Research Master Thesis

Multi-task fine-tuning for hate speech detection

Dorien Renting

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Human Language Technology)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by:	dr. Ilia Markov
2nd reader:	dr. Pia Sommerauer
Submitted:	June 30, 2023

Abstract

Hate speech is becoming increasingly prevalent online. It is important to develop tools that can accurately detect hate speech in social media messages. This thesis aims to improve the detection of explicit and implicit hate speech by means of multi-task fine-tuning of BERT. Multi-task learning is approached in two ways. First, we aim to improve the detection of hate speech by leveraging information about the sentiment and emotions expressed in the text, as well as information about whether the text is sarcastic or ironic. This is done by modelling four tasks (sentiment analysis, emotion detection, sarcasm detection and irony detection) alongside hate speech detection in a shared BERT encoder. Experiments are conducted on three different hate speech datasets. The results of the experiments suggest that the tasks can be helpful for hate speech detection, both explicit and implicit instances. A manual analysis reveals that especially sentiment and emotion information aided the detection of hate speech. Sarcasm and irony were more difficult to learn accurately, and the knowledge transfer is less strong. Secondly, we model the three different hate speech datasets as multiple tasks. Because the BERT model can learn from all three datasets at the same time, we hypothesize that this model is better at predicting hate speech than a model trained on one dataset. This hypothesis was not confirmed. The multi-task model only improved the performance on one of the three datasets. The code used for this thesis can be found at `https://github.com/drenting/VU-thesis-2023/tree/main`

Declaration of Authorship

I, Dorien Renting, declare that this thesis, titled *Multi-task fine-tuning for hate speech detection* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 30-06-2023

Signed:

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Social media platforms have grown immensely over recent years. They are used to connect to other users, but also to express opinions and share information. The growth of these platforms is accompanied by a concerning trend: the expression of hate speech. Hate speech is often defined as language that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic (Nockleby, 2000). It is important to surveil the language used online because of several reasons. Online safety is the biggest consideration. Cyberbullying, racism and other offensive language can make people feel unsafe and can have an adverse effect on mental health and even lead to suicide (Bauman et al., 2013; Saha et al., 2019; Ștefăniță and Buf, 2021; Wachs et al., 2022). The expression of hate speech online has been linked to real-life violence. Facebook, for example, has been blamed for playing a role in the deadly hate crimes in Sri Lanka (Safi, 2018) and Myanmar (Stecklow, 2018) by allowing hateful ideas to spread among their users. As such, it is essential that social media platforms take measures to flag and/or remove harmful content to protect their users from harm and nip violence in the bud.

Aside from the offline consequences of online hate speech, the expression of hate speech might in itself be criminal. While it is generally protected as free speech under the First Amendment in the United States, the European Union has made hate speech a criminal offense.[1] The EU also has been working on specifically combating online hate speech in their Digital Services Act, which aims to go against online hate speech and other harmful content.[2]

Given these concerns, social media platforms have been urged to take action to regulate the hate speech expressed on their websites. However, due to the sheer volume of content that is generated everyday, it is impossible to manually check for every post whether it violates platform guidelines or expresses hateful language. Therefore, the automatic detection of hate speech has become a major research area in computer science and natural language processing (NLP). While much research has been conducted, there are a number of challenges that make the automatic detection of hate speech difficult.

Some of the main challenges in hate speech detection are the definition (MacAvaney

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32008F0913

[2] https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/digital-services-act-ensuring-safe-and-accountable-online-environment_en

et al., 2019), the implicitness (Vidgen et al., 2019), and the distinction between mere profanity and hate speech Malmasi and Zampieri (2018). Hate speech is a subjective phenomenon, and there is no consensus on how to define it. There are many different types of offensive language, some of which are racist, while others are sexist or homophobic. This lack of clarity makes it difficult to develop generalizable algorithms that can accurately identify and categorize hate speech (MacAvaney et al., 2019; Vidgen et al., 2019), especially on new targets that were not present in the training data (Waseem et al., 2018). For example, certain terms might be used to insult or discriminate against women, but other words might be used to discriminate against Muslims. It is important to make general systems that can deal with many types and targets of hate speech (Waseem et al., 2018). Secondly, hate speech can be implicit and hidden in figurative language which makes it not immediately obvious. This can include the use of sarcasm, irony, metaphor, and other linguistic devices that make it difficult to identify hate speech (Malmasi and Zampieri, 2018; Vidgen et al., 2019). Explicit hate speech on the other hand is immediately obvious from the surface of the utterance. The last challenge addressed in this thesis is distinguishing between hate speech and profanity. Even though there is not one general definition of hate speech, many definitions have in common that it is only hate speech when specific groups of people are attacked (Fortuna and Nunes, 2018). There us thus a difference between the use of inappropriate words (profanity) and the use of such words to attack a protected group of people (hate speech). However, systems often rely on profanity to identify hate speech, as profanity is often observed in hate speech Malmasi and Zampieri (2018). There are other challenges, such as the importance of the context words are used in and the identity of the speaker (Vidgen et al., 2019), but this falls outside of the scope of this thesis.

In short, the automatic detection of hate speech on social media platforms is a complex and challenging problem. Researchers must grapple with legal and ethical considerations, as well as the sheer volume of content generated on these platforms. They must develop scalable and generalizable algorithms that can accurately identify hate speech, even when it is implicit or targets an unseen group. While progress has been made in this area, much work remains to be done to ensure the safety and well-being of users on social media platforms.

1.1 Approach and research questions

In this thesis, we propose to use multi-task learning. The aim of this thesis is two-fold: improve the identification of different types of hate speech, i.e. implicit and explicit hate speech, and improve generalizability. Multi-task learning is one approach to inductive transfer where the knowledge of one task is used for another task based on some commonality. This is done by constructing multiple classifiers that have shared layers, allowing the model to learn more generalized representations from different, but related, tasks (Caruana, 1997).

Hate speech is related to sarcasm and irony (Malmasi and Zampieri, 2018), negative emotions (Martins et al., 2018) and negative sentiment (Plaza-del Arco et al., 2021). Hate speech detection is thus related to sarcasm and irony detection, emotion recognition and sentiment analysis.

Sentiment analysis is a task aimed at determining the overall sentiment (or polarity) of a text, usually categorized as positive, negative, or neutral. Emotion detection aims at determining what emotion is expressed in a text. Hate speech is inherently negative and linked to negative emotions (Martins et al., 2018; Plaza-del Arco et al., 2021). Knowing whether a message carries such negative connotations might therefor be helpful in identifying hate speech. It might also help distinguish between the use of profanity and hate speech, as utterances with profanity but an overall neutral or positive sentiment are most likely not considered hate speech (Plaza-Del-Arco et al., 2021). Incorporating emotion and sentiment information might thus especially make the detection of explicit hate speech more precise.

Sarcasm detection and irony detection involve identifying text that contains sarcastic or ironic statements. This can be challenging as such figurative language often involves the use of language that appears positive but conveys a negative sentiment (Ghosh et al., 2020; Van Hee et al., 2018). It was established above that implicit hate speech might be so difficult to identify because it is sarcastic or ironic. Having information about sarcasm and irony might aid a hate speech detection system in correctly identifying implicit hate speech. These four tasks are implemented as auxiliary tasks in multi-task learning classifiers. They serve as supporting tasks to improve hate speech detection, which is the primary task.

The second aim of this thesis is to improve generalizability. This is done, not by learning from multiple different tasks, but by using different hate speech datasets and treating them as different tasks. Because different targets of hate speech are insulted by different words, the generalizability of a system depends on what data it was trained on. A model trained on racist speech might not the good at identifying sexist speech. Moreover, because there is no consensus on what hate speech is, most datasets have been annotated according to different guidelines and definitions. To solve both these issues, Waseem et al. (2018) proposed to train a model on multiple datasets covering both racist and sexist discourse. In this thesis, the same approach is taken: three different datasets annotated for hate speech, all with different distributions, are learned at the same time.

