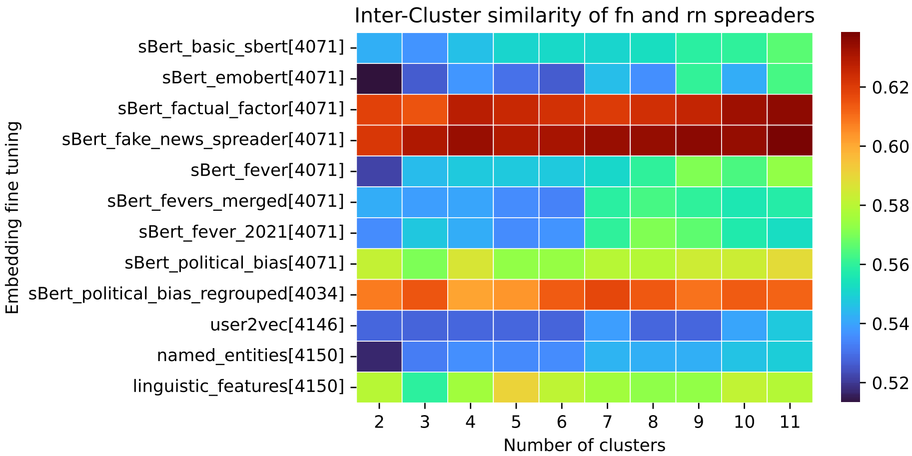


Figure 6.10: Heatmap of cluster qualities.

6.4 Static Analysis

6.4.1 Problem Formulation

We denote the user to be classified as $u^i \in \mathcal{U} = \{u^1, u^2, \ldots, u^N\}$. Each user u^i is associated with a posting history $\mathcal{H}^i = \{(p_1^i, t_1^i), (p_2^i, t_2^i), \ldots, (p_{L^i}^i, t_{L^i}^i)\}$ where p_k^i is a text authored by the user u^i, posted at time t_k^i where $t_1^i < t_2^i < \cdots < t_{L^i}^i$ and L^i is the individual posting history length of each user u^i.

Fake news spreader detection. For the following experiments, we utilize the binary labels introduced in Section 6.3.4. We therefore formulate the author profiling problem as a binary classification task to predict the class y^i of the user, where $y^i \in$ {misinformation spreader, real news spreader}.

Political bias identification. We utilize the fine-grained labels of the political bias introduced in Section 6.3.5. The left-wing supporters are the users with $pb < -0.5$, while the right-wing supporters are those with $pb > 0.5$. Accordingly, the identification of partisanship is defined as a binary classification task to predict the class y^i of the user, where $y^i \in$ {left wing, right wing}.

6.4.2 User Representations

BERT-based representations We use Sentence-BERT (SBERT) [44] to obtain the embedding e_k^i of each user's individual historical posts p_k^i. SBERT is a modification of BERT that is specifically designed to produce semantically meaningful sentence embeddings, and has achieved state-of-the-art performance on various challenging datasets [229, 230, 231], rendering this encoding method particularly suitable for representing the users.

We want to encode the users' historical context $\mathcal{H}^i$ by obtaining their user representations $E^i \in \mathbb{R}^{d_b}$. [232] empirically showed that simple average sentence embeddings compare favorably to more complex methods. Each user's historical encoding is averaged over the individual posting history length of each user L^i, as:

$$E^i = \frac{1}{L^i} \sum_{k=1}^{L^i} e_k^i$$

By utilizing this encoding method we are going to be able to capture the linguistic similarities that exist between the users' posting history. In addition, we masked the links so that the similarity is not attributed based on the links but on the text/commentary that the user has posted regarding the news.

User2Vec In addition, we also adopt User2Vec [39] to compute the initial user representation. $E^i \in \mathbb{R}^{d_u}$ of user u^i based on their corresponding historical posts $\mathcal{H}^i$, optimizing the conditional probability of texts given the author (see Section §2.2).

Encoding the psycho-linguistic features In order to analyze the relationship between users' tendency to spread fake news and their personality traits and mental processes, we use the Big Five Model and LIWC software respectively. The two methodologies are described hereafter.

The Big Five Model (BFM) [233] assumes that human personality can be summarized in five main aspects: *(i) openness to experience*, *(ii) conscientiousness*, i.e. the interactions between rational thought and instincts, *(iii) agreeableness*, or the intensity of individuals' reactions within the social context, *(iv) extraversion*, and *(v) emotional instability*. After defining these basic dimensions, this approach argues for the existence of semantic associations between them and specific sets of adjectives that are recurrent in the natural language when describing individuals' psychological traits. Accordingly, [234] derive a personality score with the following process: for each factor, they compute the mean of all the cosine similarities between the embedding representations[8] of every word in the input text and every benchmark adjective empirically observed as to be able to encode that precise personality trait; the higher this average similarity, the greater the evidence of a given factor. Neuman and Cohen also included 9 extra factors describing mental disorders, like *paranoia*, and *narcissism*.

The Linguistic Inquiry and Word Count software (LIWC) [206] applies a lexicon-based method for mapping the text into 64 psycho-linguistic categories defined to obtain evidence of many mental processes underlying the natural language, and grouped into 2 macro-categories: *(i) linguistic dimensions*, i.e. function words, common verbs and adjectives, etc. and *(ii) psychological processes* of many kinds, including the affective, cognitive, and social type. In conclusion, the LIWC representation of one document consists of the set of relative frequencies for the categories, according to the number of words identified in the text that are associated with each of them. Again, both psycho-linguistic encodings are achieved by an averaging operation over the post-level ones. In particular, it was preferred to extract the values of the LIWC features as means of the relative frequencies at the post level in order to extract the average incidence of each category within the single publication, with the

[8] The word embeddings are produced by a Word2Vec architecture, pre-trained on the Google News Corpus.

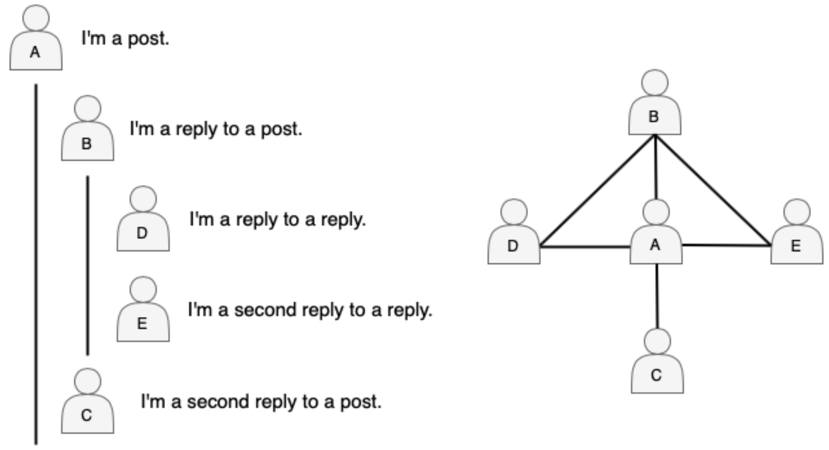


Figure 6.11: Transforming a post/reply tree in social media into a social graph network.

aim of avoiding the calculations being biased towards the most frequent classes within the composition of the global user discourse.

6.4.3 Graph Modeling

Social science argues that like-minded people tend to interact more with each other [205], therefore we construct the social graph in a way that captures the users' social interactions with each other. We define as social interaction the replies and mentions in a post thread. For each thread of posts, we connect the chain of replies to the root (i.e. the original post) of the conversation and all mentions/replies to each other. Following, these connections are translated to user connections in the social graph. This method is more clearly depicted in Figure 6.11. The social graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is comprised by a set of user nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ between these users.

Graph neural networks are able to leverage the semantic and social relations between users [201]. As users have a different influence on one another, we need to focus on users that have more relevant connections with higher influence. To model the gravity of the influences of the neighborhood to a node, we use Graph Attention Networks (GAT) [47]. GAT attends to the neighborhood of each user and assigns an importance score to the connections that contribute more to the detection of misinformation spreaders. The input to a GAT layer is a set of user embeddings $\mathcal{E} = \{E^1, \ldots, E^N\}$ where $N = |\mathcal{U}|$. A GAT layer produces updated features, $\widetilde{\mathcal{E}} = \{\widetilde{E^1}, \ldots, \widetilde{E^N}\}$, where $\widetilde{E^i} \in \mathbb{R}^{d_g}$.

Classification Layer

The overall learned representations for each user, are forwarded into a linear layer parameterized by a weight matrix $\mathbf{W^o} \in \mathbb{R}^{d_o \times d_r}$. The final prediction is computed as:

$$\hat{y} = softmax(\mathbf{W^o}\,\Gamma(\overline{h})). \tag{6.2}$$

Given the true label y for a user, we use cross-entropy loss to calculate the loss L as follows:

$$L = -\sum_{i=1}^{N} y_i\, ln(\hat{y}_i) + (1 - y_i)\, ln(1 - \hat{y}_i). \tag{6.3}$$

6.4.4 Experiments

Models used

We compare our graph-based model as described in Section 6.4.3, with a Support Vector Machine (SVM), Logistic Regression (LogReg), and a Random Forest (RnFor) classifier which are trained by using the following features:
UBERT: We use the SBERT embeddings of the documents averaged over the user's history as feature vectors.
User2Vec: To initialize the user feature vectors, we use User2Vec over the vocabulary of each user during their history.
Psycholing: We concatenate both LIWC and BFM features, to compute an initial feature vector for the users.

Performance evaluation and ablation study

Table 6.3 shows GAT's F_1 score on the Reddit dataset for the fake news spreader detection task. We compare the graph-based results by using different initialization methods, namely UBERT, User2Vec, psycho-linguistic, concatenation of User2Vec and psycho-linguistic features, and random vectors. Interestingly, the proposed model achieves the best performance by utilizing User2Vec, despite having lower dimensionality than UBERT. This is mainly attributed to the fact that User2Vec embeddings were obtained based on this dataset, while UBERT was pre-trained on a general-use corpus. The psycho-linguistic features, on their own, perform rather poorly with GAT, and concatenating them to User2Vec does not contribute to the performance. However, the psycho-linguistic features perform comparably to UBERT in the non-neural baselines, which is in line with the observations of [235].

Fake News Spreader Detection

Model	**F_1 score**
GAT + User2Vec (200)	**61.6%**
GAT + UBERT (768)	61.2%
GAT + Psycholing (83)	53.6%
GAT + User2Vec + Psycholing (283)	59.4%
GAT + Random (200)	47.8%

Table 6.3: Comparison of different user embedding techniques for the GAT model on the fake news spreader detection task. Reported values are the F_1-scores over a 5-fold Cross Validation. Bold denotes the best overall performance on the task.

Table 6.4 shows the F_1 score of the baseline models for both the political bias and fake news spreader detection tasks. For the political bias identification task, UBERT consistently obtains better results than User2Vec, and achieves the best result with SVM. On the other hand, for the Fake news spreader detection task, we observe the reversed behavior. User2Vec consistently obtains significantly better results than UBERT, and achieves the best result with a Random Forest classifier.

	Political Bias		**Fake News Spreader**	
Model	**UBERT**	**User2Vec**	**UBERT**	**User2Vec**
SVM	**66.2**%	63.0%	53.9%	61.1%
LogReg	64.7%	62.8%	58.6%	59.8%
RnFor	64.9%	63.5%	49.7%	**61.3**%

Table 6.4: Comparison of different user embedding techniques for the baseline models for both political bias and fake news spreaders detection. Reported values are the F_1-scores over a 5-fold Cross Validation. Bold denotes the best overall performance on the task.

Table 6.5 shows the ablative results of the psycho-linguistic features on the Reddit dataset for both political bias and fake news spreaders detection. In general, psycho-linguistic features show a significantly higher effectiveness in distinguishing users on the basis of political bias. Detected mental processes appear to be significantly more useful than personality factors: this result is coherent with the study conducted through the LIWC software by [236] about the link between political ideology and language use. The most relevant mental process is the *affective* kind, which correlates negatively with the target class, suggesting that right-biased users tend to express fewer emotions such as anxiety, anger, and sadness in the text. As regards the other task, the BFM encoding appears slightly more effective for identifying fake news spreaders. Indeed, since personality regulates the behavior in real contexts, it is reasonable to assume it to be also influential within virtual communities. The dominant factor is here the *openness to experience*: as expected, in those who spread fake news, there is greater rejection or less curiosity towards ideas outside their belief system. Also, the *schizotypy* disorder appears relevant, consistent with previous empirical observations [237].

We note that the psycho-linguistic features are not adaptive to the tasks since they are lexicon-based, therefore the embedding-based features achieve significantly higher F_1 scores in the political bias detection task. By comparing all results for the fake news spreader detection task, we observe that the GAT model outperforms all baselines. Therefore, social interactions constitute a promising tool for predicting the behavior of unseen users.

	Political Bias			**Fake News Spreader**		
Model	**LIWC**	**BFM**	**Both**	**LIWC**	**BFM**	**Both**
SVM	55.1%	38.8%	61.0%	56.2%	51.0%	53.9%
LogReg	63.6%	51.5%	**63.9**%	58.3%	55.1%	58.3%
RnFor	56.6%	54.8%	61.7%	55.9%	**58.4**%	54.8%

Table 6.5: Ablation study over the psycho-linguistic features and their combination for both political bias and fake news spreaders detection. Reported values are the average F_1-scores over a 5-fold Cross-Validation. Underlines denote the best result for the combination of features considered, while bold denotes the best overall performance on the task. 'Both' indicates the concatenation of both representations.

Analysis of the linguistic differences

To get an intuition for the actual linguistic differences between the two user groups of misinformation spreaders and real news spreaders, we extracted the learned token weights from the SVM model in order to study the predictiveness of the tokens for each class [238]. The most predictive tokens are shown in Table 6.6. It can be seen that there's a tendency for misinformation spreaders to reference politically left-leaning groups as "liber", "dem", "left" or "blm" (referring to the Black Lives Matter movement), while real news spreaders use the terms "fascist" and "republican" with higher frequency.

Label	Tokens
Misinformation Spreaders	china, video, come, offici, blm, corrupt, media, away, liber, order, new, trump's, seem, wrong, kill, left, dem, riot
Fact Checkers	public, first, week, understand, trial, fascist, republican, war, one, forced-birth, health, pleas, power, let, shock, view, service

Table 6.6: Top-ranked tokens for each label.

6.5 Temporal Analysis

6.5.1 Temporal Graph Construction

Encoding Users

Each user u^i is associated with a posting history $\mathcal{H}^i$. We partition the complete posting time period in equal discrete time frames τ, containing the users' posts that were posted within these time frames.

User2Vec. We adopt User2Vec [39] to compute each user's representation $E^i_\tau \in \mathbb{R}^{200}$ based on their corresponding historical posts within the time frame τ, by optimizing the conditional probability of texts given the author.

UBERT. In addition, we use Sentence-BERT (SBERT) [44] to encode each user's individual historical posts, and we obtain each user's temporal historical encoding $E^i_\tau \in \mathbb{R}^{768}$ by averaging over the posting history length within a corresponding time frame τ.

Individual graph construction

We model the user's temporal relationships by constructing a sequence of graphs $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T$ corresponding to each time frame τ. Each graph $\mathcal{G}_\tau$ is comprised by a set of user nodes $\mathcal{V}_\tau$ that have posted at least once within the time frame τ and a set of edges $\mathcal{E}_\tau$ between these users. We construct the following types of graphs.

Semantic graph. The user embeddings E_{τ}^{i} represent each user's context within the time period τ. Users with semantically similar content are close in the vector space [44] since they have similar context encoding. To construct the users' semantic graphs $\mathcal{G}_{\tau}^{sem} = (\mathcal{V}_{\tau}, \mathcal{E}_{\tau}^{sem})$, we calculate all the pairwise cosine similarities between the users' embeddings within a time period τ; $cos(E_{\tau}^{i}, E_{\tau}^{j})$. We form connections between two users only if their cosine similarity is above a high threshold θ, representing the semantic similarity between two users.

Social graph. On Reddit, users engage in various discussions with their peers. Social science argues that like-minded people tend to interact more with each other [205], therefore, for the FACTOID dataset, we are able to construct the social graph $\mathcal{G}_{\tau}^{soc} = (\mathcal{V}_{\tau}, \mathcal{E}_{\tau}^{soc})$ in a way that captures the users' social interactions with each other. We define as social interaction the replies and mentions in a post thread. For each thread of posts, we connect all the chain of replies to the root (i.e. the original post) of the conversation and all mentions/replies to each other. Next, these post connections are translated to user connections in the social graph (see Figure 6.11).

6.5.2 Temporal Analysis of Graphs

Centrality. Figure 6.12 depicts the graph centrality normalized by the number of posts. This metric helps in identifying important nodes in a graph. We can see that, in the linguistic graph, the centrality of the misinformation spreaders and real news spreaders follows a similar pattern but fluctuates a lot over time. Interestingly, there's an obvious increase in the centrality of both classes during August, right after former President Trump announced the possibility of postponing the US elections (see Table 6.2). This increase is more obvious in the misinformation spreaders, meaning that they are discussing a particular topic more extensively compared to the real news spreaders. In the social graph, we observe a great difference in the values of centrality between misinformation spreaders and real news spreaders. This metric shows that misinformation spreaders are gathered in the center of the graph, while real news spreaders are in the periphery of the graph and are not that densely connected to each other. This essentially indicates that misinformation spreaders form a densely connected "community" and marginalize real news spreaders. The centrality of the misinformation spreaders decreases over time, while in the case of real news spreaders, it fluctuates but still stays within a specific range. This apparent dynamically changing behavior of the nodes supports our choice of temporal modeling of the graphs.

Homophily. In Figure 6.13, we show the amount of homophily observed for both semantic and social graphs, which is defined as the percentage of edges that connect users with the same label. Interestingly, we observe that in the semantic graph the homophily follows different patterns in misinformation spreaders and real news spreaders, and it fluctuates over time. In the social graph, the misinformation spreaders have consistently higher homophily than real news spreaders, which means that they tend to interact and exchange opinions more with each other compared to real news spreaders. These results complement the edge analysis from Section 6.5.2 which shows that users from the same credibility group tend to socially interact more with each other, which is more apparent in misinformation spreaders.

Connections' percentage First, we study the temporal evolution of the users' semantic similarities and social interactions between different groups of users over time and associate those temporal

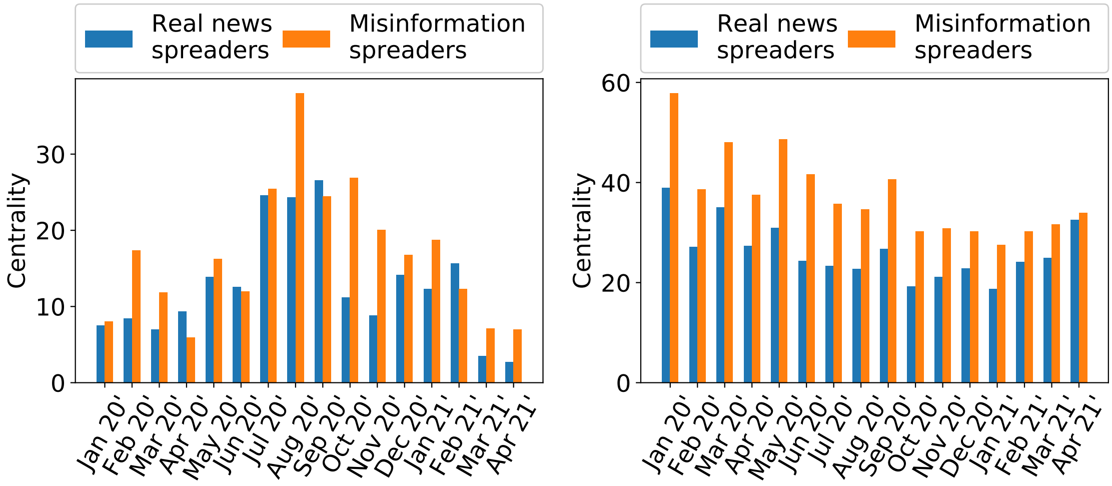


Figure 6.12: Approximated (k=1000) graph centrality normalized by post amount calculated for all time spans for the semantic (left) and social (right) graph.