The three datasets are AbuseEval (Caselli et al., 2020), TRAC (Kumar et al., 2018) and Implicit Hate Corpus (IHC) (ElSherief et al., 2021). The discourse in AbuseEval evolves mainly around American politics. TRAC is comprised of data from India and the Implicit Hate Corpus is data from different hate groups, mainly focused on ethnicity and religion. The data is thus very diverse, and a model trained on just one of these datasets might not capture the full range of possible hate speech targets. The three datasets have all been annotated for different phenomena. AbuseEval captures abusive language, TRAC aggression and IHC hateful language. All of these phenomena can all be seen as types of hate speech. Moreover, these three datasets all make the distinction between explicit and implicit hate speech. They can thus be used to test the effect of different related tasks on the detection of these two types of hate speech. The transformer-based model BERT (Devlin et al., 2019) has been shown to be effective when fine-tuned for multiple tasks at once in Spanish and English (Plaza-Del-Arco et al., 2021; Plaza-del Arco et al., 2021).

The research questions addressed in this thesis are as follows:

1. What is the effect of multi-task fine-tuning on automatic hate speech detection?
   (a) What is the effect of different auxiliary tasks on different types of hate speech?
   (b) What is the effect of training on multiple datasets for hate speech at the same time?

In this thesis, we explore the potential of multi-task fine-tuning BERT using sentiment analysis, emotion detection, sarcasm detection, and irony detection to answer the first question. The second question will be answered by implementing the three datasets annotated for hate speech in a multi-task fine-tuning setup. The main findings of this thesis suggest that multi-task learning is helpful for hate speech detection. However, the results are mixed. Irony detection is especially helpful for implicit hate speech detection. The other tasks have different effects on each of the three datasets, improving the detection of implicit hate speech on one and improving explicit hate speech on another, for example. Learning all three datasets in a multi-task setup only improves performance on one of the three datasets.

1.2 Outline

The remainder of this thesis is structured as follows. Chapter 2 provides an overview of the relevant literature on automatic hate speech detection and how multi-task fine-tuning can be utilized for this. Chapter 3 describes the data used in this thesis, and thus defines the terminology used in the rest of the thesis. Chapter 4 is concerned with the architecture of the multi-task fine-tuning model and the experiments that have been conducted with this model architecture. The results of the experiments are presented in Chapter 5. Chapter 6 provides a manual analysis of some of the models and discusses the findings. Chapter 7 concludes this thesis with a summary of the main findings and suggestions for future work.

NOTE: This thesis contains examples of language that might be offensive to some readers. They do not reflect the views of the author. Slurs have been censored.

Chapter 2

Background and Related Work

This chapter provides an overview of hate speech detection in natural language processing (NLP). First, hate speech is discussed on a conceptual level. What it is and how the concept is used for NLP is outlined. Then, different methods for automatic detection of hate speech are explained. The focus will shift from single task learning to multitask learning. The methods for single task hate speech detection range from traditional machine learning algorithms with handcrafted features (see Section 2.2.1) to end-to-end neural models (Section 2.2.2). Many different features have been found to be useful, as well as many algorithms. Using pre-trained language models such as BERT (Devlin et al., 2019) has also become a popular method. BERT and relevant hate speech papers will be described in Section 2.3. Multi-task learning is another interesting research direction for hate speech detection. Recent work is reviewed in Section 2.4.

2.1 Defining and operationalizing hate speech

Defining hate speech is not straightforward. There are legal definitions, terms of service on social media platforms and definitions researches use in automatic hate speech detection. There is no consensus on what hate speech is, what the terminology means and how to operationalize hate speech for automatic detection models. Hate speech is sometimes used as an umbrella term (in Schmidt and Wiegand (2017) for example) to cover a range of phenomena including abusive, profane and offensive language, cyberbullying and insults. However, more often it is used to refer to a specific type of offensive language, namely language that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic (Nockleby, 2000). The focus on belonging to a certain group is also found in other definitions, for example in Davidson et al. (2017); Waseem and Hovy (2016). Hate speech is then different from cyberbullying, which might target anyone for any reason, not their being part of a (marginalized) group. See Fortuna and Nunes (2018) for an overview of different definitions and related concepts to hate speech such as abusive language, offensive language, etc.

Even when there is a clear definition, applying this to data poses its own problems. Inter-annotator reliability is not always satisfactory and different types of annotators result in different annotations (Schmidt and Wiegand, 2017; Waseem, 2016; Ross et al., 2017). The low inter-annotator agreement is also often due to the lack of context (Waseem and Hovy, 2016).

To make hate speech more concrete and easier to identify, different ways of categorizing it into types have been proposed. By defining more fine-grained labels, more aspects of offensive language can be covered at once, solving some of the terminology issues. Waseem et al. (2017) were the first to take the target of the hate speech and the explicitness of the hate as point of departure. Both have been adopted by researchers. This is usually done in a hierarchical manner. First, all offensive messages are found, then those are further categorized. For example, OLID (Zampieri et al., 2019a) distinguishes targeted and untargeted hate. They further divide the targeted utterances into group targeted and individual targeted messages, thereby covering hate speech (group), cyberbullying (individual) and offensive language (untargeted messages). Similarly, the HASOC dataset (Mandl et al., 2019) has a top level annotation whether a message is hateful, offensive or profane at all, or not. These messages then get further categorized as either hate, offensive or profane and as targeted and untargeted. Another way of distinguishing between different kinds of hate is the explicitness of the hate.

2.1.1 Implicit and explicit hate speech

The explicit/implicit distinction is made in AbuseEval (Caselli et al., 2020), the Implicit Hate Corpus (IHC) (ElSherief et al., 2021) and DALC dataset (Caselli et al., 2021), to name a few. The hate or abuse in an utterance is explicit when there are certain hateful or violent words or phrases used. When this is not the case, but the utterance is still considered hateful, this is an instance of implicit hate. Implicit hate speech is an indirect attack and can be disguised in jokes or figurative language. Examples of explicit hate speech are:

1. i would have no problem putting a bullet into these women's heads. (IHC)
2. i hate n*ggers and f*ggots!!! (IHC) (censor mine)
3. @USER Well she is dumb as a box of rocks so will give her a pass (AbuseEval)

It is clear that these are hate speech because the attack is direct. In the first example, the tweeter refers to killing women, and the second example uses two slurs. The last example is an attack to someone's intelligence ('dumb as a box of rocks'). In contrast, implicit hate speech is more subtle:

1. what a genius, your mother would be proud (IHC)

In this example the user attacks the same quality as example 3 (intelligence) but does so through sarcasm. On the surface, this message only contains words with positive connotations ('genius' and 'proud'), but this message is not meant to be a compliment. Hate speech is thus more complex than just looking for negative or offensive words.

Similar to the explicit and implicit datasets, Kumar et al. (2018) label tweets and Facebook posts as being covertly aggressive, overtly aggressive or neutral for their TRAC dataset. The distinction between overt and covert aggression is similar to explicit and implicit hate speech. This dataset, as well as AbuseEval and the Implicit Hate Corpus will be used in this thesis. Please refer to Chapter 3 for a more detailed overview of these datasets.

Several studies have found that explicit hate speech is found with a higher degree of accuracy than implicit hate speech (Caselli et al., 2020, 2021; Risch et al., 2019; Risch and Krestel, 2020). This is a shortcoming that needs to be addressed. This thesis aims to improve the detection of hate speech by exploiting the similarities between hate speech detection and other NLP tasks in a multi-task fine-tuning approach.

2.2 Automatic hate speech detection

Despite the many difficulties in automatic hate speech detection, many studies have been conducted experimenting with different classifiers and feature representations. The problem is often tackled as a single-task learning objective. The single task paradigm is based on learning a single task, from one dataset. The studies mentioned below thus only focus on classifying hate speech (or related concepts). The methods for hate speech detection range from traditional machine learning algorithms with handcrafted features to end-to-end neural models.

2.2.1 Traditional machine learning

A wide range of different classification methods and features have been explored for automatic hate speech detection. A brief overview of some influential studies will be given. Refer to Schmidt and Wiegand (2017) for a more complete overview.