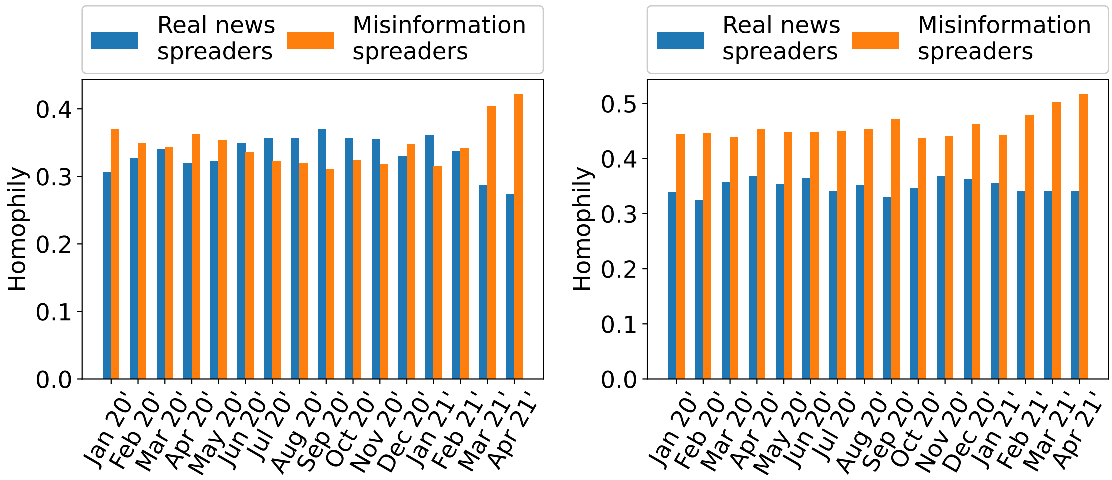


Figure 6.13: Amount of homophily observed through time for both semantic (left) and social graph (right).

fluctuations with the political landscape. We group the users by their credibility label (misinformation spreaders, real news spreaders) and define three different *edge types*: (1) edges between misinformation spreaders ('m2m'), (2) edges between real news spreader ('r2r') and (3) edges between misinformation spreaders and real news spreaders ('m2r'). We partition the users' total posting period (from the start of January 2020 until the end of April 2021) into 16 monthly time periods, and we compute the connections' percentage within each time period for all edge types. The connections' percentage can be interpreted as the normalized edge count of a particular edge type during a time period τ. We define the connections' percentage of a certain edge type as $\rho_{\text{edge type}} = r^{(\tau)}_{\text{edge type}}/R^{(\tau)}_{\text{edge type}}$, where $r^{(\tau)}_{\text{edge type}}$ is the number of edges (of that edge type) that exist between two users during the time period τ and $R^{(\tau)}_{\text{edge type}}$ is the number of all possible connections (of that edge type) at the time period τ, computed

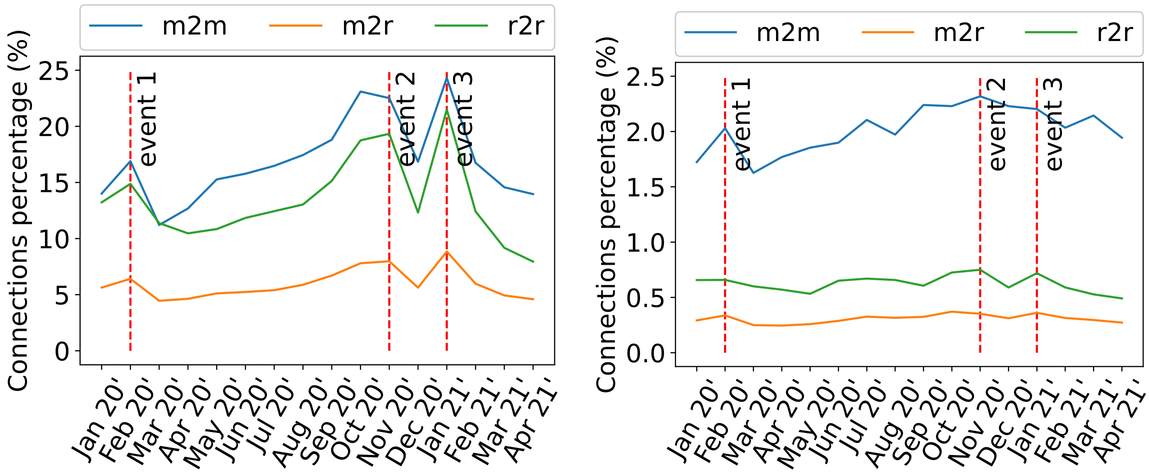


Figure 6.14: Connection percentage of per month for the semantic (left) and social graphs (right). The events shown in this figure correspond to the events mentioned in Table 6.2.

as follows:

$$
\begin{aligned}
R_{m2m}^{(\tau)} &= N_m^{(\tau)}(N_m^{(\tau)} - 1)/2 \\
R_{r2r}^{(\tau)} &= N_r^{(\tau)}(N_r^{(\tau)} - 1)/2 \\
R_{m2r}^{(\tau)} &= (N_m^{(\tau)} + N_r^{(\tau)})(N_m^{(\tau)} + N_r^{(\tau)} - 1)/2
\end{aligned}
$$

where $N_m^{(\tau)}$ is the number of misinformation spreaders and $N_r^{(\tau)}$ is the number of real news spreaders that have posted at least one post at time period τ.

For the temporal semantic graphs, an increase in this metric essentially shows an increase in the language usage similarity between different user groups. Correspondingly, for the social graphs, an increase would show that two user groups engage in discourse and share opinions in a thread.

Can we detect different temporal relationship patterns depending on the users' credibility?

Figure 6.14 depicts the connections' percentage on the semantic graph and the social graph. For both graphs, we can observe that the 'm2r' connections percentage is consistently the lowest for all time periods, indicating that on an aggregate level, misinformation spreaders and real news spreaders do not have as much context similarity to each other and avoid socially interacting with each other. On the other hand, misinformation spreaders seem to be more densely connected with each other and tend to exchange information regularly.

How do the users' temporal semantic and social relationships fluctuate based on the political scene?

Interestingly, we observe peaks in the connections' percentage during January 2020 (event 1), November 2020 (event 2), and January 2021 (event 3) for both graphs. The percentage fluctuations are more obvious in the semantic graph compared to the social graph, this is the first indication that the temporal context similarities might be more useful for the model compared to the social interactions. We provide a list of pivotal political events in Table 6.2 which evidently explains the increase in the connections' percentage and provides an intuition behind the users' behavior.

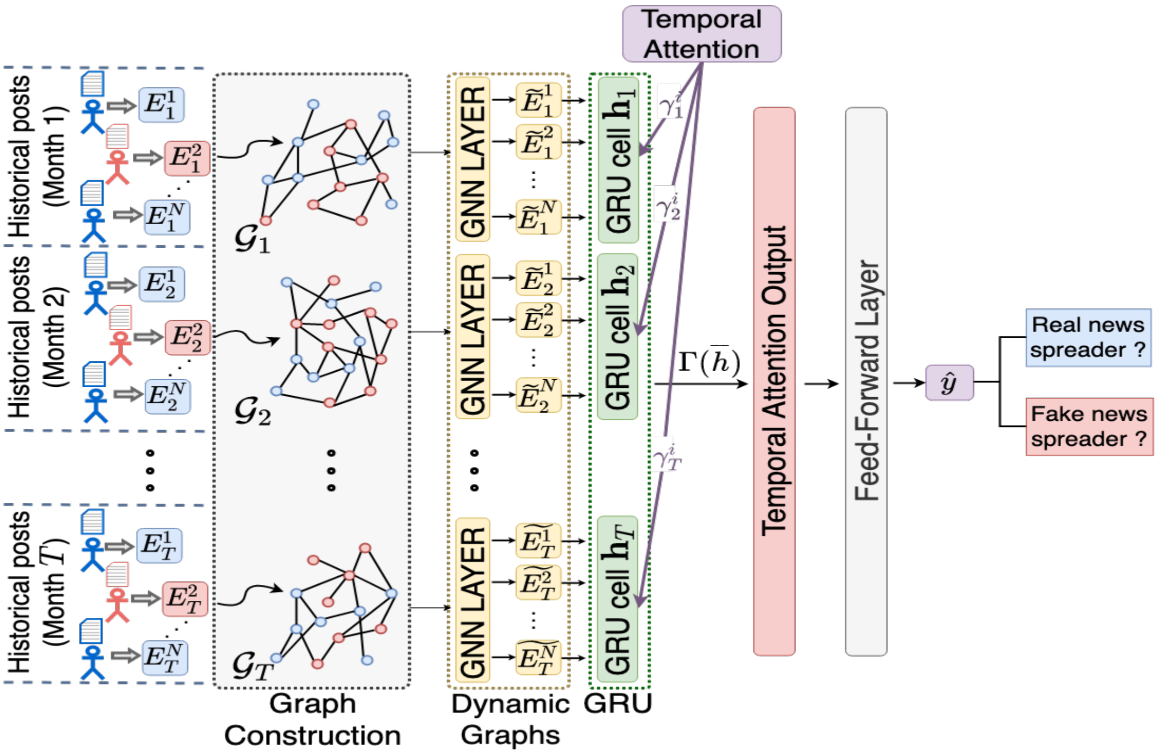


Figure 6.15: Overview of the proposed framework. We first obtain the user embeddings for each time frame and construct the temporal graphs. Next, we feed the graphs to a GNN to extract neighborhood features. For each user, we use a GRU with temporal attention to compute an overall representation of the user, which is finally forwarded to a classification layer.

6.5.3 Methodology

Graph Neural Network Layer

We utilize three different types of Graph Neural Network (GNN) layers in order to demonstrate the robustness and predictability of the users' connections. The input to the GNN layer is a set of user embeddings E_τ^i for each time frame τ. The GNN layer is shared across the time frames and produces new representations $\widetilde{E}_\tau^i$ which are learned by utilizing either the semantic or social graphs.

Graph Convolutional Neural Network. To embed the nodes in our graph, we employ Graph Convolutional Networks (GCN) [84]. GCN is a commonly used, powerful graph embedding method that encodes both local graph structure and features of the nodes, by using a layer-wise propagation rule.

Graph Attention Network. As users have a different influence on one another, we need to focus on users that have more relevant connections with higher influence. To model the importance of the influences of the neighborhood to a node, we use Graph Attention Networks (GAT) [47]. GAT attends to the neighborhood of each user and assigns an importance score to the connections that contribute more to the detection of misinformation spreaders.

Hyperbolic Graph Convolutional Neural Networks. Research has shown that GCNs often do not generalize well to hierarchical, tree-like networks such as the social graphs constructed from social media threads [239], since they operate in the Euclidean space. Building on the scale-free nature of the users' social graphs, we utilize Hyperbolic Graph Convolutional Neural Networks (HGCN) [240] which employ graph convolutions in the hyperbolic space as opposed to the standard graph convolutions. The HGCN layer projects the user embeddings in the hyperbolic space to minimize distortions and learn better representations.

Temporal Neural Network Layer

Temporal Encoding. We investigate the users' behavior over a long time-period, and we wish to encode the dynamic changes between the users' interactions over time. We argue that simply compressing the users' semantic and social connections into one static graph, would introduce too much noise and the information regarding the temporal fluctuations of the semantic and social relationships would be lost. To this end, we model the sequential dependencies through time for each user, with a Gated Recurrent Unit (GRU) [51]. The GRU encodes the dynamic user graph representations across the time axis, producing hidden states for each time frame τ.

Temporal Attention and Network Optimization. The GRU models the sequential dependencies of the temporal graph user representation, however, during the long time span of the users' posting activity, certain socio-political events, such as the election seasons, the release date of a new vaccine, etc., may cause the outburst of misinformation spreading. Therefore, we wish to model the contributions of these important time periods to the users' overall representation. To this end, we employ an attention mechanism [241] to compute an overall representation for the user with adaptive weights over the aggregated GRU hidden states.

We formulate the author profiling problem as a binary classification task to predict the class y^i of the user, where $y^i \in \{\text{misinformation spreader, real news spreader}\}$. The overall learned representations for each user are forwarded into a linear layer, and we use cross-entropy loss to calculate the difference between the true and predicted labels.

6.5.4 Experimental Setup

We investigate the reliability of the temporal semantic and social connections as features for identifying misinformation spreaders in various scenarios.

Predicting future user behavior. We analyze whether past user behavior, represented through temporal graphs, can be used to predict their future user behavior. To this end, we use the whole set of users in the training, validation, and test, but each set contains data from different time periods. Specifically, the training set consists of 8 months (Jan-Aug 20'), and the validation (Sep-Dec 20') and test sets (Jan-Apr 21') 4 months each, resulting in a consecutive 50:25:25 *time split* of the user's posting history. This stands for both datasets since they were collected around the same time period. We provide a visual depiction of this split in Figure 6.16(a).

Generalizing to unseen users. We examine which types of relationships have the ability to generalize to unseen users. In this setup we utilize a *user split*, where we divide the users into a train:validation:test sets of ratio 70:10:20 using all of their posting history. This split is also visually depicted in Figure 6.16(b).

Performance on unseen users in the future. We also aim to test whether the temporal graph features generalize on both unseen users and future content, to this end we utilize the *mixed split*. We split the users into a train:validation:test sets of ratio 70:10:20, where the train set contains users who have posted the first half (Jan-Aug 20') of the whole time period, while the validation and test sets contain a different set of users who post on the second half (Sep 20'-Apr 21'). With this setup, we evidently demonstrate the reliability of the proposed model of detecting misinformation spreaders on unseen data. A visual depiction of this split is provided in Figure 6.16(c).

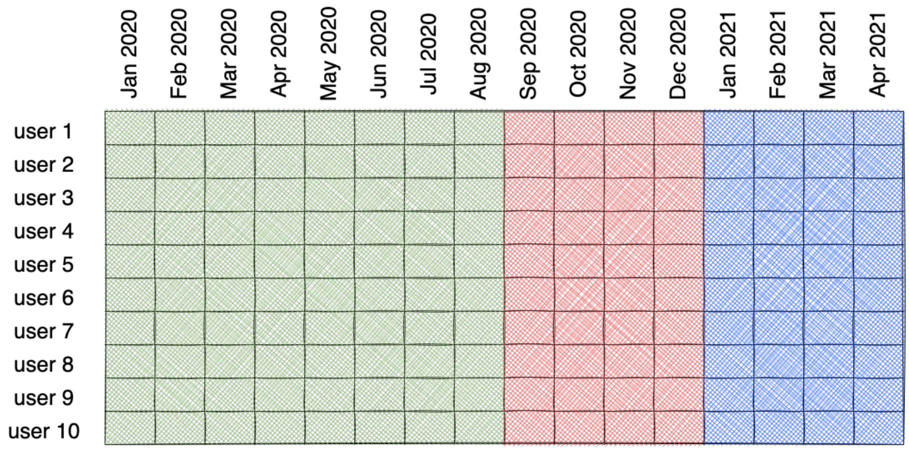
Training Setup

We use the pre-trained model `'all-mpnet-base-v2'` from SBERT[9], which achieved the best performance on various challenging similarity datasets [230]. This model has a max length set to 512, uses mean pooling, and has the output dimension $d_b = 768$. For each post in the user history, we masked the links so that the cosine similarity is not attributed based on the links. We run experiments with $\delta \in 15, 30, 60, 360$ (δ is the number of days spanned by each that each time period τ). In each sample, we randomly sample $n \in 200, 400, 800, 1200$ users, and we build a subgraph of those users for each discrete time window. In the semantic graph, we connect users with each other based on the hyperparameter $\theta \in [0, 1]$ (as defined in Section 6.5.1). We find out that our model works best with the following hyperparameters: $n = 200$, $\delta = 30$, $\theta = 0.8$. For the models initialized with User2Vec embeddings, we use the dimensions $d_g = 100$ for our graph layer and $d_r = 50$ for our GRU sequential layer. On the other hand, for the models initialized with UBERT embeddings we use the dimensions $d_g = 256$ for our graph layer and $d_r = 128$ for our GRU sequential layer. We use Adam optimizer [65] with learning rate $5e-5$, weight decay $1e-2$, and train the model for 100 epochs using early stopping with patience 20 on the validation set. We run each experiment with 5 random seeds and report the mean result on the test set in Tables 6.4, 6.9, and 6.10. DyGAT model using User2Vec embeddings as initialization has 116K parameters, while DyGCN and DyHGCN have 55K parameters. On the other hand, DyGAT using UBERT embeddings as initialization has 1M parameters, while DyGCN and DyHGCN have 427K parameters. Our experiments for each model take around 1 hour to run on NVIDIA A100-PCIE 40GB GPU. Our implementation, the annotated dataset, and the results are publicly available to facilitate reproducibility and reuse.

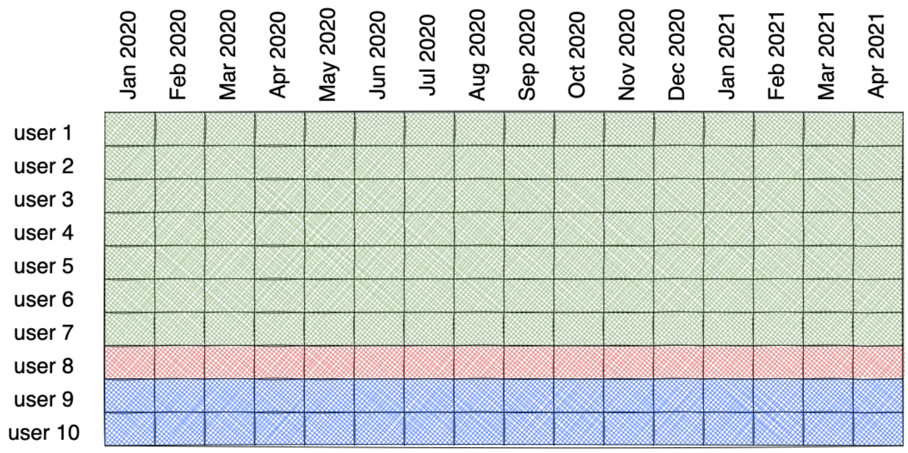
6.5.5 Results

Feature baselines First, we compare the proposed model to simple, yet strong content-based baselines by utilizing interpretable classifiers; Support Vector Machines (SVM), Logistic Regression (LR), and Random Forest (RF) using the following features:
ngrams: While word ngrams are considered as simple features, they have been used successfully in the past for identifying misinformation spreaders [242]. In this case, we utilized the word bi-grams.
statistical-emotional (StEm): We employ a feature vector ($n = 22$) with standard statistical linguistic variables (such as min, max, average number of tokens and characters, lexical diversity, etc.) [243, 198]. Additionally, we added 8 emotional dimensions to this baseline feature [244, 245].

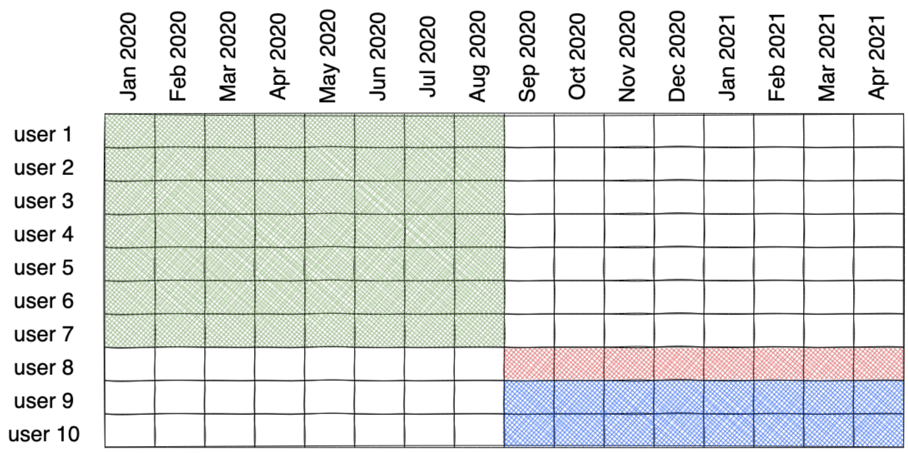
[9] https://www.sbert.net/docs/pretrained_models.html



(a) Time split. Splitting the time periods in order to predict future user behavior.



(b) User split. Splitting the users in order to predict the behavior of unseen users.



(c) Mixed split. Splitting the users and the time periods in order to predict the behavior of unseen users in the future.

Figure 6.16: Visual demonstration of the (a) Time split, (b) User split and (c) Mixed split.

	FACTOID								
	Time Split			User Split			Mixed Split		
	SVM	**LR**	**RF**	**SVM**	**LR**	**RF**	**SVM**	**LR**	**RF**
ngrams	43.6	56.4	55.4	43.4	58.4	59.5	42.5	42.5	57.6
StEm	52.5	51.6	56.8	49.1	54.9	60.6	54.1	52.1	60.3
UBERT	42.5	47.9	56.1	53.9	58.6	49.7	42.3	45.7	54
UBERT	42.5	47.9	56.1	53.9	58.6	49.7	42.3	45.7	54
U2V	47.6	52.1	61.3	50.2	55.1	56.5	46.4	53.0	59.6
DyGAT	**64.56***			63.59			63.22		
DyGCN	64.18			65.75			**64.23***		
DyHGCN	64.24			**66.75***			58.58		

Table 6.7: Baseline experimental results on the FACTOID dataset. We use Support Vector Machines (SVM), Logistic Regression (LR), and Random Forest (RF). Bold indicates the best macro F_1-score. All results are in percentages. We show that the DyGNN framework outperforms all baselines for each split in both datasets. The results with the asterisk (*) are statistically significant based on the Wilcoxon signed rank test ($p = 0.001$) compared to all the baseline methods.

	Twitter								
	Time Split			User Split			Mixed Split		
	S	**LR**	**RF**	**S**	**LR**	**RF**	**S**	**LR**	**RF**
ngrams	73.9	75.2	76.9	61.7	65.5	66.6	52.37	42.6	64.81
StEm	61.4	60.8	70.2	59.4	57.3	63.9	43.0	43.5	63.6
UBERT	62.6	77.3	71.9	64.1	64.7	64.3	36.2	59.4	65.8
U2V	-	-	-	-	-	-	-	-	-
DyGAT	**78.2***			67.30			**69.2***		
DyGCN	66.9			65.60			66.1		
DyHGCN	67.7			**73.90***			65.3		

Table 6.8: Baseline experimental results on the Twitter dataset.

UBERT: We use the SBERT embeddings of the documents averaged over the whole time frame as feature vectors.
U2V: We also utilized the User2Vec embeddings to represent the users as feature vectors.

Table 6.7 and 6.8 show the accuracy results of the baseline models compared to the dynamic graph models on the FACTOID and Twitter datasets. Note that we utilized both the social and the semantic graph and two initialization methods for the FACTOID dataset - in this table, we report the best-performing variant (for all variants see Table 6.9). For the Twitter dataset, we experimented only with the semantic graph since there are no social connections between users, and we obtained the temporal graphs with UBERT. We observe that all the proposed models significantly outperform all baseline models for both datasets. For the FACTOID dataset, the best performing dynamic graph model showed a higher macro F_1-score compared to the baseline models in all splits, which was on average 10.47% higher on the time split, 15.3% on the user split and 14.08% on the mixed split. For the Twitter dataset, the best performing dynamic graph model showed on average 8% better performance on the time split, 10.8% on the user split, and 16.8% on the mixed split.

The results on both datasets validate our claim that the specific language features become quickly outdated, while temporal semantic similarities and social interactions are more robust and constitute a better tool for (a) predicting future behavior (time split), (b) predicting the behavior of unseen users (user split), and (c) identifying misinformation spreaders on unseen data (mixed split).

		Semantic			Social		
		Time	User	Mixed	Time	User	Mixed
UBERT	DyGAT	**64.56***	57.26	60.46	62.91	61.66	63.12
UBERT	DyGCN	63.57	58.67	61.60	64.18	61.08	59.44
UBERT	DyHGCN	55.39	**66.75**	55.25	56.38	62.02	58.58
U2V	DyGAT	63.03	63.59	62.88	63.50	63.01	**63.22***
U2V	DyGCN	62.28	65.75	**64.23***	62.76	64.21	61.35
U2V	DyHGCN	42.51	42.52	47.39	**64.24***	**66.09***	56.10

Table 6.9: Comparative analysis of two embedding methods for semantic graph construction and DyGNN initialization (social graph). Reported macro F_1-score for the FACTOID dataset. All results are in percentages. Bold indicates best result. The results with the asterisk (*) are statistically significant based on the Wilcoxon signed rank test ($p = 0.001$) compared to the second best performing method.