Davidson et al. (2017) use a Logistic Regression algorithm with lexical features such as word n-grams and semantic features such as a sentiment polarity score to classify tweets as hate speech, offensive language or neither. They find that this methods works to some extent, but that the recall of the hate speech and offensive language classes are not sufficient. Warner and Hirschberg (2012) experiment with Support Vector Machines (SVM) with data from Yahoo and various other websites. In contrast to the previous study, this study is focussed specifically on stereotypes in text, and aims to classify paragraphs as being anti-Semitic, anti-black, anti-Asian, anti-woman, anti-Muslim, anti-immigrant and other hate. They also find that the recall of the hateful classes is not sufficient. Waseem and Hovy (2016) collect tweets and label them as being racist, sexist or neither. They experiment with implementing features capturing gender, location of the user and length of the tweet, in addition to the often used character n-grams. Location and length do not seem to aid the performance of a Logistic Regression classifier, and gender seems to have a small positive effect.

Besides word and character n-grams, features for hate speech detection include linguistic information (e.g. part-of-speech, dependency relations), word generalizations (e.g. word embeddings), lexical resources (e.g.list of slurs), user history and use of emojis (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018). In therms of machine learning algorithms, SVMs are popular and seem to work well (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018).

The last type of approach I want to discuss is the use of more fine-grained information for hate speech detection. The features mentioned above mainly focus on linguistic information. However, semantic features such as emotion, sentiment and metaphor have been found to be useful as well.

As hate speech is intrinsically negative, sentiment analysis might be useful for its detection. Sentiment analysis aims to find the polarity of a text, that is to say, whether it expresses a negative, positive or neutral stance towards some topic. Van Hee et al. (2015) investigate the automatic detection of cyberbullying on Dutch data from the online forum Ask.fm. They include sentiment as four features: the normalized count of negative, positive and neutral words in a text and an average polarity score based on these three counts. This information is based on a sentiment lexicon for Dutch. They found that the four sentiment features were not sufficient for classifying instances of cyberbullying. Combining these features with word unigrams, word bigrams and character trigrams, they could achieve performance that none of these features reached alone. It is not clear what the contribution of the sentiment features was exactly from this study. Other studies that used sentiment analysis as features also combined this with lexical features (Nahar et al., 2014; Yin et al., 2009) and found an increase in performance by doing so.

Hate speech is also linked to the emotional state of the speaker (Patrick, 1901). Emotion analysis has thus been utilized in several studies. The objective of textual emotion detection is identifying what emotion is expressed in a text. Pre-defined sets of emotions such as the six basic emotions (Ekman, 1992) or Plutchik's wheel of emotions (Plutchik, 1980) are often utilized. Markov et al. (2021) found that stylometric and emotional features can help traditional machine learning models identify hate speech in text. These were encoded in the following way. First, the part-of-speech of all the words in the message is found. The POS-tag is replaced by the surface form of the word when the word is a function word (this encodes the style of the author) or when the word conveys emotions according to a lexicon. Two additional emotion features are implemented: the number of emotional words in the message and the emotions the words are associated with. Compared to an SVM with just the POS features, stylometric and emotion features seem to aid the detection of hate speech. In a cross-domain setup, the highest performance was achieved when using an ensemble of the aforementioned SVM with stylometric and emotion features, a CNN classifier and a BERT classifier. Samghabadi et al. (2019) also utilizes emotion information, but takes a different approach. They collect data from the social media website for teenagers Curious Cat. They then use the DeepMoji framework (Felbo et al., 2017) to represent the text. DeepMoji takes text as input and outputs the association of that text with 64 different emojis (and in turn with the underlying emotions in the text). This information is combined with the output of a BiLSTM embedding model and gets fed to a Gated Emotion-Aware Attention model which predicts whether the text contains abusive language or not. The model with the DeepMoji representations performed better than their baselines on their own collected data, but not on other datasets. This research suggests that emotion in text might be useful for hate speech detection. Martins et al. (2018) define hate speech as 'any emotional expression imparting opinions or ideas – bringing a subjective opinion or idea to an external audience- with discriminatory purposes' (p. 61) based on Brown (2017). Emotion is thus a central part of hate speech in this study and is integrated into the experiments by means of features. In addition to the words in the message, twelve emotional features are used for classification: a count of how many words in the text are associated to the eight emotions, a score for how positive, negative and angry the text is and a binary feature whether the text contains hate words. All of this information is obtained by looking up the words in different lexicons. They experiment with Random Forest, Naive Bayes and SVM classifiers and find that the SVM works best. Their method also beats the original experiments on the data by Davidson et al. (2017), indicating that emotional features are informative for hate speech classification.

Implicit instances of hate speech often involve the use of figurative language such as metaphors Caselli et al. (2020). Whether the analysis of metaphors can actually help in detecting hate speech was investigated in Lemmens et al. (2021). They aimed to predict the type and the target of hateful Dutch Facebook comments about migrants and the LGBT community. Hateful metaphors in the messages were annotated manually and used as features for an SVM as tags and counts, in addition to token unigrams and bigrams. BERTje (De Vries et al., 2019) and RobBERT (Delobelle et al., 2020), the Dutch equivalents of BERT and RoBERTa, were also used to predict type and target and tags were used to encode the metaphor information. They found that the metaphor features could improve predicting the type of hate speech using SVM and the transformer based models. For target prediction, the SVM could be improved, but this was not the case for BERTje and RobBERT. The qualitative analysis revealed that the examples where improvement was observed, these are often implicit hate speech. Incorporating information from metaphors is thus a promising approach to hate speech detection and especially for improving the detection of implicit instances.

2.2.2 Deep learning

Recently, there has been a departure from traditional machine learning methods, in favor of deep learning approaches. The first study to apply deep learning techniques to the detection of hate speech is Badjatiya et al. (2017). They use the data gathered by Waseem and Hovy (2016) to train a CNN, an LSTM and FastText embeddings. The embeddings learned by the CNN and LSTM models were also used as input to traditional machine learning algorithms. The best score was attained by a combination of deep and traditional learning: The learned embeddings from the LSTM with random initialization were input to Gradient Boosted Decision Trees. This model significantly beat Waseem and Hovy (2016). Pitsilis et al. (2018) build on this study by using the same architecture and data but using different features. The features are the tendency of the user to tweet sexist, racist or neutral messages (each of these tendencies is its own feature) and a representation of the words in the text. These representations are based on the frequency of the word in the corpus. This model is therefore language-independent. An ensemble of five of these LSTMs with different combinations of the features has a slightly better performance than Badjatiya et al. (2017). However, these features might not be the best choice. User behaviour can change at any moment, and for new users this information is not available. Lastly, Zhang et al. (2018) used a CNN with an additional Gated Recurrent Unit (GRU) network. The input to this model are the word2vec embeddings. The intuition behind the model architecture is the feature extraction ability of CNNs and the ability to model word order information by recurrent networks. They collect their own dataset from Twitter with tweets about Muslims and refugees and label them as ‘hate’ or ‘no hate’. They also make use of six publicly available datasets. Their model architecture outperforms their baselines (SVMs and simpler versions of their CNN+GRU model) on all seven datasets. On all but one dataset, they also beat the state-of-the-art performance.

Deep learning thus seems to be current best way to approach hate speech detection. However, in the 2019 Shared Task of SemEval, an SVM model scored the highest, beating several neural network approaches Basile et al. (2019).

2.3 BERT

Besides training neural networks from scratch, fine-tuning pre-trained language models has also been done for hate speech detection. BERT is such a pre-trained language model, and is the focus of this section. First, I will explain how BERT works and how it can be used for classification. Then I will lay out some of the research that has used BERT for hate speech detection.

2.3.1 BERT model

BERT, which stands for Bidirectional Encoder Representations from Transformers, was introduced in 2019 (Devlin et al., 2019). The main mechanism at work is the transformer (Vaswani et al., 2017). BERT is a pre-trained language model that can create contextual embeddings of text. These embeddings can be used for various NLP tasks. For this, the model needs to be fine-tuned for that specific task.

There are two main versions of the BERT architecture. The base model has twelve transformer blocks and twelve self-attention heads and a hidden size of 768. $BERT_{BASE}$ has a total of 110 million tunable parameters. $BERT_{LARGE}$ has 24 transformer blocks, sixteen self-attention heads and a hidden size of 1024, which results in 340 million tunable parameters.

In its pre-training phase, the model was taught to understand language. This was done in two ways: masked language modelling (MLM) and next sentence prediction (NSP). Both of these approaches are unsupervised learning as the data is unlabelled. For MLM, random words in a sentence are masked, and the model predicts what word should be there. The model sees the whole sentence, and can thus attend to the words to the left and to the right of the masked word. Most other language models, such as GPT-2 (Radford et al., 2019), are unidirectional. They can only capture context to the left of the current word. This does not give a complete representation of a word. BERT is deeply bidirectional, that is, the whole model is bidirectional. In the second pre-training task, NSP, the model learns to predict whether sentence B follows sentence A. For this, the representation of the special [CLS] token is used. The [CLS] token thus represents the whole sentence and can be used for sequence classification. The data used for pre-training is the BookCorpus, which consists of around 800 million words, and English Wikipedia pages (2,500 million words).

The resulting BERT model provides contextual embeddings of tokens. Previous embeddings models are static (e.g. Glove (Pennington et al., 2014) and word2vec [1]). In static word embedding models, each item in the vocabulary has one embedding representation. The meaning of polysemous words can therefore not be accurately captured. BERT can represent the same word in multiple ways depending on the context.

In order to go from raw text to a BERT embedding, one must tokenize the data appropriately first. Single utterances or a pair of utterances can be represented. A special token [CLS] gets added to the beginning of the utterance. The token [SEP] is used at the end of single utterance or between two utterances to separate them. The input to BERT is always a fixed size of maximum 512 tokens. Longer sequences are truncated, and shorter sequences are padded with special token [PAD]. The WordPiece tokenizer (Wu et al., 2016) is used to tokenize the raw text. This splits longer words into pieces where it sees fit. There are cased and uncased versions of BERT. Uncased models convert all characters to lowercase. There is thus no difference between ‘book’, ‘Book’ and ‘BOOK’ . A sentence position embedding (position of each token in the sequence) and a segment embedding (whether the token belongs to the first or second sequence)

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

get added to the WordPiece embeddings. The sum of these three embeddings are fed to the transformer blocks. The output of this are the token embeddings, including the embedding of the [CLS] token.

These embeddings can then be used for classification. For token classification tasks, such as Named Entity Recognition, the embedding for each token is mapped to an output. For sequence classification, i.e. assigning a label to a whole sequence, the [CLS] embedding is used. To use BERT for classification, it needs to be fine-tuned. This is done by adding a classifier on top of BERT. This classifier often takes the form of a fully connected linear layer that outputs class probabilities. However, other classifiers can also be used, such as a classifier with hidden layers or even a CNN Mozafari et al. (2020). In the fine-tuning process, the pre-trained BERT parameters, as well as the newly initialized parameters of the classifier are fitted to the training data. It was found that the lower layers of BERT capture general linguistic information such as POS tags, while the higher layers capture semantic knowledge (Tenney et al., 2019), so one can play around with freezing some of these layers depending on what information is most useful for the downstream task (Sun et al., 2019).

2.3.2 BERT for hate speech detection

Fine-tuning BERT for hate speech detection has become a popular method. All top ranking teams in the shared task for offensive language detection in 2020 utilized a transformer model, i.e. BERT, RoBERTa or XLM-RoBERTa (Zampieri et al., 2020). The best performing teams in the 2021 shared task of toxic spans detection also utilized BERT and RoBERTa (Pavlopoulos et al., 2021), highlighting the potential of these models. Fine-tuning BERT can be done in multiple ways. Mozafari et al. (2020) try four different fine-tuning strategies with the BERT model. First, they use the simplest way of fine-tuning BERT: Putting a simple classifier on top of BERT. This is done by taking the [CLS] token representation and inputting it into a fully connected linear layer that outputs a probability for each class. Secondly, they add to this by replacing the linear classifier by a fully connected network with two hidden layers. Thirdly, instead of a simple neural network, they build a BiLSTM on top of the BERT encoder. This does not only take the [CLS] token, but the full sequence representation. Lastly, they experiment with information from all the layers of the BERT model. They take the representation from three encoder blocks at a time and send those to a CNN. They find that the CNN classifier performs best, and that only the classifier with two hidden layers does not beat previous research.

BERT is also often used in ensembles. Risch et al. (2019), for example, made an ensembles of five German BERT models. They found that the ensembles outperformed the single models. The performance was especially good on explicit offensive language, but they model had a harder time identifying messages that have implicit offensive language. The same researchers expanded on this research by applying the same techniques to English, Hindi and Bangla data (Risch and Krestel, 2020). They had the best performance in the shared task, and the ensemble was better than the single models. However, they also observed that the covert instances of hate speech were found to a lesser degree than the overt ones. This problem will be addressed in this thesis.

2.4 Multi-task learning

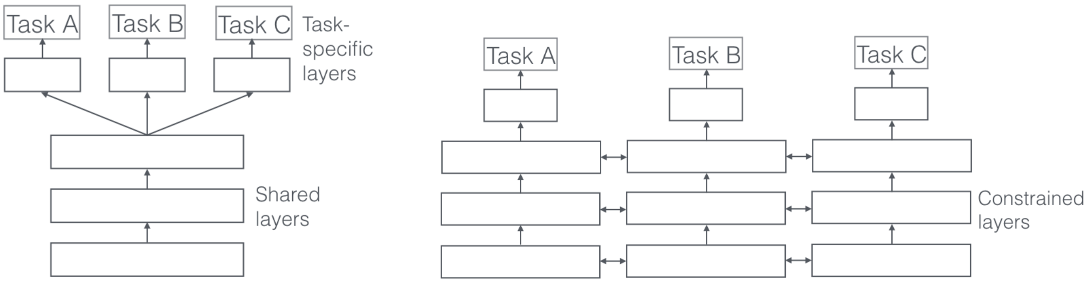
Multi-task learning (Caruana, 1997) (MTL) is a method for deep learning in which multiple tasks are learned at the same time. This section describes what multi-task learning is and how it works. Multi-task approaches to hate speech detection are outlined after that.

2.4.1 What is multi-task learning

Inductive learning in machine learning is when a model is trained by observing data in order to be able to make predictions on unseen data (Michalski, 1983). In single task learning (STL), the learner is trained for one task. For example, during training the model sees examples of data labelled for hate speech. During inference, it can thus predict for new data if it is hate speech or not. In multi-task learning, a model is trained to perform multiple tasks in parallel Caruana (1997). Multi-task learning makes use of inductive transfer, where knowledge from one task influences the inductive bias of the target task Caruana (1997). Multi-task learning is often defined as:

> Given m learning tasks $\{T_i\}_{i=1}^{m}$ where all the tasks or a subset of them are related, multi-task learning aims to learn the m tasks together to improve the learning of a model for each task T_i by using the knowledge contained in all or some of other tasks Zhang and Yang (2021).

There are two methods for multi-task learning: with hard parameter sharing and soft parameter sharing Ruder (2017). In the first, part of the neural network is shared among the different tasks. Additionally, there are task specific layers as illustrated in Figure 2.1a. In soft parameter sharing, on the other hand, each task has their own layers with parameters but they are regularized to be similar. This process is represented by horizontal arrows in Figure 2.1b.



(a) Hard parameter sharing (b) Soft parameter sharing

Figure 2.1: Two methods for multi-task learning (Ruder, 2017)

For the sake of clarity, we make a distinction between multi-task learning and transfer learning. Both operate on the notion that knowledge from one task can be used for learning another task better and more efficiently. Both are thus methods for inductive transfer. However, the method to achieve this is different. Transfer learning happens when a model trained on a source task is used as a starting point for learning a target task Torrey and Shavlik (2010). The knowledge gained from learning the source task is thus used to learn the target task. No data for the target task is available when the source task is learned and vice versa. Fine-tuning a pre-trained model such as BERT for a target task is thus an example of transfer learning. First, BERT has learned to predict masked words and next sentences, then it is fine-tuned for a target task. This can in theory be done as many times as is wanted to utilize different types of information. However, this runs the risk of catastrophic forgetting. This is when the model forgets previously learned information in favor of the newer information Kirkpatrick et al. (2017). The utility of the source tasks thus disappears. The left illustration in Figure 2.2 shows the transfer of knowledge from the first task, to the second. In contrast, in multi-task learning multiple tasks are learned at the same time Caruana (1997). Here, the helper tasks are referred to as auxiliary tasks, as opposed to source tasks. The auxiliary task is a related task to the target task that will help the learner learn the target task better. The information from all tasks is shared among all tasks. This is illustrated by the arrows in Figure 2.2, connecting all four tasks.