6.5.6 Qualitative Analysis

Table 6.9 shows the performance results on the three different experimental setups. We analyze the results of the dynamic graph models, based on the utilized graph type (semantic and social), initialization method (UBERT and User2Vec), and graph neural network type (GAT, GCN and HGCN).
Comparing graph types. We observe that the model obtains a slightly better performance by utilizing the semantic similarity graphs compared to utilizing the social graphs for all three setups. Figure 6.14 shows that the percentage of temporal connections is higher, and fluctuates more, on the semantic graphs compared to the social graphs. This may represent users sharing similar opinionated news regarding the same event, with patterns changing for a new event, while social connections stay similar.

In the time split, for the DyGAT+UBERT model, we observed that the results are not significantly different when comparing the utilization of semantic and social graphs. In the same split, for the DyHGCN+User2Vec model, we note that 24.99% of the users were classified differently by the semantic and social models, this difference is expected since the difference between the F_1-scores obtained by each graph type is more than 20%. When the semantic graph is utilized, we observe that DyHGCN+User2Vec fails to recognize any of the misinformation speaders, however, it achieves impressively high performance with the social graph. This result is justified due to the low hyperbolicity values of the semantic User2Vec graph.

In the user split, for the DyHGCN+UBERT model, we note that 32.54% of the users were classified differently from the semantic and social models, even though the difference between their macro F_1-scores is only 4%. In the same split, for the DyHGCN+User2Vec model, we note that 39.72% of the users were classified differently, however, this difference is expected since the F_1-scores obtained by the semantic and social models have more than 20% difference between them. Once more we observe a staggering difference between the F_1-scores obtained from semantic and social models, with the social model achieving the highest score.

In the mixed split, for the DyGCN+User2Vec model, we note that 27.55% of the users were classified differently. We observe that the model obtains higher recall on the fake news spreader class when the semantic relationships are utilized, instead of the social ones. In the same split, for the DyHGCN+UBERT model, we observe that 7.82% of the users were calculated differently.

Comparing initialization methods. When UBERT and User2Vec are used in the social graphs, they simply act as initialization vectors, since the social graph construction does not depend on the embedding method. When the models use the social graphs, User2Vec initialization produces better results than UBERT in all setups, despite its lower dimensionality. This performance is expected since User2Vec yields better results than UBERT when it is utilized as a baseline method (Table 6.7).

The semantic similarity graphs, on the other hand, differ when constructed with UBERT or with User2Vec. In the time split evaluation setup, the semantic graph model achieves the best performance with UBERT, while in the mixed split, the best performance is obtained with User2Vec. This is likely due to UBERT particular suitability for capturing meaningful user similarities even with a small amount of user history since SBERT (from which we obtain UBERT) is tailored for producing sentence embeddings comparable using cosine-similarity. User2Vec requires a significant amount of documents in order to obtain high-quality user representations however, it leads to a stronger generalizability on unseen data.

Comparing dynamic graph neural networks. We observe that the hyperbolic DyHGCN obtains the best performing results in 3/6 combinations of split and graph type. However, it performs poorly when it utilizes the User2Vec semantic graphs. Figure 6.17 shows the average hyperbolicity of the dynamic graphs for each month. As is known, high hyperbolicity values indicate a tree-like structure of the network [246, 247]. Due to the lower posting activity during the last months, and thus higher sparsity of the topics represented by one user, users are more dissimilar, resulting in fewer edges. This in turn leads to lower hyperbolicity during this time period, which explains the DyHGCN's poor performance with User2Vec semantic graphs. The social graph shows high hyperbolicity for all months, therefore DyHCGN achieves superior performance when utilizing the social graphs. DyGAT and DyGCN obtain the best performance once, but in contrast to DyHGCN, they both achieve results within a certain range which is neither too low nor too high.

Discussion. In conclusion, based on this comparative analysis, dynamic semantic similarity graphs

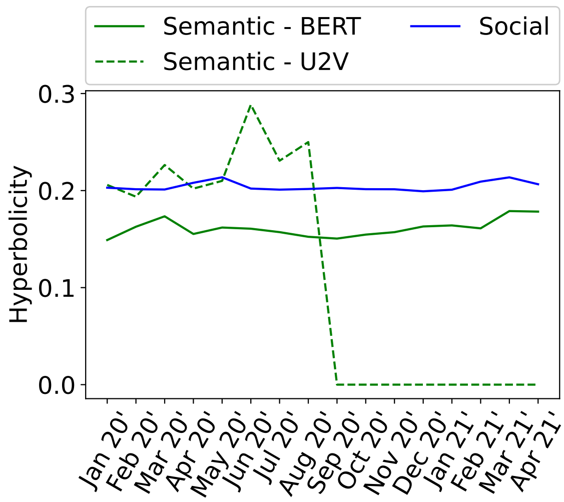


Figure 6.17: Average hyperbolicity per month.

lead to better results than dynamic social graphs, and given a large amount of user history, User2Vec is preferred for constructing these. In addition, the use of DyHGCN is recommended only when the hyperbolicity of the graph is high, alternatively, DyGAT or DyGCN provide comparable results.

6.5.7 Ablation Study - Temporal Components

	Semantic			**Social**		
	Time	**User**	**Mixed**	**Time**	**User**	**Mixed**
DyGNN	**64.56***	66.75	**64.23***	**64.24***	**66.09***	**63.22***
no temporal	55.14	53.53	60.24	62.64	59.37	56.54
no attention	62.27	**66.78***	61.97	61.01	64.51	56.32

Table 6.10: Ablation study - temporal dynamics. In this study, we remove the temporal component (keeping simple "static" GNN approach) and the attention. Results show that both components play a significant role in the model's performance. Bold indicates the best macro F_1-score. All results are in percentages. The results with the asterisk (*) are statistically significant based on the Wilcoxon signed rank test ($p = 0.001$).

We perform an ablation study on the components of the best performing dynamic graph model to demonstrate the effect of each layer on the overall performance, namely the temporal attention and the temporal graphs:

No attention. We remove the temporal attention layer from our dynamic graph model. Intuitively, this component should focus on the time periods with high misinformation spreading activity and the highest differences between user groups.

No temporal dynamics. We average each user's representations across all time frames to obtain a single user representation and remove the dynamic part of our model by merging all the graphs constructed for every discrete time frame. Specifically, we construct a single graph that includes all the user connections from all time periods and replace the GRU layer, with a linear layer. This model captures the overall semantic and social interactions of the users over their whole posting timeline, and could also be considered as a graph-based baseline.

Table 6.10 shows the ablative results over the components of the best performing dynamic graph models for all setups. We observe that removing the temporal information has a significant detrimental effect on the performance in all cases, which is on average 7.53%. This demonstrates the strong predictive power of temporal patterns in semantic and social relationships for identifying misinformation spreaders and validates our proposed framework for dynamically modeling the users' semantic and social graphs. In addition, except for the semantic graph on the user split, adding temporal attention to the users' timeline increases significantly the performance, reinforcing our hypothesis that the similarity of language use during important socio-political events is strongly indicative of misinformation spreading. We have seen that for the semantic graph using the user split, the attention weights through different time slots are the same. Due to this reason, the overall user representation is just a simple average of the GRU states. One reason why this is happening is because the temporal attention is not capturing temporal patterns of the users, that can generalize to unseen ones.

6.5.8 Error Analysis

We conducted an analysis of users that consistently get the same prediction *by at least half* of the GNN models. We identify two groups of users; consistently correctly classified, and consistently misclassified. The following error analysis is based on the results obtained on the FACTOID dataset on the user split, however, similar results were observed for the rest of the splits.

Approximately 72% of the consistently misclassified users are misinformation spreaders, which can be attributed to the class imbalance decreasing the recall.

It is harder to identify users who are borderline fake news spreaders. Table 6.11 shows, for the correctly classified and misclassified fake news (FNS) and real news spreaders (RNS), the average number of fake and real news posts, average science and factual level provided in Section 6.3 and the average no. of months of active posting. The science level of each user $\in [-1, 1]$ is the normalized weighted average of non-scientific (-1) and scientific (1) articles and the factual level $\in [-3, 3]$ is the normalized weighted average factuality of the news domains, manually labeled by journalists from very low (-3) to very high (3). [10]

		fake posts	real posts	science level	factual level	activity (months)
correctly	**FNS**	9.66	39.45	0.13	0.59	12.99
classified	**RNS**	0.29	9.95	0.70	1.76	12.57
mis-	**FNS**	3.76	22.88	0.16	0.83	11.21
classified	**RNS**	0.60	22.67	0.42	1.59	12.37

Table 6.11: Error analysis for the performance of the models. Correctly classified fake news spreaders (FNS) post more often than misclassified ones, and post more consistently over time.

As we can see, the misclassified FNS have posted a considerably lower number of fake news on average compared to the correctly classified FNS. While they also posted a lower number of real news posts, their (annotated) factual level is quite high - the source quality plays a role. For the correctly classified FNS, a high number of real news combined with a low factual level indicates that the real news sources these users are posting are borderline credible - their credibility level is only ‘mostly factual’(+1), whereas the credibility level of the fake news sources is from ‘low’(-2) to ‘very low’(-3). The correctly classified RNS tend to post significantly more scientific articles and articles with higher factuality on average than the misclassified RNS. Overall, correctly classified users of both classes post more consistently over the months compared to the misclassified users.

Since our data heuristics might include wrongly labeled posts and, by extension, users, we manually labeled 210 posts of consistently misclassified users. In this small sample, we found that approximately 14% of the posts were wrongly labeled, however less than 1% of the users would obtain a different label because of these posts. We show two examples of mislabeled posts in Table 6.12.

[10] When embedding the users, we erased the URLs from the text, so that no information about the number of links, or the names of the domains was leaked in the user embeddings, therefore none of the models could have had any prior knowledge of these factors.

Mislabeled as fake news
(...) These pieces rely on discredited sources who have peddled debunked theories about Dominion's supposed ties to Venezuela (...) These statements are completely false and have no basis in fact. (...) [link to non-credible source posting fake news]
Mislabeled as real news
The CCP (Chinese Communist Party) controls Google from within. Change my mind. [link to credible source posting real news]

Table 6.12: Mislabeled news posts.

6.6 Summary

In this chapter, we addressed the third research question, focusing on capturing temporal user behavior. To this end, we introduce in Section 6.3, FACTOID, a new user-centered dataset for misinformation spreader analysis, monitoring political discussions on Reddit since the beginning of 2020. Our dataset contains over 4K users with 3.4M Reddit posts, covering the time period before and after the US presidential elections. Apart from the fake news/real news distinction, the dataset contains fine-grained labels about the users' credibility level and political bias. As far as we are aware, this is the first fake news spreader dataset that simultaneously captures both the long-term context of user's historical posts and the interactions between users. To create the first benchmark on our data, we provide methods for identifying misinformation spreaders by utilizing the social connections between the users along with their psycho-linguistic features. In a subsequent analysis, we observe that social connections increase robustness over content features, that detecting affective mental processes correlates negatively with right-biased users, and that the openness to experience factor is lower for those who spread fake news.

Moreover, we analyze the patterns of users spreading misinformation through time. In order to capture temporal information, we proposed a dynamic graph neural network framework that generates temporal graph representations from the users' semantic similarities and social interactions through time. We find that static features become quickly outdated, while modeling temporal semantic similarities and social interactions constitutes a better tool for capturing the predictive patterns of the user's dynamic context. For example, we observed, different peaks and fluctuations in the percentages of user-to-user relationships, influenced by a variety of events and their behaviors. These dynamics of users' interactions, based on their credibility label, were the first indication that the temporal user dynamics might be more useful than the static features. Our extensive experiments and ablation study demonstrated that the temporal graphs are more efficient than content-based models or simple static graphs for predicting (a) the future misinformation spreading behavior, (b) the behavior of unseen users, and (c) misinformation spreading behavior in a zero-shot scenario. These results indicate that a model utilizing temporal user relationships is more robust and more efficient for misinformation spreader detection compared to topic-sensitive or time-agnostic models, e.g. talking about Trump doesn't make one a misinformation spreader, and it is quite normal near election time.

Through exploratory experiments, we analyzed the various aspects of the framework in order to provide insight into its usability. These experiments showed that dynamic semantic similarities lead to better results than social ones. The ablation study on the components of the model revealed that the temporal modeling of the users' semantic similarities and social interactions significantly contributes to identifying misinformation spreaders effectively. Our error analysis indicated that the misclassified fake news spreaders tend to post a very low number of fake news posts and a high number of real news posts from highly credible sources. Yet, the proposed framework is applicable as a human moderator-assistance tool for identifying users who post fake news more consistently.

In the next chapter, we will transition from text classification tasks toward the domain of personalized language generation to explore methodologies that incorporate effective user context in language models.

CHAPTER 7

Personalized Natural Language Generation

Building upon our exploration of integrating user context into text classification tasks, this chapter transitions our focus towards the domain of language generation, specifically targeting personalization aspects. Our prior work has demonstrated the value of leveraging users' large-scale posting histories to construct implicit representations that enhance NLP systems across various classification tasks. Nevertheless, language models generally treat language generation independent of individuals, without tailoring their language according to their traits.

In this chapter, we delve into personalization language generation with different types and amounts of context for users. We address the following research question:

> Research Question 4 (*RQ4*)
>
> How effectively can generative models adopt personalized perspectives using user context, and what strategies can be employed to incorporate this context into the models?

In the evolving landscape of language technologies, there is an increasing demand for personalized language generation systems that can mirror a user's individual style. These systems can generate responses that are not only relevant to the user's queries but also reflective of their personal style, thereby creating a more engaging and customized interaction experience.

To address the complexities of this task, we construct a realistic dataset with self-reported sentences from users. Additionally, we analyze the difference of several architecture modifications, that can be utilized for controlled personalized language generation. We hypothesize that the utility of different types of historical data may vary depending on the specific architecture employed. Finally, recognizing the limitations of traditional evaluation metrics in capturing the nuances of personalized language generation, we propose a new novel human evaluation setting. This evaluation setting is designed to better capture the differences between models in terms of their ability to generate personalized responses that reflect the user's style.

The key contributions of this chapter are:

- We construct a corpus, containing 95K judgments of social situations written by 6K authors, together with their self-disclosure statements.

- We design two transformer architectures to embed personal context and find that our twin encoder approach outperforms LLMs.
- We develop a novel evaluation by asking humans to rank the human response, model output, and a distractor human response, combining approaches from persona consistency.

This chapter is based on the following publication ([19]):

- **Joan Plepi**, Charles Welch, and Lucie Flek. 2024. Perspective Taking through Generating Responses to Conflict Situations. In Findings of the Association for Computational Linguistics ACL 2024, pages 6482–6497, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

The rest of the chapter is structured as follows. Section 7.1 introduces the work, and we describe the related work in Section 7.2. We describe the dataset in Section 7.3 and the problem formulation in Section 7.4. In Section 7.5 we describe the architectures used for this task, following by experimental setups and results in Section 7.6 and Section 7.7. Finally, in Section 7.8 we provide a summary for the chapter.

7.1 Introduction

Despite the steadily increasing performance language models achieve on a wide variety of tasks, they continue to struggle with theory of mind, or the ability to understand the mental state of others [248, 249]. A range of theory of mind tasks have been studied in developmental psychology, including diverse desire, diverse belief, knowledge-access, false belief, and hidden emotion [250]. Results consistently show that the hidden emotion, or understanding the difference between felt and displayed emotions, is the most difficult of these tasks [251]. Language assists in the development of the theory of mind, as it facilitates the exploration of mental states [252].

This ability is central to much of human interaction and could provide many benefits for language models as well, as being able to foresee the reactions of others allows us to better decide which action to take next. This could help language models generate responses that are safer [253], in particular for healthcare applications [254], or more personalized, e.g. to sound more empathetic [255], or provide targeted explanations [256]. In fact, there is a growing interest in a perspectivist approach to many natural language processing (NLP) tasks, which emphasizes that there is no single ground truth [161, 169, 257]. This is a more common view in generation tasks, as it is easier to see that multiple translations or continuations of a dialog are correct. However, Flek [20] emphasized the need to interpret language with its personal contextual factors to create higher performing personalized systems. Dudy et al. [258] similarly argue that additional contextual information should be incorporated in such models, particularly for natural language generation (NLG).

We construct a corpus to study perspective taking through generating responses to conflict situations. An example from our corpus can be found in Figure 7.1. We see a user asking if they did something wrong in a conversation with their girlfriend about whether or not to terminate a pregnancy. On the right, there are two responses from other users with different judgments of the situation (reasoning and verdict NTA/YTA). On the left, we see self-descriptive statements of each user. Author Y appears to be more family-oriented than Author X which may impact their judgement of the situation. Our first research sub-question is: *How should we evaluate perspective taking through the lens of NLG?*

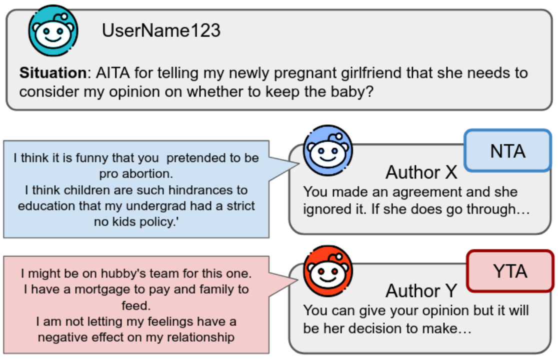


Figure 7.1: Example of a post in AITA subreddit. The example includes a situation title and two comments with different perspectives regarding the situation, plus persona sentences for the respective users.

We develop a novel evaluation by asking humans to rank the human response, model output, and a distractor human response, combining approaches from persona consistency [259] and theory of mind work [260].

Next, we ask: *Do tailored, user-contextualized architectures outperform large language models (LLMs) on this task?* We design two transformer architectures to embed personal context and find that our twin encoder approach outperforms LLMs. Lastly, we ask: *What user information is most useful to model perspective taking?* Experiments with varied user contexts showed that self-disclosure statements semantically similar to the conflict situation were most useful.

Furthermore, by training a generative model to produce perspectives, we outperform recent supervised data perspectivist models also on verdict prediction. Our corpus contains 95K judgements of conflict situations written by 6K authors filtered from Social-Chem-101 and is extended with 20-500 self-disclosure statements per author. We will release our corpus, code, and human evaluations.

7.2 Related Work

Perspective-taking is part of the process of understanding others mental states and theory of mind is fundamentally based on the understanding that individuals have different beliefs and emotions. Benchmarks for theory of mind in artificial intelligence sometimes take the form of multiple choice questions about false belief [260]. These are context specific, as a situation and knowledge of involved parties is described, but not unique to individual people. When we look at emotions, the most difficult of theory of mind tasks for children [251], different people may feel differently because of their unique perspective and experiences. As such, our work relates to work on personalized understanding and generation.

Recent models, such as RoBERTa or GPT-3, struggle with solving simple perspective-taking toy tasks [261, 262, 263], including the false belief test [264]. Currently, the best LLMs (such as GPT-4 or Llama2-70B) are to a limited extent capable of perspective-taking-based reasoning and chain-of-thought explanations in social commonsense scenarios [265, 266, 267]. However, these attempts has been heavily criticized by psychologists as over amplified by the AI community while very basic in design, likely exhibiting shortcut learning [261, 268]. Children's theory of mind ability appears associated with "exposure to rich discourse about thoughts, feelings, and intentions" [269].

Personalized Generation The first step towards successful modeling of perspective taking is being able to mimic the communication of a given person accurately. Researchers largely agree on the need to include personal, social, cultural, and situational factors into language interpretation and generation, in order to avoid harmful consequences and to steer human-centered AI alignment [21, 270, 258]. However the approaches to what and how to model vary dramatically, from an authorial style to a chat persona's factual trivia.

Architectural predecessors of personalized LLMs are persona-based dialog systems, attempting to generate a response given an input utterance and additional personal trivia information. Li et al. [271] introduce a speaker model and a speaker-addressee model. Madotto et al. [259] use only a few dialog samples to generate personalized responses, by casting personalized dialog learning as a meta-learning problem. Other works have modified sequence-to-sequence frameworks to infuse persona information in the decoder [272, 273]. Song et al. [274] introduce Persona-CVAE and Ma et al. [275] combine a history encoder, personalized post encoder, user history memory, and personalized decoder to fuse the user profile into the response.

Many personalized generation models operate on the PersonaChat dataset [276], where two crowdworkers converse with each other, attempting to represent a persona described by five short sentences, resulting in artificial conversations. Dialogs directly using words from persona sentences were later adjusted [277], however, the data still includes unrealistic personas ("to stay fit, I chase cheetah in the zoo") and unrealistic responses that enforce the usage of the trivia ("I am reading a book."-"Ok. I am a dentist."). Other data include personalized recipes [278] or movie dialogs [279].

7.3 Dataset of Social Situations

We used the dataset of Welch et al. [15] as the foundation of our work, as it addresses the emotional and social aspects of perspective taking, while requiring knowledge of personal beliefs. The authors collected data from Reddit, an online platform with many separate, focused communities called subreddits. The data is from the AITA subreddit, where users share descriptions of social situations that they are involved in and ask members of the community for their opinions. These members assess if the poster is the wrongdoer in the described situation. They provide a verdict in the form of "you're the asshole" (YTA) or "not the asshole" (NTA). The dataset was filtered from Forbes et al. [144]'s Social-Chem-101 corpus but also includes the post title, full text, all comments, and their corresponding authors. We refer to the post title as the *situation*, as the title is usually a short description of the conflict situation. The comments are preprocessed in order to extract those that contain a verdict of YTA or NTA,[1] and others were removed. In order to extract verdicts, they manually created a set of keywords for both classes and filtered the comments to remove these expressions. The initial dataset contains 21K posts, and 364K verdicts (254K NTA, 110K YTA) written by 104K different authors.

7.3.1 Extracting Self-Disclosures Statements

We expand the dataset by retrieving the comment histories for each user in the dataset. To extract the self-disclosure sentences for the users, we adapt the approach described in Mazare et al. [280].

[1] Reddit posts were crawled with the Reddit API (https://www.reddit.com/dev/api) and comments with the PushShift API (https://files.pushshift.io/reddit/comments/).

Initially, we split each comment into a sentence and kept only sentences that contain between 5 and 20 tokens. Then we add two constraints to each sentence in order to classify it as a self-disclosure sentence; (1) it must contain the tokens *I*, *my* or *mine* and (2) one verb, one noun, and one pronoun or adjective.

After performing these steps, we obtained a set of self-disclosure statements for each user. We filtered our dataset to include only users who have more than 20 and fewer than 500 statements. Our final dataset contains 20K posts and 95K verdicts written by 6K different authors, which we will release upon publication.

7.3.2 PersonaChat Discussion

Work on personalized or persona-based dialog systems has begun to incorporate contextual information in response generation. The work of Zhang et al. [276] introduced the PersonaChat dataset, where two crowd workers converse with each other while attempting to emulate a persona described by five short sentences. Models developed using this data condition on encoded persona sentences. Dinan et al. [277] extended this dataset with rephrasings of the utterances to avoid high direct word overlap with persona sentences, yet these dialogs focus directly on incorporating information from a few short phrases. Workers were instructed to use these facts in their conversations, which leads to artifacts, such as the unprompted addition of personal information to the end of unrelated utterances (e.g. "I am a lifeguard" in response to someone saying they will read a book). They do not accurately reflect the real world, e.g. "to stay in shape, I chase cheetahs at the zoo", and they ask people to emulate an identity whose life experiences (e.g. getting divorced, living in different places, being a lawyer, owning a business) could plausibly shape their views of interpersonal conflict described in our data, but through the shallow nature of crowdsourced conversations and lack of real lived experience of participants, fails to be reflected in the PersonaChat dialogs. Our dataset is instead constructed from the profiles of real people who wrote both the judgements of social situations and their persona sentences.

In an effort to quantify the differences between PersonaChat and our corpus, we measured the unigram and bigram Jaccard similarity between persona sentences and author responses. We calculated the maximum similarity between any persona sentence for an individual and their given response. This follows the idea that PersonaChat directly incorporates facts from the persona, leading to high similarity between a persona sentence and a given dialog response (for example, where an utterance says "I am about to watch Game of Thrones" and a persona sentence says "I love watching game of thrones"). We report this value averaged across all users for each corpus. We found the unigram similarities to be 0.16 and 0.12 for PersonaChat and our corpus, respectively. Our corpus had a max bigram similarity of 0.01, whereas PersonaChat's was four times higher at 0.04. This shows that even after efforts were made to reduce direct overlap in the PersonaChat corpus, the similarity between the persona sentences and responses is high.

7.4 Problem Formulation

Our task considers as a data point, a post that contains a summary of the situation description, a comment of the post containing a personal verdict about the situation, and the author of the verdict jointly with the corresponding self-disclosures. Therefore, for our generation task, we have three components: (i) the input sequence which corresponds to the main post, (ii) the target output sequence which corresponds to the comment containing the verdict, and (iii) auxiliary information. The auxiliary information can be self-disclosure statements (S) or comments (C), and allows us to address the third sub-question: What information is most useful for perspective taking?

In order to tackle this research question, we formalize the following task. For a given situation post s written from a random author, we have a set of comments $C_s = \{c^s_{a_1}, c^s_{a_2}, \ldots, c^s_{a_n}\}$ written by n different authors. Each post describing a situation s contains many comments $c^s_{a_i} \in C_s$, and an author a has many comments $c^{s_i}_a$ on different posts s_i. Hence, as we have different target outputs, for the same input sequence, we need additional information to condition our model. The generation task can be formalized as $p(c^s_a|s, a)$. For each author a the model can take advantage of $P_a = \{p^a_1, p^a_2, \ldots, p^a_k\}$, where p^a_i, denotes the $i-$th personal context for author a. We describe two different methods to extract a set of k disclosure statements for each user in the dataset.

Random sampling In this setup, we randomly sample up to k statements for each user.

Most relevant sampling We compute embeddings using SBERT [44], for all extracted self-disclosures and situation titles in our dataset. We compute the cosine similarity between an author's statements and the situations that they have commented on and select the top k most similar statements for each situation. We aggregate the top k across situations for each author and rank the statements by their frequency, again keeping the top k.

7.5 Methodology

After formally describing the base transformer (7.5.1), we outline the modifications implemented to the encoder-decoder model in order to incorporate auxiliary information. We compare our twin encoder (7.5.2) and style decoder (7.5.3) models to large language models to answer the second sub-question: Do tailored architectures outperform LLMs?

7.5.1 Base Transformer

The main architecture used in our models is an encoder-decoder transformer model [49]. The architecture aims to model $p(y|x)$. The encoder takes as an input a sequence $\mathbf{x} = \{x_1, \ldots, x_n\}$ and maps it into a sequence of representations $\mathbf{h} = \{h_1, \ldots, h_n\}$. Given $\mathbf{h}$, the decoder generates an output sequence $\mathbf{y} = \{y_1, \ldots, y_m\}$.

Given the input sequence $s = [w_1, \ldots, w_{n_s}]$, we utilize a pre-trained transformer encoder to embed the tokens of the sequence $h = encoder(s; \theta^{(enc)})$, where $h \in \mathcal{R}^{d \times n_s}$ where d is the output dimension of the encoder and n_s is the size of the input sequence. In general, in the transformer, the output probabilities can be computed as:

$$
\begin{aligned}
o &= decoder(h; \theta^{(dec)}) \\
\hat{y} &= softmax(\mathbf{W_o}^\top o)
\end{aligned}
\tag{7.1}
$$

where $\mathbf{W_o} \in \mathcal{R}^{d \times v}$ is the language model head where v is equal to the vocabulary size, and $o \in \mathcal{R}^{d \times n_t}$, are the last decoder state for the output sequence, where n_t is the size of the target sequence.

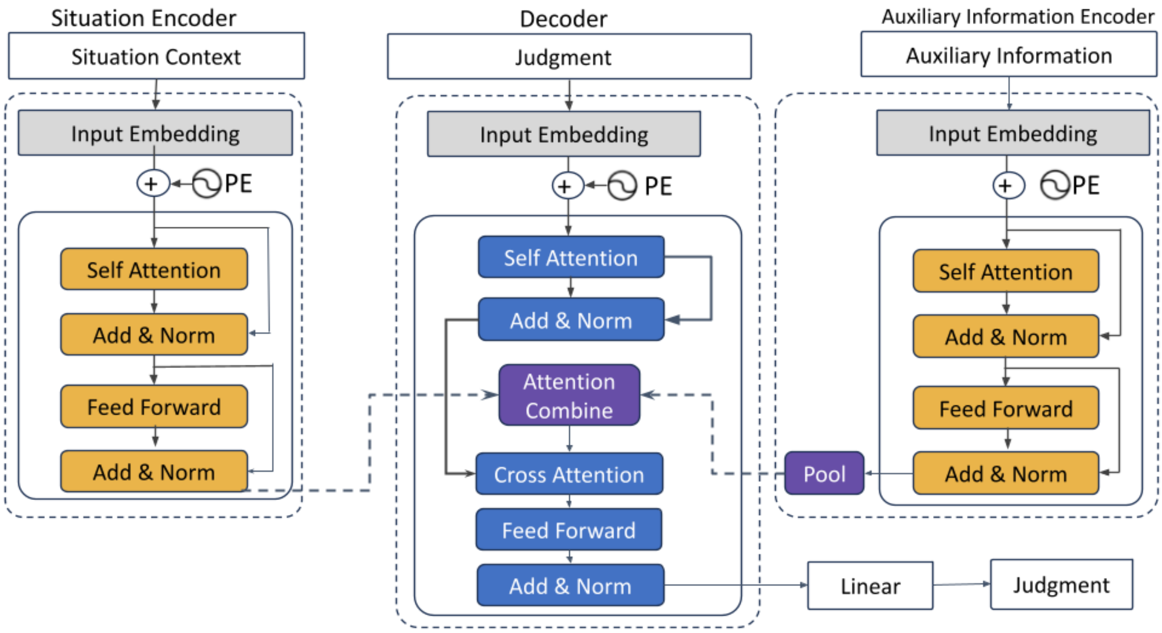


Figure 7.2: In this figure we show Twin Encoder architecture, with an extra encoder to model the auxiliary user information.

7.5.2 Twin Encoder

In Figure 7.2, we show the architecture of our first model, the twin encoder. As we described in §7.4, we are attempting to model $p(c_a^s|s, a)$, where s is the input sequence, c_a^s is the target output and a is the additional information. The sequence of personal context for user a is given by $a = [p_1^a, \dots, p_{m_a}^a]$, where $a \in \mathcal{R}^{m_a \times n_p}$. m_a is the number of auxiliary sentences, and n_p is the maximum token length in the sentences. We utilize a pre-trained transformer encoder to compute a final representation as $z = pool(encoder(a; \theta^{(enc)}))$, where $z \in \mathcal{R}^{d \times m_a}$, and $pool(\cdot)$, performs a mean-pooling over the tokens of each self-disclosure sentence. Furthermore, we compute a final representation of the auxiliary information as $\bar{z} = Att(h, z)$, where $\bar{z} \in \mathcal{R}^{d \times n_s}$ $Att(\cdot)$ is an attention layer as in [49] where the representation h of the input sequence is the query and z is the key and value. Finally, $\bar{z}$ is forwarded to the decoder layer, to perform cross attention with the decoder state.

Our twin encoder (TE) architecture is similar to the PAA model introduced in previous work [281]. Both models employ two encoder layers to model both the input context and the auxiliary information. However, the key distinction between these models lies in their approach to information processing within the decoder. The PAA model performs two cross-attentions over both encoders in the decoder and then combines the information afterward, while the TE architecture combines the encoder's information beforehand and subsequently performs one cross-attention in the decoder.

7.5.3 Style Decoder

In the second modification (Figure 7.3), we concatenate all auxiliary sentences to create the sequence of tokens $a = [w_1^{a,1}, \dots, w_{n_p}^{a,1}, \dots, w_1^{a,m_a}, \dots, w_{n_p}^{a,m_a}]$. We utilize a pre-trained transformer encoder

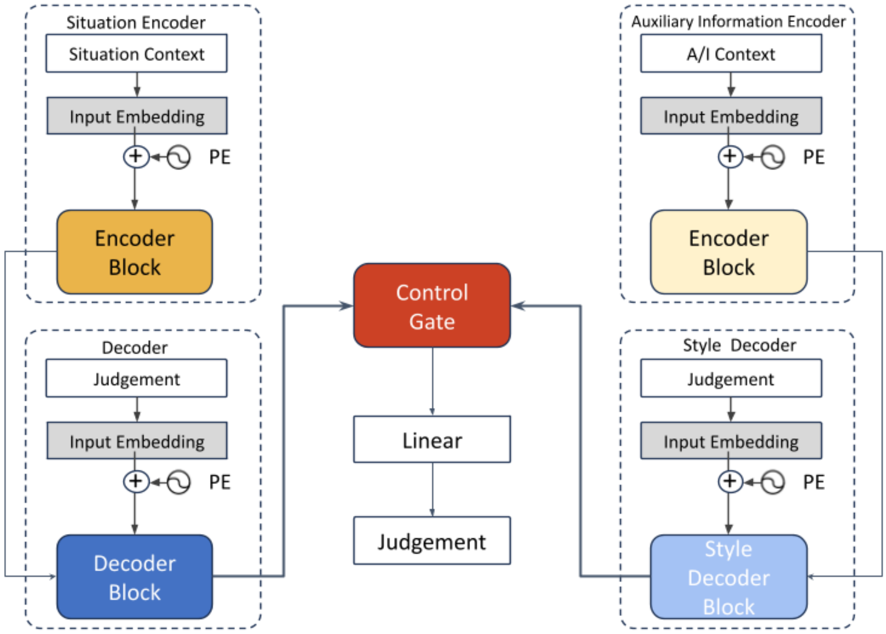


Figure 7.3: In this figure we show Style Decoder model, with a decoder that focuses on the user's style, and a control gate that controls the amount of information used from both decoders.

to compute the representations, $z = encoder(a; \theta^{(enc)})$ where $z \in \mathcal{R}^{d \times n_p m_a}$. Afterward, we compute the output distribution $\hat{y}$ as follows:

$$
\begin{aligned}
o' &= decoder(z; \theta^{(dec')}), \\
\hat{y} &= softmax(\mathbf{W_o}^\top (\alpha \cdot o + (1 - \alpha) \cdot o'))
\end{aligned}
\tag{7.2}
$$

where $o' \in \mathcal{R}^{d \times n_t}$ are the writing style decoder states, and $\alpha \in \mathcal{R}^{n_t}$. α is a learnable parameter and contains a scalar in the range of [0,1], that controls the amount of information to use out of different language heads. We compute $\alpha = \sigma(\mathbf{V}(\mathbf{W_c}[o||o'])$ where $\mathbf{W_c} \in \mathcal{R}^{d \times 2d}$, $\mathbf{V} \in \mathcal{R}^d$, and $\sigma(\cdot)$ is the sigmoid function. From the equation, the computation of α is similar to the gate computation in [282], with similar approaches used in previous works to fuse stylistic information during generation [283, 272].

7.5.4 Large Language Models

In addition to fine-tuning, we explore prompting large transformer models of the Llama2 family (containing 7 to 70 billions of parameters) by utilizing auxiliary information in the prompts. In the first setup, we adjust the prompt in order to include up to 10 examples of auxiliary information, either self-disclosure sentences or comments. Afterward, we utilize pairs of past situation titles and comments of an author to construct the prompts for the models. Moreover, we also report the results of fine-tuning the Llama2-7B model on our task. In Table 7.1, we show an example of prompts utilized for Llama2 models.

[INST] <<SYS>> I will provide self-disclosure sentences written from a user. Generate a perspective for the given situation. Perspective should be aligned with the user inferred personality. The perspective should start with the verdict YTA or NTA. YTA means You're the asshole and NTA means Not the asshole.<</SYS>> **Sampled self-disclosure sentences for user:** I don't think that talking is wrong. I just tried the game. I don't think that your wife will ever be completely faithful or honest with you. I think it was lovely for you to not only offer beef stew, but also to offer to make vegetable stew for the non-beef-eating daughter. I think you might be overreacting, because it's normal for to follow someone on multiple social media accounts. I would, in your shoes, tell my MIL that you will cut contact with her if she doesn't respect your boundaries. I want to go in a date with my husband. Yes, please help me maintain my lie so that I can date this guy. I don't blame you for not going after them for the damages. Yes, I have been married to my amazing husband for 24 years. **Situation:** AITA for telling a girl that I wasn't looking for a relationship? **Users Perspective:** [/INST]
NTA as long as you politely told her that you're just not interested in a relationship right now. And by "politely" I mean you didn't put her down in the last text.

Table 7.1: An example from our prompt with self-disclosure sentences for Llama2 models.

7.6 Experiments

In our experiments, we utilize two base models, that follow an encoder-decoder architecture. We also compare to three recent methods for incorporating personalized information:

Priming. This method was originally used in recurrent neural networks. It initially passes information about a user through the model, and then the text that needs to be classified [42]. In our approach, we sample a number of sentences from a user's history that are up to a maximum number of m tokens in order to fit into the context window of the model. Then, we concatenate this sampled text for each user at the beginning of the input text for the encoder during training.

User ID. In this approach, we append a special user token, at the end of the input text for the encoder during training. Several methods incorporate the user ID to learn user representations in the model [271, 160]. However, one drawback of this method is that it cannot generalize to unseen users during test time.

PAA Model. We also adapt the recent PAA model [281], which has shown superior performance on the PersonaChat task, to run on our dataset and compare with our proposed architectures. For the PAA model, we utilize only the self-disclosure sentences as an auxiliary input.

7.6.1 Experimental Setup

We train our models for 10 epochs, with the AdamW optimizer, using an initial learning rate of $5e-5$. We use a linear learning rate scheduler with 100 warm-up steps and early stopping on the validation set. As our base models, we are using BART [71] and FlanT5-base [72], with a maximum input length of 512, and a maximum target length of 128. BART models have up to 180M parameters, while FlanT5 models go up to 320M. For the twin encoder architectures, we found that encoding the auxiliary information separately leads to better performance, while for the style decoder, it is concatenated to create a long context. For prompting, we utilize three versions of Llama2 models [284], 7, 13, and 70 billion parameters. An example of the prompt for Llama2 models is given in Table 7.1. In the priming method, we sample $m = 100$. Our experiments run on a single NVIDIA A100 40GB GPU with an average running time (training + inference) of 6 hours. For the PAA model, we use the GPT2-medium to initialize the decoder and keep the configurations the same as described in [281]. The PAA model has 475M parameters. In order to fine-tune Llama2-7B model, we employ LoRA [285] with r=64. For this experiment, we use 4 NVIDIA A100 80GB GPU.

7.6.2 Evaluation metrics

Automatic Evaluation In the automatic evaluation for the generation task, we utilize two-word overlap-based metrics: BLEU [286] and ROUGE [287]. BLEU evaluates the quality of generated text by computing the n-grams overlap with the original comment. ROUGE is a recall-oriented adaptation of the BLEU. Instead of using n-grams, ROUGE uses the longest common subsequence to compute the F1 score. Moreover, we also use the diversity metric, to compute the number of distinct n-grams generated by the model [288]. In addition, we also compute DistS-n, which is the average number of distinct tokens across situations. Computed perplexities were in the range of 15-25, but these do not reliably indicate performance as the vocabularies for BART and FlanT5 are different.

Human Evaluation In addition to automatic metrics, we also perform a human evaluation using Prolific. Due to the costs of human evaluation, we only performed a human evaluation for our top two models, FlanT5 + TE (PS), BART + TE (PS), and FlanT5 + SD (C) which was the highest-performing style decoder model. We randomly sample 100 examples from the test set and conduct our human evaluations in two parts. Our development of the human evaluation answers the first sub-question: How should we evaluate perspective taking through the lens of NLG?

In the first part, we focus on matching self-disclosures with the generated comments. Our initial human evaluation was similar to that of prior work, which measured persona consistency. Annotators were asked if a response was consistent with a set of self-disclosure statements when presented with the same set provided to the model. We found that in almost every case across 50 examples and two annotators, the answer was yes. This evaluation is insufficient for our task where it is unlikely for self-disclosures to be directly stated or even rephrased in someone's comments; there is a less direct connection between beliefs reflected in written verdicts and the personal background reflected in self-disclosures.

Instead, we developed a ranking evaluation. Others have used a ranking of models as an evaluation, but have not ranked the response with distractor human responses [274, 289]. In our setup, we show the annotators a set of $k = 20$ most relevant persona sentences from a user a, and three comments: the comment of author c_a^s, the generated comment from the model for that user, and a comment $c_{a'}^s$, written by another user a', for the same situation s. Then we ask the annotators to rank the comments with respect to the "possibility that they have been written by the user with the given self-disclosure statements." Ranking with both correct and incorrect human responses allows us to more clearly understand model performance. This controls for the fact that multiple perspectives can be plausible for a given situation. The model must connect the perspective and the personal background. It is more difficult for models to be ranked over the ground truth than it is to outperform other human responses. We find that 70.8% of rankings have the correct human response over the incorrect one. This gives us an upper bound on model performance. In the second part of our evaluation, we focus on the fluency and relevance of the comment with respect to the situation. We show annotators the situation summary title s, and two comments: the gold comment c_a^s, and the corresponding generated comment from our model. We ask the annotators to pick the most fluent comment and the most relevant comment about the given situation summary.

Model	BL-1 ↑	BL-2 ↑	R-1 ↑	R-L ↑	Dist-1 ↑	Dist-2 ↑	DistS-1 ↑	DistS-2 ↑
PAA [281]	15.0	5.1	18.9	16.3	0.01	0.06	0.41	0.53
BART + Priming	4.6	1.9	18.4	14.8	0.02	0.14	0.52	0.61
BART + User Id	4.1	1.7	18.7	15.2	0.03	0.15	0.54	0.63
BART + TE (S)	9.9	4.2	25.4	**19.7**	0.033	0.17	0.5	0.57
BART + TE (C)	5.0	2.4	18.8	15.6	0.029	0.14	0.52	0.62
BART + SD (S)	4.2	2.0	19.1	15.8	0.03	0.15	0.41	0.55
BART + SD (C)	5.8	2.4	23.5	18.8	0.03	0.16	0.47	0.63
FlanT5 + Priming	10.7	4.2	15.7	13.6	0.02	0.1	0.59	0.75
FlanT5 + User Id	5.7	2.4	19.9	15.7	0.029	0.14	0.61	0.77
FlanT5 + TE (S)	**25.3**	**9.0**	**25.6**	17.6	**0.053**	**0.387**	**0.73**	**0.92**
FlanT5 + TE (C)	7.6	2.9	18.2	12.0	0.032	0.25	0.62	0.73
FlanT5 + SD (S)	11.9	5.1	17.1	11.4	0.04	0.29	0.65	0.8
FlanT5 + SD (C)	18.3	5.9	18.8	12.5	0.04	0.29	0.64	0.79

Table 7.2: Automatic metrics of fine-tuned models, for our based models with priming, user id, twin encoder (TE), and style decoder (SD). We report BLEU-1 (BL-1), BLEU-2 (BL-2), ROUGE-1 (R-1), ROUGE-L (R-L) scores in the range of 0-100 and diversity metrics Distinct n-grams (Dist-n), and Distinct n-grams across situations (DistS-n) in the range 0-1. (S) means the model uses self-disclosure sentences as additional information, (C) past comments. The auxiliary set of information is extracted using the most similar method.

7.7 Results and Analysis

Similar self-disclosure statements were most useful for perspective taking. In Table 7.2, we report the automatic results for all combinations of architectures from our models. Furthermore, BART-based models were the most sensitive with respect to the retrieval method used to extract the set of auxiliary sentences. When random self-disclosure sentences or comments were utilized, the generation of the BART-based model would degrade, and upon manual inspection of the results, the generated output would contain only NTA/YTA tokens.