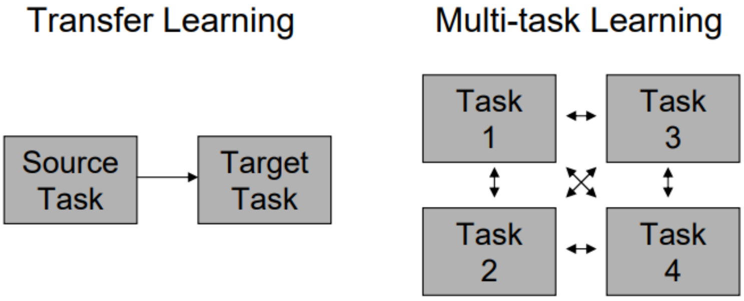


Figure 2.2: In transfer learning information flows in one direction. In multi-task learning, information can flow freely among all tasks. Figure by Torrey and Shavlik (2010).

In this thesis, a combination of the two methods is utilized. We will call this method multi-task fine-tuning. The flow of information is illustrated in Figure 2.3. The pre-trained model BERT is used a starting point. Then this model is fine-tuned on multiple tasks at the same time. Multi-task fine-tuning leverages both types of inductive transfer mentioned above. First, there is knowledge transfer from the pre-trained model to the downstream tasks (signified by the arrow from the source task to the multi-task learning box in Figure 2.3). This is an example of transfer learning. Secondly, there is knowledge transfer among the multiple downstream tasks. This is the multi-task aspect of the training method. The multi-task learning box in Figure 2.3 illustrates that information flows among all tasks simultaneously. One BERT model is shared among the different downstream tasks, making this a hard parameter sharing method. The downstream tasks in this thesis are hate speech detection, which is the target task, and sentiment analysis, emotion detection, sarcasm detection and irony detection, which will serve as auxiliary tasks.

2.4.2 Multi-task learning for hate speech detection

As was established in Section 2.2, hate speech can be detected using features that encode sentiment and emotion. Besides using this information as features, multi-task learning is another way of leveraging semantic information for the desired task.

Three studies have taken this approach. The first study to do so is Rajamanickam et al. (2020), which combined emotion detection and abusive language detection. Three BiLSTMs, are implemented, one with hard parameter sharing (one encoder), and two with soft parameter sharing (double encoder and double gated encoder). The double

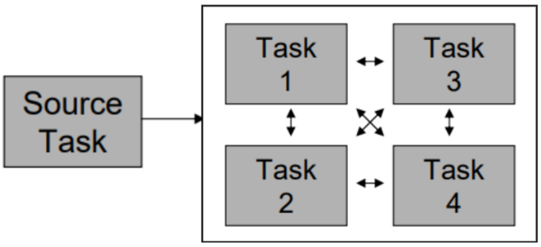


Figure 2.3: Multi-task fine-tuning. The information from the source task flows to all other tasks. The information from tasks 1-4 is shared among them. Figure adapted from Torrey and Shavlik (2010).

gated encoder model performed significantly better than the single task baseline, while the other two models only showed a small improvement. They found that when the model found a negative emotion, the model was also better at predicting hate than the single task model, and when a positive emotion was detected the model was better at classifying non hate. This study compared multi-task learning with transfer learning and found that multi-task learning acheived higher performance than transfer learning. A similar study was done in with Spanish data, but now also incorporating sentiment Plaza-Del-Arco et al. (2021). The Spanish BERT was fine-tuned for emotion detection, sentiment analysis and hate speech detection. They found that, as compared to single task fine-tuning, they could increase the recall of the hateful class. When a negative emotion or sentiment was found, the model was better at detecting hate speech. This corroborates the findings in Rajamanickam et al. (2020). The same was found in the third study. Plaza-del Arco et al. (2021) used emotion detection, sentiment analysis and target identification as auxiliary tasks for offensive language detection on English data. They train models on two tasks at once and all four tasks at once. The model trained on all tasks outperforms a single task baseline and the other multi-task models . Again, the recall of the offensive class is improved significantly.

Other studies have taken a different approach. Section 2.1 outlined the many different types of hate speech. These can all be seen as separate classification tasks (racism detection, sexism detection, target identification, etc). Waseem et al. (2018) and Kapil and Ekbal (2020) have implemented these tasks in a multi-task setup.

In order to increase generalizability across domains, annotation schemes and cultural contexts, Waseem et al. (2018) combine racism detection, sexism detection, offensive language detection and hate speech detection. Their Multilayer perceptron feed forward neural network has shared layers and private layers. The private layers are task specific, while the shared layers learn from all tasks simultaneously. Where in the above studies hate speech detection was the target task, and the others auxiliary tasks, in this study all tasks are equal. The multi-task model outperforms the single task baselines. Kapil and Ekbal (2020) focused on hateful, offensive, racist, sexist and aggressive language and harassment. They also implement a shared-private neural network. They hypothesize that a multi-task model trained on multiple similar tasks has better performance than a model trained on one of the tasks, because there is more data to learn from. All combinations of binary, ternary and quaternary multi-task models are explored. This study also found that multi-task learning performs better than single task learning.

(Implicit) hate speech is also related to sarcasm, irony and other forms of humorous or figurative language (Caselli et al., 2020). Recall from Section 2.2.1 that the effect of using metaphor information has been explored in Lemmens et al. (2021). Leveraging information about figurative language has not been researched in a multi-task setup for textual hate speech detection. However, some work has been done on multi-modal hate speech detection in memes. Classification of memes involves text and images, so it is a multi-modal problem. Chauhan et al. (2020) worked on four tasks simultaneously: humour, sarcasm, offensive content and motivation. Each of the tasks have multiple labels such as 'funny', 'very funny' and 'not funny' for the humour task and 'slightly offensive', 'hateful offensive' and 'not offensive' for the offensive content detection task. They implement two attention based mechanisms: one that finds the relationship between the different classes and one that finds the relationship between the different tasks. These are the shared parameters. Task specific layers for each of the four tasks are implemented for classfication. Compared to learning each task individually, multi-task learning is better at all tasks except for offensive content detection. The analysis of the attention mechanisms showed that the sarcasm and offensive content tasks attended most to each other. This suggests that sarcasm might be useful for hate speech detection in a multi-task learning setup.

Maity et al. (2022) focus on cyberbullying in memes and treat sentiment, emotion, sarcasm and harmfulness as auxiliary information. They gather memes from Twitter and Reddit and annotated them for bullying, sentiment, emotion sarcasm and harmfulness. It is not entirely clear what the difference between the bullying and harmfulness labels are. This study experiments with different neural feature extraction methods. The multi-task models are not compared to single task models that only learn cyberbullying detection. Hence, it is difficult to see what the effect of multi-task learning was in this study. However, cyberbulling detection combined with all three auxiliary tasks is the best performing model compared to other multi-task models where only one to two auxiliary tasks were implemented at a time. This is a promising finding. For both of these studies, the memes were annotated with all the information that was of interest (cyberbullying, sarcasm, emotion. etc.). In contrast, the multi-task studies working solely on textual data use different datasets for each task. Each piece of text is thus only annotated for one task.