Moreover, in Table 7.3, we report the similarity of the generated text with the auxiliary sentences provided. We provide the maximum, mean, and minimum similarity between the verdict and the sentences in the extracted set. In addition, we provide as a reference, the similarity between the

Model	Max ↑	Mean ↑	Min ↑
Gold Reference	0.31	0.1	-0.03
PAA [281]	0.26	0.08	-0.05
BART + Priming	0.26	0.08	-0.05
BART + User Id	0.25	0.07	-0.05
BART + TE (S)	0.27	0.08	-0.05
BART + TE (C)	0.27	0.08	-0.05
BART + SD (S)	0.25	0.07	-0.05
BART + SD (C)	0.26	0.08	-0.05
FlanT5 + Priming	0.25	0.08	-0.05
FlanT5 + User Id	0.25	0.08	-0.05
FlanT5 + TE (S)	**0.29**	**0.1**	**-0.04**
FlanT5 + TE (C)	0.25	0.08	-0.05
FlanT5 + SD (S)	0.26	0.08	-0.05
FlanT5 + SD (C)	0.26	0.08	-0.05

Table 7.3: We provide the maximum (Max), mean and minimum (Min) similarity between the verdict and the auxiliary sentences in the extracted set.

original judgments and the auxiliary information. We notice that our FlanT5 + TE (S) model has the best results and the closest to the gold reference.

In Table 7.4, we report the results for FlanT5 + TE (PS), with different numbers of self-disclosure sentences as context. Our experiments are run with $\{5, 10, 15, 20, 25, 30\}$. We notice that the best-performing model uses 20 self-disclosure sentences. However, the differences between the models' performance are small, and one can trade off small performance values, with computational speed-up, by using only the top-5 self-disclosure sentences. Incorporating more information is helpful up to a point where the context gets longer and includes fewer similar statements.

Sentences	BLEU-1	BLEU-2	R-1	R-L
5	24.1	8.4	25.4	17.7
10	24.6	8.8	26.0	18.2
15	24.4	8.7	25.8	18.0
20	25.3	9.0	25.6	17.6
25	24.1	8.3	25.0	17.5
30	22.9	7.7	24.7	17.4

Table 7.4: Automatic metrics (R=ROUGE) of the FlanT5 + TE (PS) model with varying number of self-disclosure sentences in the range $[5 - 30]$.

The twin encoder architecture performed best for both models. The key difference between the two architectures is that information about the situation and the auxiliary context is combined. In the twin encoder, information is combined before the decoder performs the cross-attention with the encoder states, while in the style decoder, the information is combined after the decoder. Hence, in our case, it proved to be more useful to use only one decoder layer and combine the information earlier, as opposed to previous work [272]. In general, the FlanT5 variations proved to perform better, which may be attributed to the size difference of the base models (250M vs 140M). In addition, FlanT5 + TE (PS) performs better than the PAA model despite having fewer parameters. Moreover, FlanT5 + TE (PS), has the most diverse responses, even across situations, with scores close to the original responses on Reddit.[2] Among priming and user ID, that do not require any architecture changes, priming was better. However, in the case of FlanT5 + priming, it generated excessively long responses resulting in nonsense judgments.

Model	**BLEU-2**	**Dist-2**	**DistS-2**
Llama2-7B FT (S)	**6.5**	0.16	**0.93**
Llama2-7B (S)	5.4	0.1	0.63
Llama2-13B (S)	4.8	0.15	0.82
Llama2-70B (S)	5.8	0.14	0.49
Llama2-7B (C)	4.6	0.053	0.8
Llama2-13B (C)	6.4	0.22	0.9
Llama2-70B (C)	**6.5**	**0.27**	0.81
Llama2-7B (P)	3.5	0.02	0.77
Llama2-13B (P)	4.5	0.12	0.88
Llama2-70B (P)	6.4	0.25	0.86

Table 7.5: Automatic metrics for different Llama2 models prompted with: 1) self-disclosure (S), 2) comments (C), 3) pairs of past situation/comments, and fine-tuned (FT) version of Llama2-7B model.

LLMs performed worse than tailored architectures. Table 7.5 shows the results of Llama2 variants for different prompts. Large versions of Llama2 13B and 70B models perform best when the prompt contains past examples of user comments (C). On the other hand, the performance drops for both 7B and 13B models, when utilizing pairs of past situations/comments of the author. This drop in performance may be attributed to the expanded context size resulting from the incorporation of past situations. Additionally, we also report the results for the fine-tuned Llama2-7B model with self-disclosure sentences. Performance of the model is improved compared to in-context learning with prompting, reaching the performance of Llama2-70B model. Nevertheless, all the large models performed worse than our top model, despite having almost 100 times more parameters.

Model	**Generated over Incorrect ↑**	**Generated over Correct ↑**
BART + TE (S)	62.8%	38.9%
FlanT5 + TE (S)	**67.2%**	**42%**
FlanT5 + SD (C)	49.4%	39.4%

Table 7.6: Human evaluation results related to the ranking of comments with respect to the given persona. Correct is ranked over incorrect 70.8% of the time, providing an upper bound for generated over correct.

Human evaluation confirms the superiority of our model, showing a gap between human performance. In Table 7.6 we show the results for the first part of the survey, which is related more to

[2] DistS-1 and DistS-2 for original comments on Reddit were 0.76 and 0.93 respectively.

Model	Fluency ↑	Relevance ↑
BART + TE (S)	**43%**	**42%**
FlanT5 + TE (S)	30.6%	25.6%
FlanT5 + SD (C)	41.7%	40%

Table 7.7: Human evaluation results for our top two models BART and FlanT5 fine-tuned with Twin Encoder (TE) with self-disclosure sentences (S), and FlanT5 + Style Decoder (SD), with comments.

alignment between the generated response and the self-disclosure sentences of the user. We report the average accuracy for the number of times the generated comment was higher in rank over the incorrect and the correct one. FlanT5 + TE (PS), is performing the best across all metrics, with almost 5% better accuracy in selecting the generated comment over the incorrect one. This finding suggests that the more diverse responses align closer to the self-disclosure sentences of the users. The generated is ranked over the correct response 42% of the time, showing room for improvement, as we would expect this to be close to 50% if they were indistinguishable. The agreement between annotators is 0.45 for the FlanT5 + TE (PS), which is a moderate agreement, while the other two models show fair agreement with 0.27 and 0.22. The results for the human evaluation related to comment fluency and relevance, are shown in Table 7.7. We report the average accuracy of human annotators in selecting the generated comment in the evaluation. Human annotators selected the BART + TE (PS) model most often. The main reason for these results might be due to the length of the comment. BART + TE (PS), on average, has shorter responses (25.3 for BART versus 49.9 for FlanT5). The Cohen Kappa for these annotations is 0.3 for FlanT5 + TE (PS), 0.27 for BART + TE (PS), and 0.24 for FlanT5 + SD (C), which shows a fair agreement between the annotators.

	Verdict			Situations			Authors		
Model	**BLEU-1**	**Acc**	**F1**	**BLEU-2**	**Acc**	**F1**	**BLEU-1**	**Acc**	**F1**
Avg. Embed [16]	-	**86.1**	**83.3**	-	66.5	56.2	-	**86.0**	**83.2**
BART + TE (PS)	4.2	83.4	79.2	3.1	**68.1**	51.2	8.6	74.0	62.2
FlanT5 + TE (PS)	9.0	77.1	73.5	22.2	65.5	**56.8**	23.1	76.2	72.8

Table 7.8: Perspectivist classification for our best two models and the best model from previous work, the averaged embeddings (Avg. Embed).

Generated perspectives are competitive with perspectivist classification. As a follow-up analysis, we evaluated our model on the perspective classification task from Plepi et al. [16] by extracting the labels (NTA/YTA) from the generated comments. We use their splits; the verdict split (default for our generative experiments), situation, and author splits, which have disjoint sets of situations and authors respectively, across train, validation, and test.

Table 7.8 presents the results of perspective classification for our top two performing models, compared to the personalized model with average embeddings [16]. The previous work performs better on author and verdict splits. It has the advantage of training explicitly for the classification task. Their average embedding model was the highest performing overall, though their priming method achieved 69.6% accuracy on the situation split (Section 5.5), which proved to be the most difficult split. Our FlanT5 + TE (PS) model has a slightly better F1-score for the situation split

Situation	FlanT5 + TE (S)	BART + TE (S)
AITA for clapping back on a woman who turned out to be deranged after she made a huge scene on me buying meat products in a grocery store?	NTA - she's gonna learn a valuable lesson from it. It doesn't help her to be a jerk.	NTA. It's not your fault she didn't get help. If she wanted a confrontation she should have done it herself.
AITA for being cold to my ex-fiancé and demanding a paternity test after her infidelity?	NTA. It's an unfortunate situation and the kid is going to end up yours, and you have every right to go out of your way to make it right.	NTA. You're not responsible for her life, and you don't owe her anything. Good on you for sticking up for yourself.
AITA for expecting my wife to do more of the housework because I bring in more income?	YTA There are times where you have to justify your sex life, or try to "compromise" with your partners. Your wife (the A) is willing to do the bulk of the housework and is willing (the B) to work hard	YTA. You are the asshole for expecting your wife to work more hours than you are for asking her to do more work.
AITA for choosing my girlfriend over my parents?	NTA. Your parents are toxic, they will only ruin your life the next time. And it is the least that you could do.	NTA. Your parents are the assholes.
AITA for not wanting to cover/swap shifts with coworkers who are 'sick' with period pain?	NTA They need to see a doctor and be able to manage their own period pain. Just tell them to go get a different doctor and take care of yourself first.	YTA. You don't have to cover someone else's shift just because they're in pain. It's not like you're a bad person for not taking care of your coworkers.
AITA for being angry at my sister for not attending our fathers funeral because of his criminal history?	YTA. She is a convicted murderer, not you. She is angry at him.	YTA. It's your fault not hers.

Table 7.9: We show the generated outputs from two of our top models, for different situations.

Generated Examples Table 7.9, shows generated examples from our best models, FlanT5 + TE (S) and BART + TE (S) for the same personas per situation. We pick a variety of situations with different perspectives. This table shows the relevance and fluency of the generated outputs with respect to the situation. In addition, we report in Table 7.10, several comments per situation generated from our model, for different personas. Here we can notice the diversity of generated comments and perspectives per situation, given different personas.

Furthermore, we note in Table 7.11, samples of self-disclosure sentences from a user in our dataset, together with the generated texts from our FlanT5 + TE (S) model and the original comment. One can infer from the sample that the person giving the judgment had kids while in an intimate relationship and is no longer in that relationship. They give a judgment in similar situations related to these experiences, the first one involving an ex, and the other one involving marriage. The first situation, is a controversial one, with a ratio of original NTA/YTA verdicts equal to 0.47. We notice that in both cases, our model correctly predicts the verdict, together with the rationale behind it.

Situation	FlanT5 + TE (S)
AITA for dating a girl because of her sexual history?	NTA It's reasonable for you to not be interested in a particular type of person YTA. Regardless of your feelings and intentions, you didn't date her because of her sex life. YTA, thats a huge asshole move. If you wanted to break up with her, you should have had a talk with her.
AITA for saying no to celebrating my nephews birthday at my wedding?	NTA, that's your wedding to do as you please. YTA. She's already told you she wouldn't have a big bday celebration
AITA for not letting my partner see my search history?	NTA, in all honesty and trust you are 100% secure, in the world and trust should never be abused NTA -your girlfriend sounds weirdly controlling and untrusting YTA. my husband does this too, and he is probably hiding stuff. I guess you too.

Table 7.10: We show the generated outputs from our top model, for different situations and different personas.

Self-Disclosure Sentences: I think once my kids get busy with activities and sports and things it will be like oh, dad used to date her and she had kids... But given the distance I don't see them maintaining a relationship. I do have a college fund that is funded by my side of the family for them - my ex doesn't contribute to that. I like the idea of simply thanking her and being honest that you can't reciprocate right now, but appreciate the thought. Just to get it off my plate and to also set the boundary that if it's not emergent, it's not something I will respond to immediately. I always tell my kids that it's ok to like something and not like other things, but it's not ok to be dismissive to others of what they enjoy. I explained to my kids that all parents (whether it's a friend's mom or their dad) have different rules for their homes.
Situation A: AITA for going into my ex-mother-in-law's to chit chat after dropping my kids off? **Generated:** NTA. Your ex and MIL are being very childish. Your ex didn't want you and his mom to talk so much. There was no reason for her not to be friendly with you. **Gold:** NTA. Here's why I think this is true, in a divorce everyone is hurt, everyone has an opinion about what relationships need to be maintained and cut off. Your ex-MIL still wants to be friendly with you, that's her position. She invited you in. You still enjoy talking to your ex-MIL so you accepted reluctantly. Your ex does not like that you and his mom are friendly, which is fair on his part too, he has another relationship.
Situation B: AITA for not wanting to attend my friend's wedding since I am not a bridesmaid? **Generated:** NTA. I understand the bridesmaids feeling you wanted to be there for her, but you can't choose what her party is. That's just a problem for her. **Gold:** NTA. I don't think it's wrong to avoid the wedding considering you state it's a financial burden that you'd rather avoid if you aren't in the wedding. You thought your friendship was in a different place than she did. She's not picked you, not an asshole for being upset - you are entitled to your feelings.

Table 7.11: Sampled self-disclosure sentences from a user in our dataset, together with the generated texts from our FlanT5 + TE (S) model and the original comment.

7.8 Summary

In this chapter, we addressed the fourth research question, focusing on personalized language generation and the type of information that is more helpful for generating personalized language.

We discussed the limitations of previous work on persona-based dialog and three areas of improvement. First, we investigated the differences between artificial and realistic personas and introduced the PersonaSocialNorms corpus in Section 7.3, which contains real self-disclosure comments from the authors and judgments of conflict situations. In addition, we formulated the problem of language generation, given a data point, a post, and a personal verdict, together with the corresponding self-disclosure sentences of the author. Second, we studied which information is most useful to perspective taking, finding that self-disclosure statements that were similar to the situation were most useful. Additionally, we explored the trade-off between the amount of additional personalized information and the performance of the models. Third, we compared tailored architectures to LLMs, including two novel methods, finding that our twin encoder architecture outperformed recent work, FlanT5, and Llama2 models. Lastly, we found that previous consistency evaluation metrics were inadequate and proposed a human ranking evaluation that includes similar human responses. Additionally, we found that our generation model performed competitively with previous work on perspective classification.

CHAPTER 8

Conclusions and Future Work

In this closing chapter, we summarize the significant contributions made throughout this dissertation. We elaborate on the main findings, related to each research question defined in Section 1.3. Moreover, we acknowledge the limitations encountered during our research. Recognizing these limitations is crucial, as it lays a foundation for continuous improvement and advancement in the field. Building upon this foundation, we identify promising research directions that can build on the findings of this dissertation, proposing new inquiries and methodologies that could further enrich the field.

8.1 Conclusions

> ***RQ1*: Can we enhance text classification tasks by incorporating authors' context?**

In addressing this research question, we explored social networks of authors interactions, and contextual information to enhance text classification tasks.We focused on the sarcasm detection task as a case study, due to the subjectivity in its interpretation and the difference between the intended meaning of the utterance and its literal meaning. Therefore, this task exemplifies how additional context can significantly improve text classification. To facilitate this, we collected a dataset from Twitter (now X) for sarcasm detection, using an enhanced approach of reactive supervision method [87]. This method, allowed us to collect additional conversational and user context, as well as to capture different types of sarcasm. Moreover, we developed a heterogeneous graph structure that integrated both user and text nodes, along with their interconnecting edges. Our approach centered on employing a deep graph attention-based model, that can jointly model both users' social networks and the relations between a sarcastic tweet and the conversational context. Our results showed, that this graph structure significantly outperforms state-of-the-art results, underscoring the importance of modeling social network interactions as graphs, jointly with text, as an effective way to learn better representations for both text and users.

The key contributions of our work for *RQ1* include:

- **Model Development:** We present the first graph attention-based model to identify sarcasm on social media by explicitly modeling authors' social and historical context jointly, capturing complex relations between a sarcastic tweet and its conversational context.

- **Performance Enhancement:** We demonstrate that exploiting these relationships increases performance in the sarcasm detection task, reaching state-of-the-art results on the recent SPIRS dataset [87], which we expand with author history. We examine the impact of different parts of the context, captured by attention weights, in modeling sarcastic utterances.
- **Dataset Collection:** We collect a new dataset on Twitter by extending a semi-supervised method that uses reactive supervision and provides additional contextual information.
- **Insights into Sarcasm Detection:** Our experiments revealed that while user-based models are adept at detecting an author's sarcastic intentions, they are less effective in identifying how sarcasm is perceived by others.

> ***RQ2*:** How can we model the context of recipients to accurately predict their responses to various discourses?

In addressing *RQ2*, we focused on modeling the context of recipients to accurately model their responses to various discourses. To tackle this problem, we constructed a dataset from Reddit, that contains description of real-life conflicts, and users' perspectives regarding these conflicts. Additionally, we annotated a set of 500 conflicts with six aspects of conflict. Moreover, we introduced a novel problem setting, of predicting whether someone will perceive the actions of one individual as right or wrong in a given situation. To model annotators, we explored several personalization methods, namely average embeddings, priming, authorship attribution, graph attention network, and annotator ID. Overall, we found that averaging embeddings provided a strong and relatively simple approach. The authorship attribution and graph attention networks were consistently high-performing across splits, while for the situation split, annotator-level accuracy was highest with the priming approach. These methods outperform the common approach of representing authors with a single ID. We performed our experiments by applying different data splits and analyzing the results by individuals, demographics, available data, and the type of task. Our analysis, across tasks, showed a direct correlation between the closeness of the relationship between parties in conflict and the effectiveness of personalization. Our key findings show the importance of personalization in modeling recipients, in order to better understand their perspectives across various real-life social situations.

The key contributions of our work for *RQ2* include:

- **Dataset Collection:** We utilize Reddit community to construct a dataset that contains individual assessments of conflict situations. Additionally, this dataset includes: 1) Clustering of the descriptions of social situations involving interpersonal conflict 2) a set of 500 conflicts annotated with six aspects of conflict 3) Annotators' context, including their past comments together with their demographics.
- **Comprehensive Analysis:** We conducted a thorough analysis of model performance, considering factors such as individual user characteristics, demographic groupings, the volume of available data, and task type.
- **Problem Setting:** A discussion of the relation between data perspectivism and personalization, and introducing a new problem setting for subjective task analysis.

- **Methodological Comparison:** A novel comparison of various personalization methods under the new problem setting.

> ***RQ3*:** Do the dynamic user representations help to capture the temporal behavior of users related to their social networks?

To address this research question, we explored datasets and methodology to model and capture temporal behavior. We introduced FACTOID, a new user-centered dataset for misinformation spreader analysis, that focuses on the political discussions on Reddit from the beginning of January 2020 until April 2021. Apart from the fake news/real news distinction, the dataset contains fine-grained labels about the users' credibility level and political bias. First, we conducted experiments under the static setting, for identifying misinformation spreaders. We provided various methods that utilize the social connections between the users along with their psycho-linguistic features.

In addition, we provided a temporal analysis of our dataset, exploring the user behavior over a range of months. We found different peaks and fluctuations in the percentages of user-to-user relationships, influenced by a variety of events and their behaviors. These initial findings suggested that understanding the temporal dynamics of user behavior could offer more insights than static analysis alone. To better capture the dynamic context of users and their interactions, we proposed a dynamic graph neural network framework that generates temporal graph representations from the users' semantic similarities and social interactions over time. Our extensive experiments and ablation study demonstrated that the temporal graphs are more efficient than content-based models or simple static graphs for predicting (a) the future misinformation spreading behavior, (b) the behavior of unseen users, and (c) misinformation spreading behavior in a zero-shot scenario. Through exploratory experiments, we analyzed the various aspects of the framework to provide insight into its usability. These experiments showed that dynamic semantic similarities lead to better results than social ones. The ablation study on the components of the model revealed that the temporal modeling of the users' semantic similarities and social interactions significantly contributes to identifying misinformation spreaders effectively. Our error analysis indicated that the misclassified fake news spreaders tend to post a very low number of fake news posts and a high number of real news posts from highly credible sources. Yet, the proposed framework is applicable as a human moderator-assistance tool for identifying users who post fake news more consistently.

The key contributions of our work for *RQ3* include:

- **Dataset Collection:** We introduce FACTOID, a user-level **FAC**tuality and p**O**litical b**I**as **D**ataset, that contains a set of 4,150 news-spreading users with 3.3M Reddit posts in discussions on contemporary political topics, covering the time period from January 2020 to April 2021 on individual user level.

- **Static Setting Experiments:** We conducted classification experiments to identify misinformation spreaders in a static setting, leveraging users' social networks and a combination of their posting history representations and psycho-linguistic features.

- **Dynamic Framework Development:** We develop a dynamic graph neural network framework for (a) predicting the users' future misinformation spreading behavior, (b) predicting the behavior of unseen users, and (c) predicting misinformation spreading behavior in a zero-shot scenario.

- **Performance:** We show that our proposed dynamic framework outperforms the baseline content-based models as well as the static graph model, highlighting the effectiveness of temporal graph representations.

> ***RQ4*:** How effectively can generative models adopt personalized perspectives using user context, and what strategies can be employed to incorporate this context into the models?

To address this research question, we explore a personalized language generation task. We examined this through the lens of generating perspectives on conflict situations. In this study, we utilized PersonaSocialNorms dataset containing judgments, and realistic self-disclosure statements written by the annotators. Initially, we studied which information is most useful for perspective taking, finding that self-disclosure statements that were similar to the situation were most useful. Furthermore, we encourage the use of representative and diverse self-disclosure sentences or past comments written by the user related to the topic. We experimented with ways to incorporate the self-disclosure statements, in different adapted encoder-decoder architectures. In the first architecture, twin encoder (TE), we extend the transformer architecture with an additional encoder, to capture the personalized information, while in the style decoder (SD) architecture, we utilize an additional decoder, in order to focus on personalized features during generation. We compared our tailored architectures to an autoregressive Llama2 model, finding that our twin encoder architecture outperformed recent work. Additionally, we found that our generation model performed competitively with previous work on perspective classification.

Lastly, we found that previous consistency evaluation metrics were inadequate and proposed a human ranking evaluation that includes similar human responses. To evaluate our models, we applied this new evaluation schema by asking humans to rank the human response, model output, and a distractor human response.

The key contributions of our work for *RQ4* include:

- **Dataset Adaptation:** We construct a corpus, containing 95K judgments of social situations written by 6K authors, together with their self-disclosure statements.
- **Model Development:** We design two transformer architectures twin encoder and style decoder, tailored for incorporating auxiliary personalized information by extending the encoder or decoder of original transformers, and find that our twin encoder approach outperforms LLMs.
- **Evaluation Setting:** We develop a novel evaluation by asking humans to rank the human response, model output, and a distractor human response, combining approaches from persona consistency.

8.2 Limitations and Future Work

This thesis has presented our research endeavors in learning dynamic user representations on social media, employing various methodologies across different tasks. Although we achieved adequate results to validate our research questions, these contributions are intended to ignite further exploration and discussion in this field. However, like all research, ours too comes with its limitations, some of which were not fully addressed within the scope of this thesis and can be explored in future work. We list the following main limitations, together with the future work that can extend our work to tackle these limitations:

- **Data Quality:** Throughout our research, we constructed various datasets where some of those focus on a) text classification, expanded with additional user information, and b) user classification. Expanding the existing datasets with user information might result in sparse social networks, due to the primary focus of the dataset being on the text. Such datasets usually contain 1) only one text of a user, having a 1:1 mapping between user and text, and 2) text from users that do not interact often, and also users that have no interactions with each other, which may result in graphs that have a low density. Future work could enhance data collection for personalized NLP tasks by following more closely user relationships and interactions. Such work, can collect datasets by focusing on an initial random seed of users, and further expand upon their texts, or their networks before collecting another seed of users to repeat the process. In such a way, we make sure to construct high-density graphs and better explore their social interactions related to the specific topic of interest. However, upon collection of the initial seed of users, one should pay attention to having a diverse set of users, such as they contain a diverse set of interests, behaviors, perspectives, and opinions.

- **Automatic User Labels:** Automatically labeled datasets should be utilized with caution since they might include wrongly labeled posts and, by extension, wrongly labeled users. For example, upon manual inspection, many posts contained multiple links from mixed sources (credible and non-credible). We argue that the ratio of the non-credible to credible news sources posted in one post should be considered as a labeling threshold instead. More specifically, if more than half the sources within one post are non-credible, only then should it be labeled as misinformation. We acknowledge that there is a very thin line separating real news spreaders and misinformation spreaders.

  Future research could refine these methods, possibly introducing a new category of 'potential misinformation spreaders' for users whose posts fall into ambiguous territory. To further improve the automatic labels, one can increase the amount of manual annotations. For example, we can constrain the topic of interest for capturing misinformation spreaders, and train annotators to separate users that tend to spread misinformation or not for that particular topic. This can be explored as an intersection with additional labels for that topic, like the users' stance, or their beliefs.

- **Large Scale User History:** Utilizing extensive user histories can introduce noise unrelated to the task at hand, potentially affecting certain methods like priming. While initial attempts at cleaning and processing this data were made, future research could explore more sophisticated techniques to enhance the quality and relevance of user history data.

- **Context Size:** During the time of this research, we were often constrained by models' maximum context length. By improving the quality with more manual labeling and reducing the size of user information, future work can further explore the priming method, which is similar to retrieval augmented generation (RAG). With advancements in large language models (LLMs) and increased context sizes, future research could delve deeper into these areas, both in the training and evaluation phases, for better controlling personalized generation outputs.

Future Work

The field of personalized Natural Language Processing (NLP) is rapidly evolving, particularly with the advancements in Large Language Models (LLMs) and expanding context sizes. Our research, while primarily centered around social media data, has the potential to be extended to other domains, such as healthcare, automotive assistance, finance, law, and beyond. For instance, our preliminary work [290] in modeling disease paths with anonymized patient data showcases the potential for applying these methodologies in medical contexts. Our graph modeling approaches can be adapted to capture the relations between disease paths and patient histories, and are likely to significantly enhance medical diagnostic processes.

Moreover, there exists promising research for augmenting medical assistant chatbots with the ability to customize interactions based on the nuanced preferences and behaviors of users. By embedding personalized attributes into these medical assistants, there's an opportunity to improve their emotional and cognitive empathy, thereby fostering a deeper connection with users [291, 255]. This improvement in empathetic interaction requires the systems to not only grasp the textual content, but also to carefully interpret the underlying emotions and viewpoints of the users. Furthermore, our temporal Dynamic Graph Neural Network (DyGNN) framework can be adapted to interpret the temporal dynamics of an individual's emotional state, and enhance the system's understanding of their perspectives.

Another interesting research direction consists of transferring the approaches introduced in this work to personalized sales assistants, moving beyond traditional recommendation systems. These personalized sales assistants, interact with the customers, by engaging in dialogues that exchange subjective product opinions [292]. This approach can be further extended to improve the assistants' understanding of both the customers' past and present preferences, thereby enhancing the quality of recommendations and overall satisfaction with the purchasing experience. Additionally, the heterogeneous graph modeling introduced earlier in Chapter 4, offers a promising direction on being used to capture relations between users and items, in order to predict preferred items of new users based on their social network similarity.

Moreover, we also propose to further explore the challenge of cold-start users. Such users may lack sufficient additional context, making it difficult to compute an initial representation for them. Future work can explore different strategies on how to utilize the current users to initialize cold-start users and their effect on the performance of NLP systems. Another line of research can focus on grouping users based on their interests, behavior, and perspectives, in order to capture group context instead of individual context. Grouping similar users in a single representation can help to reduce the number of computations and keep the number of additional representations fixed (contrary to having to add new user representations for each unseen user), in addition to reducing the sparsity of individual context. For instance, SBIC dataset [293], contains demographic information for their annotators, which can be further utilized to group annotators in order to reduce their sparsity in annotations.

On a broader scale, this thesis has laid the groundwork for representing and incorporating user context into NLP systems to enhance their performance. The benefits of our approaches to learning user context can be transferred to learning any additional context, such as triples from a knowledge graph (KG), which can serve as an additional context that supports a piece of text [294, 295]. Given an external KG, one can encode the subgraphs, in order to improve performance in tasks like fact-checking [296], or open domain question answering [297]. For instance, the emerging area of LLM has achieved a significant increase in their language understanding and generation capabilities. However, such advancements, come with a concerning trend of LLMs tendency to generate text that is not factual or faithful, referring as hallucinations [298]. By augmenting LLMs, with Graph Neural Network (GNN) sublayers that can model input queries alongside with supporting subgraphs from KGs, it is possible to significantly enhance the factual accuracy and reasoning abilities of these models [299, 300]. Such an approach not only addresses the issue of hallucinations but also paves the way for more reliable and contextually informed NLP systems, capable of overcoming challenges of real-world information and knowledge representation.

Future research can build on the foundations laid by the contributions of this thesis, potentially leading to the development of personalized systems across various domains. The methodologies and resources we have developed offer a robust starting point for future explorations, aiming to enhance the personalization and accuracy of NLP systems in diverse settings.

8.3 Ethical Considerations

The ability to automatically approximate personal characteristics of online users in order to improve natural language classification algorithms requires us to consider a range of ethical concerns, including: (1) privacy and user consent, (2) representativeness of the data for generalization, and (3) user vulnerability to a potential model or data misuse or misinterpretation.

Use of any user data for personalization shall be transparent, and limited to the given purpose, no individual posts shall be republished [301]. Researchers are advised to take account of users' expectations [302, 303, 304] when collecting public data such as Twitter (now X), or Reddit. In this case, when we collect a dataset from scratch, or expand the original datasets with a more extensive user history, we utilize publicly available Twitter or Reddit data in a purely observational [305], and non-intrusive manner. All user data is kept separately on protected servers, linked to the raw text and network data only through anonymous IDs.

Shah et al. [306] identify four different sources of bias in NLP models: selection bias, label bias, model overamplification, and semantic bias. While we can't exclude any of those, the selection bias should be kept in mind in particular, when reusing the presented model, as it is unclear to which extent augmented datasets with user history represent a sample of the overall population on social media.

In addition, any user-augmented classification efforts risk invoking stereotyping and essentialism, as the algorithm may lean towards labeling people rather than posts (e.g. "this is a sarcastic person" or "this is a misinformation spreader"). Such stereotypes can cause harm even if they are accurate on average differences [307]. These can be emphasized by the semblance of objectivity created by the use of a computer algorithm [308]. It is important to be mindful of these effects when interpreting the model results in its own end-application context. Furthermore, personalization, as presented in our work, can be used to automatically infer people's opinions about social norms and other subjective stances, sentiments, or perceptions. This could be desired in some applications but could be undesired or even harmful in others. Bias in models can cause misrepresentation and negatively impact populations [309, 141]. These populations may be represented with the demographics we identify, or may result from sample bias. Those not well represented in our data may be negatively affected by the application of models such as ours, depending on their use. Although the implications depend on the application, we generally suggest that if such a method is used in practice, that end-users are made aware of how their data is being used and given the choice to not be a part of automated decision processes based on these inferences.

Bibliography

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, arXiv preprint arXiv:1301.3781 (2013) (cit. on pp. 1, 18, 19).

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Advances in neural information processing systems*, 2013 3111 (cit. on pp. 1, 19).

[3] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018 2227 (cit. on p. 1).

[4] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL-HLT*, 2019 4171 (cit. on pp. 1, 28, 32, 34, 35, 64).

[5] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, *Improving Language Understanding by Generative Pre-Training*, URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018) (cit. on pp. 1, 28, 31).

[6] M. McPherson, L. Smith-Lovin, and J. M. Cook, *Birds of a feather: Homophily in social networks*, Annual review of sociology **27** (2001) 415 (cit. on p. 1).

[7] M. Bucholtz and K. Hall, *Identity and interaction: A sociocultural linguistic approach*, Discourse studies **7** (2005) 585 (cit. on pp. 1, 39).

[8] P. Eckert and S. McConnell-Ginet, *Language and gender*, Cambridge University Press, 2013 (cit. on pp. 1, 39).

[9] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, "Who says what to whom on twitter," *Proceedings of the 20th international conference on World wide web*, 2011 705 (cit. on p. 1).

[10] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011 65 (cit. on p. 1).

[11] M. Del Vicario, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, *Modeling confirmation bias and polarization*, Scientific reports **7** (2017) 40391 (cit. on p. 1).

[12] S. Giorgi, K. L. Nguyen, J. C. Eichstaedt, M. L. Kern, D. B. Yaden, M. Kosinski, M. E. Seligman, L. H. Ungar, H. A. Schwartz, and G. Park, *Regional Personality Assessment Through Social Media Language*, Journal of personality **90** (2022) 405 (cit. on p. 1).

[13] J. Plepi and L. Flek, "Perceived and Intended Sarcasm Detection with Graph Attention Networks," *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021 4746 (cit. on pp. 8, 40, 47, 55, 67, 76).

[14] J. Plepi, M. Buski, and L. Flek, "Personalized Intended and Perceived Sarcasm Detection on Twitter," *Proceedings of the 3rd Workshop on Computational Linguistics for the Political and Social Sciences*, Association for Computational Lingustics, 2023 8 (cit. on pp. 8, 40).

[15] C. Welch, J. Plepi, B. Neuendorf, and L. Flek, "Understanding Interpersonal Conflict Types and their Impact on Perception Classification," *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, 2022 79 (cit. on pp. 9, 54, 56, 110).

[16] J. Plepi, B. Neuendorf, L. Flek, and C. Welch, "Unifying Data Perspectivism and Personalization: An Application to Social Norms," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022 7391 (cit. on pp. 9, 54, 120).

[17] F. Sakketou, J. Plepi, R. Cervero, H. J. Geiss, P. Rosso, and L. Flek, "FACTOID: A New Dataset for Identifying Misinformation Spreaders and Political Bias," *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, European Language Resources Association, 2022 3231 (cit. on pp. 9, 74).

[18] J. Plepi, F. Sakketou, H.-J. Geiss, and L. Flek, "Temporal Graph Analysis of Misinformation Spreaders in Social Media," *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, 2022 89 (cit. on pp. 9, 74).

[19] J. Plepi, C. Welch, and L. Flek, "Perspective Taking through Generating Responses to Conflict Situations," *Findings of the Association for Computational Linguistics ACL 2024*, 2024 6482 (cit. on pp. 10, 108).

[20] L. Flek, "Returning the N to NLP: Towards Contextually Personalized Classification Models," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020 7828 (cit. on pp. 15, 54, 108).

[21] D. Hovy and D. Yang, "The Importance of Modeling Social Factors of Language: Theory and Practice," *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021 588 (cit. on pp. 15, 110).

[22] J. Verhoeff, W. Goffman, and J. Belzer,
Inefficiency of the use of Boolean functions for information retrieval systems,
Communications of the ACM **4** (1961) 557 (cit. on p. 15).

[23] B. Miller, J. Riedl, and J. A. Konstan,
"Experiences with grouplens: Making usenet useful again,"
Proceedings of the 1997 Usenix Winter technical conference, vol. 17, 1997 (cit. on p. 15).

[24] T. D. Wilson, *Models in information behaviour research*,
Journal of documentation **55** (1999) 249 (cit. on p. 15).

[25] D. Hovy, "Demographic Factors Improve Classification Performance," *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)*, 2015 (cit. on pp. 15, 57).

[26] A. Benton, R. Arora, and M. Dredze, "Learning Multiview Embeddings of Twitter Users," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016 14 (cit. on p. 15).

[27] J. D. Carroll,
"Generalization of canonical correlation analysis to three of more sets of variables,"
APA 76th Annual Convention, San Francisco, CA, August 30-September 3, 1968, 1968 (cit. on p. 15).

[28] V. Lynn, Y. Son, V. Kulkarni, N. Balasubramanian, and H. A. Schwartz,
"Human Centered NLP with User-Factor Adaptation,"
Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017 (cit. on pp. 16, 57).

[29] M. Zamani, H. A. Schwartz, V. Lynn, S. Giorgi, and N. Balasubramanian,
"Residualized Factor Adaptation for Community Social Media Prediction Tasks,"
Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018 3560 (cit. on p. 16).

[30] Y. A. Kolchinski and C. Potts, "Representing Social Media Users for Sarcasm Detection," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018 1115 (cit. on pp. 16, 57).

[31] C. Welch, J. K. Kummerfeld, V. Pérez-Rosas, and R. Mihalcea,
"Compositional Demographic Word Embeddings," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020 (cit. on pp. 16, 58).

[32] N. Soni, M. Matero, N. Balasubramanian, and H. A. Schwartz, *Human language modeling*, arXiv preprint arXiv:2205.05128 (2022) (cit. on p. 17).

[33] Y. Yang and J. Eisenstein,
Overcoming Language Variation in Sentiment Analysis with Social Attention,
Transactions of the Association for Computational Linguistics (2017) (cit. on pp. 17, 57).

[34] R. Mishra, P. Prakhar Sinha, R. Sawhney, D. Mahata, P. Mathur, and R. Ratn Shah, "SNAP-BATNET: Cascading Author Profiling and Social Network Graphs for Suicide Ideation Detection on Social Media," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, 2019 147 (cit. on pp. 17, 41).

[35] M. Del Tredici, D. Marcheggiani, S. Schulte im Walde, and R. Fernández, "You Shall Know a User by the Company It Keeps: Dynamic Representations for Social Media Users in NLP," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019 4707 (cit. on p. 18).

[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations*, 2018 (cit. on pp. 18, 36).

[37] J. Pan, R. Bhardwaj, W. Lu, H. L. Chieu, X. Pan, and N. Y. Puay, "Twitter Homophily: Network Based Prediction of User's Occupation," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019 2633 (cit. on p. 18).

[38] P. Mishra, M. Del Tredici, H. Yannakoudakis, and E. Shutova, "Abusive Language Detection with Graph Convolutional Networks," *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)*, 2019 2145 (cit. on p. 18).

[39] S. Amir, B. C. Wallace, H. Lyu, P. Carvalho, and M. J. Silva, "Modelling Context with User Embeddings for Sarcasm Detection in Social Media," *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016 167 (cit. on pp. 19, 41, 47, 88, 92).

[40] F. Morin and Y. Bengio, "Hierarchical Probabilistic Neural Network Language Model," *International workshop on artificial intelligence and statistics*, PMLR, 2005 246 (cit. on p. 20).

[41] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, *Natural Language Processing (Almost) from Scratch*, Journal of machine learning research **12** (2011) 2493 (cit. on p. 20).

[42] M. King and P. Cook, "Evaluating Approaches to Personalizing Language Models," *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020 (cit. on pp. 21, 57, 67, 115).

[43] N. Lee, Y. Bang, A. Madotto, and P. Fung, *Misinformation has High Perplexity*, arXiv preprint arXiv:2006.04666 (2020) (cit. on p. 21).

[44] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019 3982 (cit. on pp. 21, 29, 30, 46, 61, 87, 92, 93, 112).

[45] R. W. Gibbs, *Irony in Talk Among Friends*, Metaphor and Symbol **15** (2000) 5 (cit. on p. 23).

[46] M. L. Dress, R. J. Kreuz, K. E. Link, and G. M. Caucci, *Regional Variation in the Use of Sarcasm*, Journal of Language and Social Psychology **27** (2008) 71 (cit. on p. 23).

[47] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph Attention Networks*, International Conference on Learning Representations (2018) (cit. on pp. 23, 76, 89, 96).

[48] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *3rd International Conference on Learning Representations, ICLR 2015*, 2015 (cit. on p. 23).

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in neural information processing systems*, 2017 5998 (cit. on pp. 23, 25–27, 31, 32, 112, 113).

[50] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, Neural computation **9** (1997) 1735 (cit. on p. 25).

[51] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches," *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014 103 (cit. on pp. 25, 97).

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016 770 (cit. on p. 25).

[53] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer Normalization*, stat **1050** (2016) 21 (cit. on p. 25).

[54] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, *Transformer-xl: Attentive Language Models Beyond a Fixed-Length Context*, arXiv preprint arXiv:1901.02860 (2019) (cit. on p. 28).

[55] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., *Google's Neural Machine Translation System: Bridging the Gap Between Human and Machine Translation*, arXiv preprint arXiv:1609.08144 (2016) (cit. on p. 28).

[56] W. L. Taylor, *"Cloze Procedure": A New Tool for Measuring Readability*, Journalism quarterly **30** (1953) 415 (cit. on p. 29).

[57] Y. Jernite, S. R. Bowman, and D. Sontag, *Discourse-Based Objectives for Fast Unsupervised Sentence Representation Learning*, arXiv preprint arXiv:1705.00557 (2017) (cit. on p. 29).

[58] L. Logeswaran and H. Lee, *An Efficient Framework for Learning Sentence Representations*, arXiv preprint arXiv:1803.02893 (2018) (cit. on p. 29).

[59] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A Robustly Optimized BERT Pretraining Approach*, arXiv preprint arXiv:1907.11692 (2019) (cit. on p. 30).

[60] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A Unified Embedding for Face Recognition and Clustering," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015 815 (cit. on p. 30).

[61] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A Large Annotated Corpus for Learning Natural Language Inference," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015 632 (cit. on pp. 30, 35).

[62] A. Williams, N. Nangia, and S. Bowman, "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018 1112 (cit. on p. 30).

[63] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, *Generating Wikipedia by Summarizing Long Sequences*, arXiv preprint arXiv:1801.10198 (2018) (cit. on p. 31).

[64] H. Robbins and S. Monro, *A Stochastic Approximation Method*, The annals of mathematical statistics (1951) 400 (cit. on p. 32).

[65] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv preprint arXiv:1412.6980 (2014) (cit. on pp. 32, 98).

[66] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016 1715 (cit. on pp. 32, 34).

[67] D. Hendrycks and K. Gimpel, *Gaussian Error Linear Units (gelus)*, arXiv preprint arXiv:1606.08415 (2016) (cit. on p. 32).

[68] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., *Language Models are Unsupervised Multitask Learners*, OpenAI blog **1** (2019) 9 (cit. on p. 32).

[69] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., *Language Models are Few-Shot Learners*, (2020) 1877 (cit. on p. 32).

[70] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, *Scaling Laws for Neural Language Models*, arXiv preprint arXiv:2001.08361 (2020) (cit. on p. 32).

[71] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020 7871 (cit. on pp. 32, 33, 116).

[72] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, et al., *Scaling Instruction-Finetuned Language Models*, arXiv preprint arXiv:2210.11416 (2022) (cit. on pp. 34, 35, 116).

[73] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, The Journal of Machine Learning Research **21** (2020) 5485 (cit. on p. 33).

[74] R. J. Williams and D. Zipser, *A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*, Neural computation **1** (1989) 270 (cit. on p. 34).

[75] T. Kudo and J. Richardson, *SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing*, EMNLP 2018 (2018) 66 (cit. on p. 34).

[76] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. kavukcuoglu, "Interaction Networks for Learning about Objects, Relations and Physics," *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016 4509 (cit. on p. 35).

[77] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, "Graph Networks as Learnable Physics Engines for Inference and Control," *International Conference on Machine Learning*, PMLR, 2018 4470 (cit. on p. 35).

[78] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein Interface Prediction Using Graph Convolutional Networks," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017 6533 (cit. on p. 35).

[79] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen, "Graph Convolutional Networks with Markov Random Field Reasoning for Social Spammer Detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 01, 2020 1054 (cit. on p. 35).

[80] Z. Yang, W. Cohen, and R. Salakhudinov, "Revisiting Semi-Supervised Learning with Graph Embeddings," *International conference on machine learning*, PMLR, 2016 40 (cit. on p. 35).

[81] T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto, "Knowledge Transfer for Out-of-Knowledge-Base Entities: A Graph Neural Network Approach," *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017 1802 (cit. on p. 35).

[82] H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song, "Learning Combinatorial Optimization Algorithms over Graphs," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017 6351 (cit. on p. 35).

[83] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, *Graph Neural Networks: A Review of Methods and Applications*, AI Open **1** (2020) 57 (cit. on p. 35).

[84] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *International Conference on Learning Representations*, 2017 (cit. on pp. 35, 96).

[85] Q. Liu, M. Nickel, and D. Kiela, *Hyperbolic Graph Neural Networks*, Advances in neural information processing systems **32** (2019) (cit. on p. 36).

[86] E. Reiter, "Natural Language Generation Challenges for Explainable AI," *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI 2019)*, 2019 3 (cit. on p. 39).