2.5 Summary and takeaway

There are many different ways of defining and categorizing hate speech. In this study, we refrain from defining hate speech in a particular manner. Rather, we comply with the terminology and definitions that have been proposed by other researchers to gather and annotate data. These are abusive language in AbuseEval Caselli et al. (2020), verbal aggression in TRAC Kumar et al. (2018) and hateful language in IHC (ElSherief et al., 2021). Hate speech is used as an umbrella term encompassing all of these terms. The focus of this study is on the detection of two different kinds of hate speech: implicit and explicit hate speech. Different machine learning methods, both traditional and deep, have been explored to detect these. A common finding is that the recall of hate speech is too low (Davidson et al., 2017; Warner and Hirschberg, 2012), especially implicit hate speech (Caselli et al., 2020, 2021; Risch et al., 2019; Risch and Krestel, 2020). The present study aims to go against this limitation by increasing the amount of hate found. This is done through multi-task learning. Leveraging sentiment analysis and emotion detection has been found to be effective for hate speech detection (Plaza-Del-Arco et al., 2021; Plaza-del Arco et al., 2021; Rajamanickam et al., 2020). Hate speech is also related to a host of other linguistic phenomena, such as sarcasm, irony and other types of figurative language (Malmasi and Zampieri, 2018; Vidgen et al., 2019). Utilizing this type of information has promising results on hate speech in memes (Chauhan et al., 2020; Maity et al., 2022). In this thesis, sentiment, emotion, sarcasm and irony are all explored. This is the first study to cover such a wide range of phenomena, on multiple textual datasets. This study is also the first to focus on different types of hate speech, i.e. implicit and explicit. This thesis includes a detailed manual analysis investigating the effect of the different auxiliary tasks on implicit and explicit hate speech. Finally, this chapter showed that multi-task learning might also be useful when different hate speech datasets are treated like different tasks. Waseem et al. (2018) and Kapil and Ekbal (2020) found that training on more than one dataset for the same task can improve performance, compared to training on one of them at once. This thesis tests whether the same principle is effective for three other datasets that have not been researched in this capacity.

Chapter 3

Data and terminology

Multiple datasets were used for this thesis. All tasks have their own dataset(s). All the data comes from social media sites, namely Twitter, Facebook and Reddit. This ensures that all the data are in the form of short messages, which can be represented at once by BERT. The data for the three different primary tasks are explained in the first section, as well as the similarities and differences between the datasets. The remaining sections describe the data used for the the four auxiliary tasks.

3.1 Hate Speech

Three different datasets were used for the hate speech detection task. They have all been annotated with different guidelines and all focus on different phenomena, i.e. abusive language, aggression and implicit hate speech. All of these together will be referred to as hate speech, as that can be seen as an umbrella term. The main commonality between the three datasets is that they all operationalize the concept of implicit hate speech. As is described in Chapter 2, implicit hate speech is difficult to recognize by automatic methods.

3.1.1 AbuseEval

AbuseEval v1.0 (Caselli et al., 2020) is a re-annotated version of the Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a). The data was originally gathered and annotated for the SemEval 2019 shared task on offensive language detection (Zampieri et al., 2019b). The dataset consists of English tweets gathered using a list of keywords that might signal offensive content. These include structures such as 'she is' and 'you are' but also content words such as ‘conservatives’, ‘antifa’, ‘gun control’, ‘MAGA’ and ‘liberals’. This dataset thus contains a lot of tweets with a political tone. The original annotation has three levels: a) is the message offensive or not offensive, b) for offensive messages, does it have a target or not, and c) for the targeted messages, is the target a group, individual or other. Offensive language was defined as ‘containing any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct. This includes insults, threats, and posts containing profane language or swear words’ (Zampieri et al., 2019a, p. 1416).

The data was re-annotated with new guidelines for AbuseEval v1.0. Instead of annotating for offensiveness, the focus is on abusiveness, and the explicitness of the abuse. Caselli et al. (2020) define abusive language as ‘hurtful language that a speaker

		Training		Test		Total	
Dataset	Label	No.	%	No.	%	No.	%
AbuseEval	1 (*explicit abuse*)	2,023	15%	106	12%	2,129	15%
	2 (*implicit abuse*)	726	5%	72	8%	798	6%
	0 (*not abuse*)	10,491	79%	682	79%	11,173	79%
	Total	13,240		860		14,100	
TRAC	1 (*overt aggression*)	3,419	23%	144	16%	3,563	22%
	2 (*covert aggression*)	5,297	35%	142	16%	5,439	34%
	0 (*not aggression*)	6,284	42%	630	69%	6,914	43%
	Total	15,000		916		15,916	
IHC	1 (*explicit hate*)	871	5%	218	5%	1,089	5%
	2 (*implicit hate*)	5,680	33%	1,420	33%	7,100	33%
	0 (*not hate*)	10,633	62%	2,658	62%	13,291	62%
	Total	17,184		4,296		21,480	

Table 3.1: Distribution of data classes of datasets used. Due to rounding, not all percentages add up to 100%

uses to insult or offend another individual or a group of individuals based on their personal qualities, appearance, social status, opinions, statements, or actions' (p. 6197). Abusive language thus always targets someone, while offensive language could just be the use of profanity without targeting someone. This definition of abusive language is thus very close to that of hate speech Nockleby (2000). The data was labelled with three different labels: *explicit abuse*, *implicit abuse* and *not abuse*. Explicit abuse is abuse that is evident from the use of clearly negative words, idioms or constructions. Implicit abuse 'can be hidden with sarcasm, metonymy, irony, litotes, euphemism, and inside jokes among other linguistic devices' (Caselli et al., 2020, p. 6197). It does not have any surface evidence for abuse, but abuse can be inferred. The full dataset was annotated by the authors of the study. The majority of the tweets have been labelled as not abusive. Only a small portion of the dataset contains implicit abuse, as is evident from Table 3.1. A slightly larger portion of the data was labelled as *explicit abuse*. The overwhelming majority, however, is not abusive.

3.1.2 TRAC

As a second dataset for hate speech detection, the data for the Trolling, Aggression and Cyberbullying (TRAC) shared task was used Kumar et al. (2018). The data is taken from Facebook and is labelled as overt aggression, covert aggression and not aggressive.

Verbal aggression is defined as 'any kind of linguistic behaviour which intends to damage the social identity of the target person and lower their status and prestige' (Kumar et al., 2018, p. 1425). They distinguish between overt and covert aggression. The use of 'specific kind of lexical items or lexical features' (Kumar et al., 2018, p. 1426) constitutes overt aggression. Covert aggression is indirect and might be hidden in polite structures, satire and rhetorical questions. Overt and covert thus mean the same thing as explicit and implicit.

The data comes from Facebook pages from news websites, forums, political groups, student organizations, and pages discussing incidents in Indian universities. They gathered comments in English and Hindi, but for this thesis only the English data was used.

Four PhD students in linguistics annotated the Facebook comments according to the three classes and their definitions as given above. There were more fine-grained annotation categories, such as physical, sexual or identity threat. As we are interested in implicit or covert hate speech, this level of annotations was not utilized. The development set was added to the train set. Unlike AbuseEval, the *covert aggression* class is not under-represented. In fact, the dataset contains more covert aggression than overt aggression. However, the class imbalance is less severe than in AbuseEval.

3.1.3 Implicit Hate Corpus

The third dataset is the Implicit Hate Corpus (IHC). The authors define implicit hate speech as 'the use of coded or indirect language [...] to disparage a protected group or individual, or to convey prejudical and harmful views about them' (ElSherief et al., 2021, p. 348). They further divide implicit hate speech into six categories: White grievance (WG) is when majority groups are expressed to be the real bearers of racism, while minority groups are perceived to be privileged. Implicit incitement to violence (V) is when power of a hate group is expressed or when a violent hate group is idolized. Normal incitement to violence falls under explicit hate speech. Inferiority (IN) is when one group is expressed as being superior to another by dehumanizing the other. Ironic hateful utterances (IR) contain some sort of figurative or humorous language. It is important to include this, because hate groups use irony to hide their hateful messages (Dreisbach, 2021). In this dataset, the irony category mostly includes jokes in the form of a question answer pair. See Table 3.2 for an example of such a joke. Stereotypes (S) might be used to assign negative traits to minority groups. Language that conveys subtle threats (T) to the safety of the target, be it via pain, injury, damage, loss or violation of rights are the last type of implicit hate speech. Just like incitement to violence, direct threats to the target are seen as explicit hate speech.

All of these types of implicit hate speech are present in the corpus under the label *implicit hate*. The corpus also contains a smaller portion of explicit hate speech.

Tweets posted by hate groups such as Black separatists and white nationalists were gathered. The data was annotated by crowdworkers on Amazon Mechanical Turk. The tweets were first labelled as implicit hate, explicit hate and not hate. Each tweet was labelled by three workers, and a majority vote was applied to get the final gold labels. Later, the instances of implicit hate were annotated again with the six categories and 'other' if none of them fit. This time, trained research assistants performed the annotations. Some of the tweets previously labelled as hate speech were identified to be non hateful in this stage. There are thus two corpora. The high level corpus contains implicit hate speech, explicit hate speech and non hateful tweets. The second corpus only contains implicit hate speech, labelled with the fine-grained schema.

For this thesis, the high level corpus was split into train (80%) and test data (20%). These subsets have the same distribution of the class labels as the full corpus has. Table 3.1

3.1.4 Comparison

There are two big differences between the datasets: the annotation guidelines and the target of the hate. The three datasets have been annotated according to different annotation guidelines and definitions. AbuseEval focusses on abuse towards people based on anything. IHC on the other hand specifies that abusive language is only hate

Dataset	Label	Text
AbuseEval	1	@USER Please call trump out for what he is...a NARCISSISTIC SOCIOPATH!
AbuseEval	1	@USER #metoo are all racist!
AbuseEval	2	@USER @USER Wonder how many children he molested
AbuseEval	2	#GreatestThingsAboutThe90s liberals weren't unhinged
TRAC	1	We want to get rid of u Indians......why don't u hear our loud cries
TRAC	1	You are pro muslim pig
TRAC	2	Judiciary system of India.........????????
TRAC	2	This is happening in BJP ruled state. Shame on the adminstration
IHC	2 (WG)	black lives matter white lives don't? sounds racist.
IHC	2 (V)	kkk alive and well white brother hood unite ku klux klan white power mikey
IHC	2 (IN)	what have black people contributed to science?
IHC	2 (IR)	What's the difference between a black man and property? I don't know.
IHC	2 (S)	best way to stop gun violence in america? disarm all black people
IHC	2 (T)	all jews should be evicted from white countries
IHC	1	muslims are despicable people.

Table 3.2: Examples of implicit and explicit hate speech

speech when it is about a 'protected group or individual'. This protection extends to qualities like race, gender and cultural identity. This is an important difference that is clear in the data. Compare, for example, the tweets from AbuseEval and IHC in Table 3.2. The tweets in AbuseEval are about former president Trump, supporters of the #metoo campaign, liberals and an unspecified individual. These attacks are based on the person itself, or their choices (to support #metoo, to be liberal). The tweets in IHC are targeting black people, Jews and Muslims, because they are black, Jewish or Muslim. TRAC is focussed on verbal aggression, which is defined as aggressive language towards people. However, upon inspection of the dataset, negative stances towards other things than people are observed. One of the Facebook comments is criticising the Indian judiciary system and this is labelled as covert aggression. The way the data was gathered has also impacted what targets are present in the datasets. AbuseEval contains mainly tweets about American politics because keywords such as 'gun control' and 'MAGA' are used. IHC only has messages tweeted by known hate groups. These tweets target protected groups, such as race and religion. TRAC was gathered on Indian Facebook pages. The discussions there revolve around politics and current events in and around India mainly. In both AbuseEval and TRAC there are attacks on protected groups, but they are not limited to this.

3.1.5 Composite dataset

These three datasets are combined to set a baseline. The training data of the three datasets is combined into one training set. The implicit and covert labels are all converted to implicit hate speech and the explicit and overt labels are mapped to explicit hate speech. All the negative labels are not hate speech. The resulting dataset is more balanced than AbuseEval and IHC. There is more implicit hate speech represented than explicit hate speech (see Table 3.3. This dataset now also has a wider range of targets and perpetrators. The test sets remain separate. This ensures that the results of all experiments can be compared.

Label	No.	%
1	6313	14%
2	11703	26%
0	27408	60%
Total	45424	

Table 3.3: Distribution of classes of composite training dataset

3.2 Auxiliary tasks

Four auxiliary tasks are explored in this thesis. They are sentiment analysis, emotion detection, sarcasm detection and irony detection. What each of these tasks are is explained in the sections below, as well as the dataset used for the respective task.

3.2.1 Sentiment analysis

Sentiment analysis in NLP is the task to find the polarity of a text. The polarity of a text refers to the opinions and attitudes expressed in the text towards something (Medhat et al., 2014). Usually sentiment analysis is seen as a classification problem, i.e. a text is either positive, negative or neutral.

For this thesis, data from SemEval-2016 (Task 4: Sentiment Analysis in Twitter) (Nakov et al., 2016) was used. The test set published for the shared task is used as training data here as it is a large dataset. It was also made available as training data for the 2017 iteration of the same task. As the name of the task suggests, all the data is from Twitter. The data was gathered between October and December 2015. The tweets are about 200 topics, found using a named entity extractor. In order to balance the classes somewhat, only tweets that include at least one sentiment bearing word were included. The data is labelled on the document level, i.e. each tweet has one label. The tweets were labelled by five crowdworkers on CrowdFlower on a five point polarity scale. This ranged from -2 (negative) to +2 (positive). The assigned labels were consolidated and transformed into a three point scale, which results in the class labels positive, neutral and negative. The resulting dataset consists of 20632 tweets, most of which have been labelled as neutral. See Table 3.4 for the number of tweets per label.

3.2.2 Emotion detection

Emotion detection, also called emotion recognition, is the task of identifying which emotion is expressed in a piece of data, be it text or images or videos of facial expressions. In textual emotion detection, we try to predict the emotions expressed in a text (Acheampong et al., 2020). Often, emotion detection is done according to some schema based in psychology. Eckman's six basic emotions (Ekman, 1992) or Plutchik's wheel of emotions (Plutchik, 1980) are popular. usually, human annotations are needed to label the data with these classes. Machine learning can be used to learn these annotations in order to predict the same emotions in new text.

The Twitter Emotion Corpus (TEC) (Mohammad, 2012) was used for the emotion classification task in this thesis. This dataset consists of tweets posted between November 15, 2011 and December 6, 2011. The tweets were found by searching for tweets with hashtags signalling one of the six basic emotions (anger, disgust, fear, joy, sadness and surprise) Ekman (1992). An angry tweet thus has #anger, a disgusted tweet #disgust. The tweets are thus self labelled by the user, and subject to irony/sarcasm. Automatically labelling tweets according to hashtags is cheap (no human annotators needed), so a large number of tweets were gathered. The tweets with less than three English words were removed by the original researcher. However, there were still tweets which were mainly written in languages other than English. These were removed using the langdetct package in python[1]. 19,349 English tweets remain after filtering. The dataset is not balanced, with most tweets being joyous and the least tweets expressing disgust. See Table 3.4 for the number of tweets per label. There is no neutral label and all the tweets have one emotion.

3.2.3 Sarcasm detection

Sarcasm is usually defined as verbal irony, where the there is some incongruity between the sentiment expressed in the text and the situation (Sarsam et al., 2020). 'I am so happy that my car broke down' is an example of sarcasm as the sentiment is positive (‘happy’) but the situation is negative ('my car broke down') and the speaker of this utterance most likely was not happy about it. Sarcasm detection in text is the task of deciding whether a text is sarcastic or not sarcastic. For this thesis, we use two datasets made available for the shared task on sarcasm detection for the second workshop on figurative language processing (Ghosh et al., 2020). These dataset are subsets from other datasets by Khodak et al. (2018) and Ghosh et al. (2018).

Khodak et al. (2018) introduced the Self-Annotated Reddit Corpus (SARC). This is a large dataset of over 500 million Reddit comments, of which over a million are sarcastic. The comments were all posted between January 2009 and April 2017. The sarcastic comments were found by searching for comments with \s. This is an indicator that the message is sarcastic. This dataset is thus labelled with intended sarcasm, because the author of the comment meant it sarcastically. This is different from perceived sarcasm, where the reader interprets the message as sarcastic. Intended sarcasm is easier to compile into a dataset because signals such as \s can be used to find it. There is no labor required to label the messages manually. 4400 comments from the large dataset were taken as training data for the shared task. The previous comments in the thread are included as context.