[87] B. Shmueli, L.-W. Ku, and S. Ray, "Reactive Supervision: A New Method for Collecting Sarcasm Data," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020 2553 (cit. on pp. 39, 41–43, 50, 125, 126).

[88] R. W. Gibbs, *On the psycholinguistics of sarcasm*, Journal of experimental psychology: general **115** (1986) 3 (cit. on p. 40).

[89] S. Dews, J. Kaplan, and E. Winner, *Why Not Say It Directly? The Social Functions of Irony*, Discourse processes **19** (1995) 347 (cit. on p. 40).

[90] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013 704 (cit. on p. 40).

[91] D. G. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," *Lrec 2014 proceedings*, ELRA, 2014 (cit. on p. 40).

[92] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, *Sentiment analysis is a big suitcase*, IEEE Intelligent Systems **32** (2017) 74 (cit. on p. 40).

[93] P. Carvalho, L. Sarmento, M. J. Silva, and E. De Oliveira, "Clues for detecting irony in user-generated contents: oh...!! it's" so easy";-," *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, 2009 53 (cit. on p. 40).

[94] D. Davidov, O. Tsur, and A. Rappoport, "Semi-Supervised Recognition of Sarcasm in Twitter and Amazon," *Proceedings of the fourteenth conference on computational natural language learning*, 2010 107 (cit. on p. 40).

[95] A. Reyes, P. Rosso, and T. Veale, *A multidimensional approach for detecting irony in twitter*, Language resources and evaluation **47** (2013) 239 (cit. on p. 40).

[96] O. Tsur, D. Davidov, and A. Rappoport, "ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 4, 1, 2010 (cit. on p. 40).

[97] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: a closer look," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011 581 (cit. on p. 40).

[98] B. C. Wallace, L. Kertz, E. Charniak, et al., "Humans require context to infer ironic intent (so computers probably do, too)," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014 512 (cit. on p. 40).

[99] A. Khattri, A. Joshi, P. Bhattacharyya, and M. Carman, "Your sentiment precedes you: Using an author's historical tweets to predict sarcasm," *Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 2015 25 (cit. on p. 40).

[100] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm Detection on Twitter: A Behavioral Modeling Approach," *Proceedings of the eighth ACM international conference on web search and data mining*, 2015 97 (cit. on p. 40).

[101] Y. Tay, A. T. Luu, S. C. Hui, and J. Su, "Reasoning with Sarcasm by Reading In-Between," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018 1010 (cit. on p. 40).

[102] D. Bamman and N. Smith, "Contextualized Sarcasm Detection on Twitter," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 9, 1, 2015 (cit. on pp. 40, 41, 47, 57).

[103] S. Amir, B. C. Wallace, H. Lyu, and P. C. M. J. Silva, *Modelling context with user embeddings for sarcasm detection in social media*, arXiv preprint arXiv:1607.00976 (2016) (cit. on p. 40).

[104] S. Oprea and W. Magdy, "Exploring Author Context for Detecting Intended vs Perceived Sarcasm," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019 (cit. on pp. 40, 55).

[105] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, "CASCADE: Contextual Sarcasm Detection in Online Discussion Forums," *Proceedings of the 27th International Conference on Computational Linguistics*, 2018 1837 (cit. on p. 40).

[106] P. Rockwell and E. M. Theriot, *Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis*, Communication Research Reports **18** (2001) 44 (cit. on p. 40).

[107] R. W. Gibbs, *Irony in talk among friends*, Metaphor and symbol **15** (2000) 5 (cit. on p. 40).

[108] M. L. Dress, R. J. Kreuz, K. E. Link, and G. M. Caucci, *Regional Variation in the Use of Sarcasm*, Journal of Language and Social Psychology **27** (2008) 71 (cit. on pp. 40, 41).

[109] A. E. Marwick and D. Boyd, *I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience*, New media & society **13** (2011) 114 (cit. on p. 40).

[110] A. Joshi, P. Bhattacharyya, and M. J. Carman, *Automatic Sarcasm Detection: A Survey*, ACM Computing Surveys (CSUR) **50** (2016) 1 (cit. on p. 41).

[111] A.-C. Băroiu and Ș. Trăușan-Matu, *Automatic Sarcasm Detection: Systematic Literature Review*, Information **13** (2022) 399 (cit. on p. 41).

[112] E. Filatova, "Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing," *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2012 392 (cit. on p. 41).

[113] D. Davidov, O. Tsur, and A. Rappoport, "Semi-Supervised Recognition of Sarcasm in Twitter and Amazon," *Proceedings of the fourteenth conference on computational natural language learning*, 2010 107 (cit. on p. 41).

[114] F. Barbieri, H. Saggion, and F. Ronzano, "Modelling Sarcasm in Twitter, a Novel Approach," *proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 2014 50 (cit. on p. 41).

[115] T. Ptáček, I. Habernal, and J. Hong, "Sarcasm Detection on Czech and English Twitter," *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, 2014 213 (cit. on p. 41).

[116] M. Khodak, N. Saunshi, and K. Vodrahalli, *A Large Self-Annotated Corpus for Sarcasm*, arXiv preprint arXiv:1704.05579 (2017) (cit. on pp. 41, 45).

[117] P. Rockwell and E. M. Theriot, *Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis*, Communication Research Reports **18** (2001) 44 (cit. on p. 41).

[118] M. L. Dress, R. J. Kreuz, K. E. Link, and G. M. Caucci, *Regional Variation in the Use of Sarcasm*, Journal of Language and Social Psychology **27** (2008) 71 (cit. on p. 41).

[119] S. Oprea and W. Magdy, "iSarcasm: A Dataset of Intended Sarcasm," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020 1279 (cit. on pp. 41, 49).

[120] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm Detection on Twitter: A Behavioral Modeling Approach," *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, Association for Computing Machinery, 2015 (cit. on p. 41).

[121] A. Khattri, A. Joshi, P. Bhattacharyya, and M. Carman, "Your Sentiment Precedes You: Using an Author's Historical Tweets to Predict Sarcasm," *Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 2015 25 (cit. on p. 41).

[122] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," *International conference on machine learning*, PMLR, 2014 1188 (cit. on p. 41).

[123] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, "CASCADE: Contextual Sarcasm Detection in Online Discussion Forums," *Proceedings of the 27th International Conference on Computational Linguistics*, 2018 1837 (cit. on p. 41).

[124] A. Grover and J. Leskovec, "Node2vec: Scalable Feature Learning for Networks," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016 (cit. on p. 41).

[125] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network Embedding As Matrix Factorization: Unifying DeepWalk, LINE, PTE, and Node2Vec," *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, 2018 459 (cit. on p. 41).

[126] B. Mathew, R. Dutt, P. Goyal, and A. Mukherjee, "Spread of Hate Speech in Online Social Media," *Proceedings of the 10th ACM Conference on Web Science*, Association for Computing Machinery, 2019 (cit. on p. 41).

[127] J. Qian, M. ElSherief, E. Belding, and W. Y. Wang, "Leveraging Intra-User and Inter-User Representation Learning for Automated Hate Speech Detection," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018 118 (cit. on p. 41).

[128] S. Chandra, P. Mishra, H. Yannakoudakis, M. Nimishakavi, M. Saeidi, and E. Shutova, *Graph-Based Modeling of Online Communities for Fake News Detection*, arXiv preprint arXiv:2008.06274 (2020) (cit. on p. 41).

[129] J. L. Davis, T. P. Love, and G. Killen, *Seriously Funny: The Political Work of Humor on Social Media*, New Media & Society **20** (2018) 3898 (cit. on p. 45).

[130] N. Lourie, R. Le Bras, and Y. Choi, "SCRUPLES: A Corpus of Community Ethical Judgments on 32,000 Real-Life Anecdotes," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 15, 2021 (cit. on pp. 53, 54, 56).

[131] V. Basile, M. Fell, T. Fornaciari, D. Hovy, S. Paun, B. Plank, M. Poesio, and A. Uma, "We Need to Consider Disagreement in Evaluation," *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, 2021 (cit. on pp. 53, 54).

[132] G. Pereira, R. Prada, and P. A. Santos, *Integrating social power into the decision-making of cognitive agents*, Artificial Intelligence **241** (2016) (cit. on p. 54).

[133] C. Ovesdotter Alm, "Subjective Natural Language Problems: Motivations, Applications, Characterizations, and Implications," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011 (cit. on p. 54).

[134] V. Basile, "It's the End of the Gold Standard as We Know It," *International Conference of the Italian Association for Artificial Intelligence*, Springer, 2020 (cit. on p. 55).

[135] A. M. Davani, M. Díaz, and V. Prabhakaran, *Dealing with disagreements: Looking beyond the majority vote in subjective annotations*, arXiv preprint arXiv:2110.05719 (2021) (cit. on p. 55).

[136] T. Fornaciari, A. Uma, S. Paun, B. Plank, D. Hovy, and M. Poesio, "Beyond Black & White: Leveraging Annotator Disagreement via Soft-Label Multi-Task Learning," *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021 (cit. on pp. 55, 57).

[137] E. M. Bender and B. Friedman, *Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science*, Transactions of the Association for Computational Linguistics **6** (2018) 587 (cit. on p. 55).

[138] J. A. Meaney, S. Wilson, L. Chiruzzo, A. Lopez, and W. Magdy, "SemEval 2021 Task 7: HaHackathon, Detecting and Rating Humor and Offense," *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, 2021 (cit. on p. 55).

[139] B. Kennedy, M. Atari, A. M. Davani, L. Yeh, A. Omrani, Y. Kim, K. Coombs, S. Havaldar, G. Portillo-Wightman, E. Gonzalez, et al., *The Gab Hate Corpus: A collection of 27k posts annotated for hate speech*, (2018) (cit. on p. 55).

[140] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A Dataset of Fine-Grained Emotions," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020 (cit. on p. 55).

[141] M. Sap, S. Swayamdipta, L. Vianna, X. Zhou, Y. Choi, and N. A. Smith, "Annotators with Attitudes: How Annotator Beliefs And Identities Bias Toxic Language Detection," *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022 5884 (cit. on pp. 55, 57, 131).

[142] E. Leonardelli, S. Menini, A. Palmero Aprosio, M. Guerini, and S. Tonelli, "Agreeing to Disagree: Annotating Offensive Language Datasets with Annotators' Disagreement," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021 (cit. on p. 55).

[143] M. Yeomans, J. Minson, H. Collins, F. Chen, and F. Gino, *Conversational receptiveness: Improving engagement with opposing views*, Organizational Behavior and Human Decision Processes **160** (2020) (cit. on p. 55).

[144] M. Forbes, J. D. Hwang, V. Shwartz, M. Sap, and Y. Choi, "Social Chemistry 101: Learning to Reason about Social and Moral Norms," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020 (cit. on pp. 56, 58, 60, 110).

[145] I. S. Efstathiadis, G. Paulino-Passos, and F. Toni, "Explainable Patterns for Distinction and Prediction of Moral Judgement on Reddit," *Proceedings of the 1st Workshop on Human and Machine Decisions (WHMD 2021)*, 2021 (cit. on p. 56).

[146] N. Botzer, S. Gu, and T. Weninger, "Analysis of Moral Judgement on Reddit," 2022 (cit. on pp. 56, 58, 63–65, 67).

[147] S. De Candia, *Modeling the boundaries of social norms online*, MA thesis, 2021 (cit. on pp. 56, 58).

[148] K. Zhou, A. Smith, and L. Lee, "Assessing Cognitive Linguistic Influences in the Assignment of Blame," *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, 2021 (cit. on p. 56).

[149] R. L. Pinkley, *Dimensions of Conflict Frame: Disputant Interpretations of Conflict.*, Journal of applied psychology (1990) (cit. on p. 56).

[150] K. A. Jehn, *A Multimethod Examination of the Benefits and Detriments of Intragroup Conflict*, Administrative science quarterly (1995) (cit. on pp. 56, 59).

[151] K. A. Jehn, *A Qualitative Analysis of Conflict Types and Dimensions in Organizational Groups*, Administrative science quarterly (1997) (cit. on p. 56).

[152] M. A. Korsgaard, S. Soyoung Jeong, D. M. Mahony, and A. H. Pitariu, *A Multilevel View of Intragroup Conflict*, Journal of management **34** (2008) (cit. on pp. 56, 58).

[153] C. Bendersky, J. Bear, K. Behfar, L. R. Weingart, G. Todorova, and K. A. Jehn, "Identifying Gaps Between the Conceptualization of Conflict and its Measurement," *Handbook of conflict management research*, Edward Elgar Publishing, 2014 (cit. on pp. 56, 58).

[154] H. Barki and J. Hartwick, *Conceptualizing the Construct of Interpersonal Conflict*, International journal of conflict management (2004) (cit. on pp. 56, 58).

[155] A. F. F. Hussein and Y. H. S. Al-Mamary, *Conflicts: Their Types, and their Negative and Positive Effects on Organizations*, International Journal of Scientific & Technology Research **8** (2019) (cit. on p. 57).

[156] F. Mireshghallah, V. Shrivastava, M. Shokouhi, T. Berg-Kirkpatrick, R. Sim, and D. Dimitriadis, *UserIdentifier: Implicit User Representations for Simple and Effective Personalized Sentiment Analysis*, arXiv preprint (2021) (cit. on p. 57).

[157] A. Jaech and M. Ostendorf, “Personalized Language Model for Query Auto-Completion,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018 700 (cit. on p. 57).

[158] C. Wu, F. Wu, T. Qi, J. Lian, Y. Huang, and X. Xie, “PTUM: Pre-training User Model from Unlabeled User Behaviors via Self-supervision,” *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020 (cit. on p. 57).

[159] C. Welch, J. K. Kummerfeld, V. Pérez-Rosas, and R. Mihalcea, “Exploring the Value of Personalized Word Embeddings,” *Proceedings of the 28th International Conference on Computational Linguistics*, 2020 6856 (cit. on p. 57).

[160] C. Welch, C. Gu, J. Kummerfeld, V. Perez-Rosas, and R. Mihalcea, “Leveraging Similar Users for Personalized Language Modeling with Limited Data,” *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022 (cit. on pp. 57, 67, 115).

[161] L. Aroyo and C. Welty, *Truth is a Lie: Crowd Truth and the Seven Myths of Human Annotation*, AI Magazine (2015) (cit. on pp. 57, 108).

[162] S. Nowak and S. Rüger, “How Reliable are Annotations via Crowdsourcing: A Study About Inter-Annotator Agreement for Multi-Label Image Annotation,” *Proceedings of the international conference on Multimedia information retrieval*, 2010 (cit. on p. 57).

[163] M. Sabou, K. Bontcheva, L. Derczynski, and A. Scharl, “Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines,” *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, 2014 (cit. on p. 57).

[164] Z. Waseem and D. Hovy, “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” *Proceedings of the NAACL Student Research Workshop*, 2016 (cit. on p. 57).

[165] M. Diaz, I. Johnson, A. Lazar, A. M. Piper, and D. Gergle, “Addressing Age-Related Bias in Sentiment Analysis,” *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2019 (cit. on p. 57).

[166] V. Prabhakaran, A. Mostafazadeh Davani, and M. Diaz, “On Releasing Annotator-Level Labels and Information in Datasets,” *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, 2021 (cit. on p. 57).

[167] M. Geva, Y. Goldberg, and J. Berant, “Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019 1161 (cit. on p. 57).

[168] A. Mostafazadeh Davani, M. Diaz, and V. Prabhakaran, *Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations*, Transactions of the Association for Computational Linguistics (2022) 92 (cit. on p. 57).

[169] F. Cabitza, A. Campagner, and V. Basile, "Toward a Perspectivist Turn in Ground Truthing for Predictive Computing," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 6, 2023 6860 (cit. on pp. 57, 108).

[170] F. Cabitza, A. Campagner, and L. M. Sconfienza, *As if sand were stone. New concepts and metrics to probe the ground on which to build trustable AI*, BMC Medical Informatics and Decision Making **20** (2020) (cit. on p. 57).

[171] B. W. Matthews, *Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme*, Biochimica et Biophysica Acta (BBA)-Protein Structure **405** (1975) 442 (cit. on p. 60).

[172] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, *Fast unfolding of communities in large networks*, Journal of Statistical Mechanics: Theory and Experiment **2008** (2008) (cit. on p. 60).

[173] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017 2999 (cit. on pp. 64, 66).

[174] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, arXiv preprint arXiv:1910.01108 (2019) (cit. on p. 66).

[175] X. Zhou and R. Zafarani, *A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities*, ACM Computing Surveys (CSUR) **53** (2020) 1 (cit. on p. 74).

[176] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, *Defending Against Neural Fake News*, Advances in neural information processing systems **32** (2019) (cit. on p. 74).

[177] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, *Fake News Detection on Social Media using Geometric Deep Learning*, CoRR (2019) (cit. on p. 74).

[178] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, *Fake News Detection on Social Media: A Data Mining Perspective*, CoRR **abs/1708.01967** (2017) (cit. on p. 74).

[179] Á. Figueira and L. Oliveira, *The Current State of Fake News: Challenges and Opportunities*, Procedia computer science **121** (2017) 817 (cit. on p. 74).

[180] M. Visentin, G. Pizzi, and M. Pichierri, *Fake News, Real Problems for Brands: The Impact of Content Truthfulness and Source Credibility on consumers' Behavioral Intentions toward the Advertised Brands*, Journal of Interactive Marketing **45** (2019) 99 (cit. on p. 74).

[181] Y. Cheng and Z. F. Chen, *The Influence of Presumed Fake News Influence: Examining Public Support for Corporate Corrective Response, Media Literacy Interventions, and Governmental Regulation*, Mass Communication and Society **23** (2020) 705 (cit. on p. 74).

[182] S. McKay and C. Tenove, *Disinformation as a Threat to Deliberative Democracy*, Political Research Quarterly (2020) 1065912920938143 (cit. on p. 74).

[183] D. P. Redlawsk, A. J. W. Civettini, and K. M. Emmerson, *The Affective Tipping Point: Do Motivated Reasoners Ever "Get It"?* Political Psychology **31** (2010) 563 (cit. on p. 74).

[184] B. Nyhan and J. Reifler, *When Corrections Fail: The Persistence of Political Misperceptions*, Political Behavior **32** (2010) 303 (cit. on p. 74).

[185] B. Swire, U. Ecker, and S. Lewandowsky, *The role of Familiarity in Correcting Inaccurate Information*, English, Journal of Experimental Psychology: Learning, Memory, and Cognition **43** (2017) 1948 (cit. on p. 74).

[186] A. J. Berinsky, *Rumors and Health Care Reform: Experiments in Political Misinformation*, British Journal of Political Science **47** (2017) 241 (cit. on p. 74).

[187] M. Osmundsen, A. Bor, P. B. Vahlstrup, A. Bechmann, and M. B. Petersen, *Partisan Polarization Is the Primary Psychological Motivation behind Political Fake News Sharing on Twitter*, American Political Science Review **115** (2021) 999 (cit. on p. 75).

[188] B. D. Horne, J. Nørregaard, and S. Adali, *Robust Fake News Detection Over Time and Attack*, ACM Trans. Intell. Syst. Technol. **11** (2019) (cit. on p. 75).

[189] S. Kwon, M. Cha, and K. Jung, *Rumor Detection over Varying Time Windows*, PLOS ONE **12** (2017) 1 (cit. on p. 75).

[190] T. R. Reddy, B. V. Vardhan, and P. V. Reddy, *A survey on authorship profiling techniques*, International Journal of Applied Engineering Research **11** (2016) 3092 (cit. on p. 76).

[191] C. Cai, L. Li, and D. Zengi, "Behavior Enhanced Deep Bot Detection in Social Media," *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2017 128 (cit. on p. 76).

[192] S. Hurtado, P. Ray, and R. Marculescu, "Bot Detection in Reddit Political Discussion," *Proceedings of the Fourth International Workshop on Social Sensing*, Association for Computing Machinery, 2019 30 (cit. on p. 76).

[193] D. Kosmajac and V. Keselj, "Twitter User Profiling: Bot and Gender Identification - Notebook for PAN at CLEF 2019," *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22-25, 2020, Proceedings*, Springer, 2020 141 (cit. on p. 76).

[194] W. Daelemans, M. Kestemont, E. Manjavacas, M. Potthast, F. Rangel Pardo, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Tschuggnall, M. Wiegmann, and E. Zangerle, "Overview of PAN 2019: Bots and Gender Profiling, Celebrity Profiling, Cross-Domain Authorship Attribution and Style Change Detection," 2019 402 (cit. on p. 76).

[195] M. Glenski, T. Weninger, and S. Volkova,
Identifying and Understanding User Reactions to Deceptive and Trusted Social News Sources, CoRR **abs/1805.12032** (2018) (cit. on p. 76).

[196] N. Vo and K. Lee,
"Learning from Fact-Checkers: Analysis and Generation of Fact-Checking Language," *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019 335 (cit. on p. 76).

[197] K. Shu, S. Wang, and H. Liu,
"Understanding User Profiles on Social Media for Fake News Detection," *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018 430 (cit. on p. 76).

[198] F. M. Rangel, A. Giachanou, B. Ghanem, and P. Rosso, "Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter," *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, vol. 2696, CEUR Workshop Proceedings, CEUR-WS.org, 2020 (cit. on pp. 76, 98).

[199] A. Giachanou, E. A. Ríssola, B. Ghanem, F. Crestani, and P. Rosso, "The Role of Personality and Linguistic Patterns in Discriminating Between Fake News Spreaders and Fact Checkers," *Natural Language Processing and Information Systems*, Springer International Publishing, 2020 181 (cit. on pp. 76, 77).

[200] Y. Mu and N. Aletras,
Identifying Twitter users who repost unreliable news sources with linguistic information, PeerJ Comput. Sci. **6** (2020) e325 (cit. on p. 76).

[201] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu,
A Comprehensive Survey on Graph Neural Networks,
IEEE Transactions on Neural Networks and Learning Systems (2021) (cit. on pp. 76, 77, 89).

[202] R. Sawhney, H. Joshi, R. R. Shah, and L. Flek, "Suicide Ideation Detection via Social and Temporal User Representations using Hyperbolic Learning," *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021 2176 (cit. on p. 76).

[203] E. Kacupaj, J. Plepi, K. Singh, H. Thakkar, J. Lehmann, and M. Maleshkova, "Conversational Question Answering over Knowledge Graphs with Transformer and Graph Attention Networks," *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021 850 (cit. on p. 76).

[204] Y. Ren and J. Zhang, *HGAT: Hierarchical Graph Attention Network for Fake News Detection*, ArXiv (2020) (cit. on p. 76).

[205] A. Bahns, C. Crandall, O. Gillath, and K. Preacher, *Similarity in Relationships as Niche Construction: Choice, Stability, and Influence Within Dyads in a Free Choice Environment*, Journal of Personality and Social Psychology **11** (2017) 329 (cit. on pp. 76, 77, 89, 93).

[206] Y. Tausczik and J. Pennebaker,
The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods, Journal of Language and Social Psychology **29** (2010) 24 (cit. on pp. 76, 88).

[207] C. Girlea, R. Girju, and E. Amir, "Psycholinguistic Features for Deceptive Role Detection in Werewolf," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016 417 (cit. on p. 76).

[208] X. Zhou, A. Jain, V. V. Phoha, and R. Zafarani, *Fake News Early Detection*, Digital Threats: Research and Practice (2020) 1 (cit. on p. 76).

[209] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic Detection of Fake News," *COLING*, 2018 (cit. on p. 76).

[210] R. Cervero, P. Rosso, and G. Pasi, "Profiling Fake News Spreaders: Personality and Visual Information Matter," *Proc. 26th Int. Conf. on Applications of Natural Language to Information Systems, NLDB-2021*, 2021 355 (cit. on pp. 76, 77).

[211] L. Qiu, H. Lin, J. E. Ramsay, and F. Yang, *You are what you tweet: Personality expression and perception on Twitter*, Journal of Research in Personality **46** (2012) 710 (cit. on p. 77).

[212] İ. Ergu, Z. Işik, and İ. Yankayış, *Predicting Personality with Twitter Data and Machine Learning Models*, 2019 Innovations in Intelligent Systems and Applications Conference (ASYU) (2019) 1 (cit. on p. 77).

[213] K. N. P. Kumar and M. L. Gavrilova, *Personality Traits Classification on Twitter*, 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (2019) 1 (cit. on p. 77).

[214] M. Gjurković and J. Šnajder, "Reddit: A Gold Mine for Personality Prediction," *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, 2018 87 (cit. on p. 77).

[215] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, *Temporal Graph Networks for Deep Learning on Dynamic Graphs*, 2020 (cit. on p. 77).

[216] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured Sequence Modeling with Graph Convolutional Recurrent Networks," *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*, Springer, 2018 362 (cit. on p. 77).

[217] W. Han, Y. Miao, K. Li, M. Wu, F. Yang, L. Zhou, V. Prabhakaran, W. Chen, and E. Chen, "Chronos: A Graph Engine for Temporal Graph Analysis," *Proceedings of the Ninth European Conference on Computer Systems*, 2014 (cit. on p. 77).

[218] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, *Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting*, Proceedings of the AAAI Conference on Artificial Intelligence (2019) 922 (cit. on p. 77).

[219] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," *International Conference on Learning Representations (ICLR '18)*, 2018 (cit. on p. 77).

[220] S. Yan, Y. Xiong, and D. Lin,
Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition, 2018 (cit. on p. 77).

[221] R. Sawhney, S. Agarwal, A. Wadhwa, and R. R. Shah, "Deep Attentive Learning for Stock Movement Prediction From Social Media Text and Company Correlations," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020 8415 (cit. on p. 77).

[222] M. R. Jamnik and D. J. Lane,
The Use of Reddit as an Inexpensive Source for High-Quality Data,
Practical Assessment, Research and Evaluation **22** (2017) 5 (cit. on p. 77).

[223] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, and P. Nakov,
"Predicting Factuality of Reporting and Bias of News Media Sources,"
Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018 3528 (cit. on p. 79).

[224] mediabiasfactcheck.com, *Methodology*, 2021, (visited on 08/10/2021) (cit. on p. 79).

[225] Columbia Journalism Review, *CJR index of fake-news, clickbait, and hate sites*, 2021, (visited on 08/10/2021) (cit. on p. 79).

[226] A. Shrestha and F. Spezzano, "Online Misinformation: From the Deceiver to the Victim," *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2019 847 (cit. on p. 82).

[227] R. K. Garrett and R. M. Bond, *Conservatives' Susceptibility to Political Misperceptions*, Science Advances (2021) (cit. on p. 82).

[228] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal,
"FEVER: a Large-scale Dataset for Fact Extraction and VERification,"
Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018 809 (cit. on p. 87).

[229] E. Agirre, M. Diab, D. Cer, and A. Gonzalez-Agirre,
"SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity," *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Montréal, QC, Canada*, 2012 385 (cit. on p. 88).

[230] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation,"
Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, 2017 (cit. on pp. 88, 98).

[231] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli,
"SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment,"
Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), 2014 1 (cit. on p. 88).

[232] J. H. Lee, J. C. Collados, L. E. Anke, and S. Schockaert, "Capturing Word Order in Averaging Based Sentence Embeddings," *ECAI*, 2020 (cit. on p. 88).

[233] O. P. John, S. Srivastava, et al., *The Big-Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives*, Handbook of personality: Theory and research (2nd ed., pp. 102–138). Guilford Press. (1999) (cit. on p. 88).

[234] Y. Neuman and Y. Cohen, *A Vectorial Semantics Approach to Personality Assessment*, Scientific reports **4** (2014) 1 (cit. on p. 88).

[235] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking," Association for Computational Linguistics, 2017 2931 (cit. on p. 90).

[236] K. N. Jordan, J. Sterling, J. W. Pennebaker, and R. L. Boyd, *Examining long-term trends in politics and culture through language of political leaders and cultural institutions*, Proceedings of the National Academy of Sciences (2019) (cit. on p. 91).

[237] E. Buckels, P. Trapnell, T. Andjelovic, and D. Paulhus, *Internet Trolling and Everyday Sadism: Parallel Effects on Pain Perception and Moral Judgment*, Journal of Personality **87** (2018) (cit. on p. 91).

[238] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, *Gene selection for cancer classification using support vector machines*, Machine learning **46** (2002) 389 (cit. on p. 92).

[239] W. Chen, W. Fang, G. Hu, and M. W. Mahoney, *On the Hyperbolicity of Small-World and Treelike Random Graphs*, Internet Mathematics **9** (2013) 434 (cit. on p. 97).

[240] I. Chami, Z. Ying, C. Ré, and J. Leskovec, *Hyperbolic Graph Convolutional Neural Networks*, Advances in neural information processing systems **32** (2019) (cit. on p. 97).

[241] D. Bahdanau, K. Cho, and Y. Bengio, *Neural Machine Translation by Jointly Learning to Align and Translate*, arXiv preprint arXiv:1409.0473 (2014) (cit. on p. 97).

[242] I. Vogel and M. Meghana, *Fake News Spreader Detection on Twitter using Character N-Grams*, 2020 (cit. on p. 98).

[243] J. Buda and F. Bolonyai, "An Ensemble Model Using N-grams and Statistical Features to Identify Fake News Spreaders on Twitter.," *CLEF (Working Notes)*, 2020 (cit. on p. 98).

[244] E. Fersini, J. Armanini, and M. D'Intorni, "Profiling Fake News Spreaders: Stylometry, Personality, Emotions and Embeddings," *CLEF*, 2020 (cit. on p. 98).

[245] S. M. Mohammad and P. D. Turney, *Crowdsourcing a Word–Emotion Association Lexicon*, Computational intelligence **29** (2013) 436 (cit. on p. 98).

[246] W. Chen, W. Fang, G. Hu, and M. W. Mahoney, *On the hyperbolicity of small-world and treelike random graphs*, Internet Mathematics **9** (2013) 434 (cit. on p. 102).

[247] S. Aparicio, J. Villazón-Terrazas, and G. Álvarez, *A Model for Scale-Free Networks: Application to Twitter*, Entropy **17** (2015) 5848 (cit. on p. 102).

[248] S. M. Carlson, M. A. Koenig, and M. B. Harms, *Theory of mind*, Wiley Interdisciplinary Reviews: Cognitive Science **4** (2013) 391 (cit. on p. 108).

[249] C. Langley, B. I. Cirstea, F. Cuzzolin, and B. J. Sahakian, *Theory of mind and preference learning at the interface of cognitive science, neuroscience, and AI: A review*, Frontiers in Artificial Intelligence **5** (2022) 62 (cit. on p. 108).

[250] H. M. Wellman and D. Liu, *Scaling of theory-of-mind tasks*, Child development **75** (2004) 523 (cit. on p. 108).

[251] H. M. Wellman, *Theory of mind: The state of the art*, European Journal of Developmental Psychology **15** (2018) 728 (cit. on pp. 108, 109).

[252] J. W. Astington and J. A. Baird, *Why language matters for theory of mind*, Oxford University Press, 2005 (cit. on p. 108).

[253] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., *On the opportunities and risks of foundation models*, arXiv preprint arXiv:2108.07258 (2021) (cit. on p. 108).

[254] Z. Ma, Y. Mei, and Z. Su, "Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support," *AMIA Annual Symposium Proceedings*, vol. 2023, American Medical Informatics Association, 2023 1105 (cit. on p. 108).

[255] A. Lahnala, C. Welch, D. Jurgens, and L. Flek, "A Critical Reflection and Forward Perspective on Empathy and Natural Language Processing," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2022 (cit. on pp. 108, 130).

[256] L. Li, Y. Zhang, and L. Chen, "Personalized Transformer for Explainable Recommendation," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021 4947 (cit. on p. 108).

[257] T. Sorensen, J. Moore, J. Fisher, M. Gordon, N. Mireshghallah, C. M. Rytting, A. Ye, L. Jiang, X. Lu, N. Dziri, et al., *A Roadmap to Pluralistic Alignment*, arXiv preprint arXiv:2402.05070 (2024) (cit. on p. 108).

[258] S. Dudy, S. Bedrick, and B. Webber, "Refocusing on Relevance: Personalization in NLG," *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, vol. 2021, NIH Public Access, 2021 5190 (cit. on pp. 108, 110).

[259] A. Madotto, Z. Lin, C.-S. Wu, and P. Fung, "Personalizing Dialogue Agents via Meta-Learning," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019 5454 (cit. on pp. 109, 110).

[260] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi, "Social IQa: Commonsense Reasoning about Social Interactions," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019 4463 (cit. on p. 109).

[261] N. Shapira, M. Levy, S. H. Alavi, X. Zhou, Y. Choi, Y. Goldberg, M. Sap, and V. Shwartz, *Clever hans or neural theory of mind? stress testing social reasoning in large language models*, arXiv preprint arXiv:2305.14763 (2023) (cit. on p. 109).

[262] S. Trott, C. Jones, T. Chang, J. Michaelov, and B. Bergen, *Do Large Language Models know what humans know?* Cognitive Science **47** (2023) e13309 (cit. on p. 109).

[263] A. Nematzadeh, K. Burns, E. Grant, A. Gopnik, and T. Griffiths, "Evaluating Theory of Mind in Question Answering," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018 2392 (cit. on p. 109).

[264] J. W. Astington and D. R. Olson, *The cognitive revolution in children's understanding of mind*, Human development **38** (1995) 179 (cit. on p. 109).

[265] K. Gandhi, J.-P. Fränken, and T. G. N. D. Goodman, *Understanding Social Reasoning in Language Models with Language Models*, arXiv preprint arXiv:2306.15448 (2023) (cit. on p. 109).

[266] Z. Jin, S. Levine, F. Gonzalez Adauto, O. Kamal, M. Sap, M. Sachan, R. Mihalcea, J. Tenenbaum, and B. Schölkopf, *When to make exceptions: Exploring language models as accounts of human moral judgment*, Advances in neural information processing systems **35** (2022) 28458 (cit. on p. 109).

[267] M. Sap, R. Le Bras, D. Fried, and Y. Choi, "Neural Theory-of-Mind? On the Limits of Social Intelligence in Large LMs," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022 3762 (cit. on p. 109).

[268] T. Ullman, *Large language models fail on trivial alterations to theory-of-mind tasks*, arXiv preprint arXiv:2302.08399 (2023) (cit. on p. 109).

[269] C. Hughes and R. T. Devine, *A social perspective on theory of mind*, Handbook of child psychology and developmental science, socioemotional processes **3** (2015) 564 (cit. on p. 109).

[270] N. Soni, M. Matero, N. Balasubramanian, and H. A. Schwartz, "Human Language Modeling," *Findings of the Association for Computational Linguistics: ACL 2022*, 2022 622 (cit. on p. 110).

[271] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan, "A Persona-Based Neural Conversation Model," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016 994 (cit. on pp. 110, 115).

[272] Y. Zheng, G. Chen, M. Huang, S. Liu, and X. Zhu,
Personalized dialogue generation with diversified traits,
arXiv preprint arXiv:1901.09672 (2019) (cit. on pp. 110, 114, 118).

[273] Y. Zheng, R. Zhang, M. Huang, and X. Mao,
"A pre-training based personalized dialogue generation model with persona-sparse data,"
Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 05, 2020 9693
(cit. on p. 110).

[274] H. Song, W.-N. Zhang, Y. Cui, D. Wang, and T. Liu,
Exploiting persona information for diverse generation of conversational responses,
arXiv preprint arXiv:1905.12188 (2019) (cit. on pp. 110, 116).

[275] Z. Ma, Z. Dou, Y. Zhu, H. Zhong, and J.-R. Wen,
"One Chatbot Per Person: Creating Personalized Chatbots Based on Implicit User Profiles,"
Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, Association for Computing Machinery, 2021 555
(cit. on p. 110).

[276] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston,
"Personalizing Dialogue Agents: I have a dog, do you have pets too?" *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,
2018 2204 (cit. on pp. 110, 111).

[277] E. Dinan, V. Logacheva, V. Malykh, A. Miller, K. Shuster, J. Urbanek, D. Kiela, A. Szlam, I. Serban, R. Lowe, et al., "The Second Conversational Intelligence Challenge (convai2),"
The NeurIPS'18 Competition: From Machine Learning to Intelligent Conversations,
Springer, 2020 187 (cit. on pp. 110, 111).

[278] B. P. Majumder, S. Li, J. Ni, and J. McAuley,
"Generating Personalized Recipes from Historical User Preferences," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019 5976
(cit. on p. 110).

[279] S. Vincent, R. Sumner, A. Dowek, C. Blundell, E. Preston, C. Bayliss, C. Oakley, and C. Scarton,
Personalised Language Modelling of Screen Characters Using Rich Metadata Annotations,
arXiv preprint arXiv:2303.16618 (2023) (cit. on p. 110).

[280] P.-E. Mazaré, S. Humeau, M. Raison, and A. Bordes,
"Training Millions of Personalized Dialogue Agents,"
Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,
2018 2775 (cit. on p. 110).

[281] Q. Huang, Y. Zhang, T. Ko, X. Liu, B. Wu, W. Wang, and H. Tang,
"Personalized dialogue generation with persona-adaptive attention,"
Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 11, 2023 12916
(cit. on pp. 113, 115–118).

[282] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, arXiv preprint arXiv:1412.3555 (2014) (cit. on p. 114).

[283] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, "Emotional chatting machine: Emotional conversation generation with internal and external memory," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 1, 2018 (cit. on p. 114).

[284] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., *Llama 2: Open foundation and fine-tuned chat models*, arXiv preprint arXiv:2307.09288 (2023) (cit. on p. 116).

[285] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, *Lora: Low-rank adaptation of large language models*, arXiv preprint arXiv:2106.09685 (2021) (cit. on p. 116).

[286] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002 311 (cit. on p. 116).

[287] C.-Y. Lin and F. J. Och, "Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics," *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 2004 605 (cit. on p. 116).

[288] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A Diversity-Promoting Objective Function for Neural Conversation Models," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016 110 (cit. on p. 116).

[289] Y. Tang, B. Wang, M. Fang, D. Zhao, K. Huang, R. He, and Y. Hou, *Enhancing Personalized Dialogue Generation with Contrastive Latent Variables: Combining Sparse and Dense Persona*, arXiv preprint arXiv:2305.11482 (2023) (cit. on p. 116).

[290] N. Heilig, J. Kirchhoff, F. Stumpe, J. Plepi, L. Flek, and H. Paulheim, *Refining diagnosis paths for medical diagnosis based on an augmented knowledge graph*, arXiv preprint arXiv:2204.13329 (2022) (cit. on p. 130).

[291] Y. Ma, K. L. Nguyen, F. Z. Xing, and E. Cambria, *A survey on empathetic dialogue systems*, Information Fusion **64** (2020) 50 (cit. on p. 130).

[292] V. Sadiri Javadi, M. Potthast, and L. Flek, "OpinionConv: Conversational Product Search with Grounded Opinions," *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2023 66 (cit. on p. 130).

[293] M. Sap, S. Gabriel, L. Qin, D. Jurafsky, N. A. Smith, and Y. Choi, "Social Bias Frames: Reasoning about Social and Power Implications of Language," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020 5477 (cit. on p. 130).

[294] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. S. Liang, and J. Leskovec, *Deep bidirectional language-knowledge graph pretraining*, Advances in Neural Information Processing Systems **35** (2022) 37309 (cit. on p. 130).

[295] D. Yu, C. Zhu, Y. Fang, W. Yu, S. Wang, Y. Xu, X. Ren, Y. Yang, and M. Zeng, "KG-FiD: Infusing Knowledge Graph in Fusion-in-Decoder for Open-Domain Question Answering," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022 4961 (cit. on p. 130).

[296] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal, "The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task," *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, 2021 1 (cit. on p. 130).

[297] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020 6769 (cit. on p. 130).

[298] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, *Survey of hallucination in natural language generation*, ACM Computing Surveys **55** (2023) 1 (cit. on p. 131).

[299] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering," *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021 535 (cit. on p. 131).

[300] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C. Manning, and J. Leskovec, "GreaseLM: Graph REASoning Enhanced Language Models for Question Answering," *International Conference on Representation Learning (ICLR)*, 2022 (cit. on p. 131).

[301] C. Hewson and T. Buchanan, "Ethics guidelines for internet-mediated research," The British Psychological Society, 2013 (cit. on p. 131).

[302] M. L. Williams, P. Burnap, and L. Sloan, *Towards an ethical framework for publishing Twitter data in social research: Taking into account users' views, online context and algorithmic estimation*, Sociology **51** (2017) 1149 (cit. on p. 131).

[303] K. Shilton and S. Sayles, "" We Aren't All Going to Be on the Same Page about Ethics": Ethical Practices and Challenges in Research on Digital and Social Media," *2016 49th Hawaii International Conference on System Sciences (HICSS)*, IEEE, 2016 1909 (cit. on p. 131).

[304] L. Townsend and C. Wallace, *Social media research: A guide to ethics*, University of Aberdeen **1** (2016) 16 (cit. on p. 131).

[305] C. Norval and T. Henderson, *Contextual Consent: Ethical Mining of Social Media for Health Research*, CoRR **abs/1701.07765** (2017) (cit. on p. 131).

[306] D. S. Shah, H. A. Schwartz, and D. Hovy, “Predictive Biases in Natural Language Processing Models: A Conceptual Framework and Overview,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020 5248 (cit. on p. 131).

[307] L. A. Rudman and P. Glick, *The social psychology of gender: How power and intimacy shape gender relations.*, Guilford Press (2008) (cit. on p. 131).

[308] C. Koolen and A. van Cranenburgh, “These are not the Stereotypes You are Looking For: Bias and Fairness in Authorial Gender Attribution,” *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, 2017 12 (cit. on p. 131).

[309] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach, “Language (Technology) is Power: A Critical Survey of “Bias” in NLP,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020 5454 (cit. on p. 131).

List of Publications

- ***Conference Papers (peer reviewed)***

  1. **Joan Plepi** and Lucie Flek. 2021. "Perceived and Intended Sarcasm Detection with Graph Attention Networks." In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4746–4753, Punta Cana, Dominican Republic. Association for Computational Linguistics. DOI: 10.18653/v1/2021.findings-emnlp.408
  2. **Joan Plepi**, Flora Sakketou, Riccardo Cervero, Henri Jacques Geiss, Paolo Rosso, and Lucie Flek. 2022. FACTOID: A New Dataset for Identifying Misinformation Spreaders and Political Bias. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 3231–3241, Marseille, France. European Language Resources Association.
  3. **Joan Plepi**, Flora Sakketou, Henri-Jacques Geiss, and Lucie Flek. 2022. Temporal Graph Analysis of Misinformation Spreaders in Social Media. In Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, pages 89–104, Gyeongju, Republic of Korea. Association for Computational Linguistics.
  4. **Joan Plepi**, Béla Neuendorf, Lucie Flek, and Charles Welch. 2022. Unifying Data Perspectivism and Personalization: An Application to Social Norms. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7391–7402, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. DOI: 10.18653/v1/2022.emnlp-main.500
  5. **Joan Plepi**, Charles Welch, and Lucie Flek. 2024. Perspective Taking through Generating Responses to Conflict Situations. In Findings of the Association for Computational Linguistics ACL 2024, pages 6482–6497, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics. DOI: 10.18653/v1/2024.findings-acl.387

- ***Workshop Articles (peer reviewed)***

  6. Charles Welch, **Joan Plepi**, Béla Neuendorf, and Lucie Flek. 2022. Understanding Interpersonal Conflict Types and their Impact on Perception Classification. In Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS), pages 79–88, Abu Dhabi, UAE. Association for Computational Linguistics. 10.18653/v1/2022.nlpcss-1.10
  7. **Joan Plepi**, Magdalena Buski, and Lucie Flek. 2023. Personalized Intended and Perceived Sarcasm Detection on Twitter. In Proceedings of the 3rd Workshop on Computational Linguistics for the Political and Social Sciences, pages 8–18, Ingolstadt, Germany. Association for Computational Lingustics.

- ***Miscellaneous Papers (peer reviewed)***

  Following publication originated during the thesis, but is not part of the thesis itself.

  8. Heilig, Niclas, Jan Kirchhoff, Florian Stumpe, **Joan Plepi**, Lucie Flek, and Heiko Paulheim. Refining diagnosis paths for medical diagnosis based on an augmented knowledge graph. CEUR Workshop Proceedings 2022.

List of Figures

List of Tables