The data from Twitter Ghosh et al. (2018) is also self labelled. Tweets with #sarcasm, #sarcastic and #irony at the end of the tweet were mined. The non-sarcastic tweets had to convey either a negative or positive sentiment. This is because it is easier to distinguish between sarcasm and objective utterances than between sarcasm and tweets that contain sentiment. The tasks is thus quite challenging. Only tweets which are a reply to another tweet were included. 5000 of these tweets, and their conversational context, were selected as training data for the shared task.

Both the Reddit and Twitter data are perfectly balanced. The Twitter data was appended to the Reddit comments and used as one training set for this study. The contextual tweets and comments are not used in this thesis, as the focus is on hate speech, not sarcasm detection. See Table 3.4 for the distribution of the classes.

[1] https://pypi.org/project/langdetect/

Dataset	Label	No.	%
SemEval-2016-4 (SA)	Positive	7,059	34%
	Negative	3,231	16%
	Neutral	10,342	50%
	Total	20,632	
TEC (ED)	Anger	1,497	8%
	Disgust	736	4%
	Fear	2,562	13%
	Joy	7,894	41%
	Sadness	3,610	19%
	Surprise	3,050	16%
	Total	19,349	
Twitter+Reddit (SD)	Sarcasm	4,700	50%
	Not sarcasm	4,700	50%
	Total	9,400	
SemEval-2018-3 (ID)	Irony	1,901	50%
	Not irony	1,916	50%
	Total	3,817	

Table 3.4: Distribution of classes for the datasets for the auiliary tasks. SA = sentiment analysis, ED = emotion detection, SD = sarcasm detection, ID = irony detection. Due to rounding, not all percentages add up to 100%

3.2.4 Irony detection

Irony in text is very similar to sarcasm, but broader. Verbal irony, when what is meant is not what is said, is the same as sarcasm. There is also situational irony Van Hee et al. (2018). For irony detection, the data gathered for the SemEval-2018 shared task on Irony Detection in English Tweets (task 3) (Van Hee et al., 2018) is used. 3000 tweets were gathered by searching for #irony, #sarcasm and #not among other to find tweets with irony. These were annotated manually by linguistics students to ensure they are ironic. Non-ironic tweets were added to the dataset to balance the class distribution. The datasets thus distinguishes between irony and not irony; a binary classification. All the tweets were gathered in December 2014 and January 2015. The hashtag signalling irony was removed from all the tweets for the final train and test sets. Tweets written in languages other than English, retweets and duplicates were also removed. Only the training set was used for this study. Table 3.4 shows the number of tweets per label and the total number of tweets.

3.3 Text preprocessing

To prepare the data for BERT, a number of preprocessing steps are applied. All mentions of Twitter users are converted to @USER, so the specific names cannot be used for classification. Tweets often contain hashtags. Multi word hashtags are segmented into individual words with wordsegment in python[2] and the hashtag itself is removed. For example, the hashtag #getout would be transformed into two words: ‘get out’. All URLs are replaced with the placeholder URL. The maximum sequence length was set

[2] https://grantjenks.com/docs/wordsegment/

to 40. Longer messages are truncated, shorter ones are padded on the right. Longer sequence lengths result in a lot of padding, and this was found to not be beneficial to learning in preliminary experiments. The data is also lowercased and tokenized with the BERT WordPiece tokenizer. All the data is preprocessed in this same way.

Chapter 4

Method

Central to the method of this thesis is multi-task fine-tuning. The goal of this thesis is to explore the effect that combining different (auxiliary) tasks has on the detection of hate speech. The architecture of the multi-task model utilized in this thesis is explained in Section 4.1. Three different, but similar, hate speech tasks are explored, as well as four auxiliary tasks. The auxiliary tasks serve solely as helpers, we are not interested in the performance of these tasks. The auxiliary tasks that are explored in this thesis are sentiment analysis, emotion classification, sarcasm detection and metaphor detection. Multiple experiments will be conducted to gain insight into where we can yield an increase in performance. The conducted experiments and the method for evaluation are are explained in this section.

4.1 Multi-task fine-tuning architecture

In multi-task learning, there are multiple classifiers that share some hidden layers. In this thesis, this is achieved by fine-tuning BERT, where the BERT encoder is the shared part and private classifiers are added. The architecture and code to implement it is based on Sun et al. (2019).

BERT outputs representations of text. For single task fine-tuning, a classification head is added on top of BERT. This classifier takes the representations and turns them into class predictions. For multi-task fine-tuning, multiple classification heads, one for each task, get added. The BERT encoder is shared among the task specific classifiers. A schematic view of this setup is presented in Figure 4.1. Essentially, the text (presented in red) gets represented by the BERT encoder (the output is the green text), the representation of the whole message is then directed to its matching classification head. During training, the examples that have been annotated with abusive labels are thus directed to the abusive language classification head, the data for the sentiment analysis task are directed to the designated sentiment classifier and so on. The classification heads are structured as follows. The representation of the CLS token is taken from the last hidden state of the BERT encoder. This CLS token is used for next sentence prediction in the pre-training of BERT and thus represents the whole sentence. This token representation is passed to a linear layer with input size 768 (size of the CLS token) and outputs as many dimensions as there are labels for that specific task. This output is a probability score for each of the labels. The label with the highest probability is the final class prediction. In the fine-tuning process, the pre-trained weights of the BERT encoder and the randomly initialized weights of the classification heads (linear

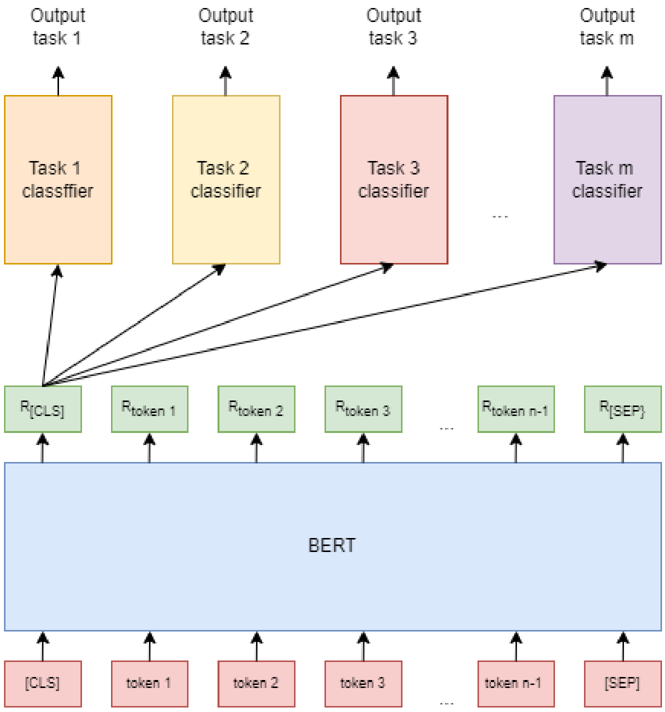


Figure 4.1: Multi-task fine-tuning architecture with input sequence length n and m tasks.

layers) are updated after each batch of training examples. Each batch consists of data from one task. The order of the batches is shuffled, so the model learns from all tasks simultaneously, as opposed to transfer learning where first one task is learned, and then another. How much and in which direction the weights are updated is based on the Cross Entropy loss of the predicted output compared to the gold label. The Adam optimizer, a method for optimization similar to stochastic gradient descent, (Kingma and Ba, 2015) is used to update the weights. Dropout is implemented to prevent overfitting. This means that random neurons are set to zero, based on a predefined probability.

During inference, the message can be directed to any classification head, depending on the desired output. Here we want to predict hate speech, but also want to see what the other classifiers predict, so the test data is directed to all the classifiers. It should be noted that, while hate speech is referred to as the primary task, all task are treated equally inside the model and in the training process. We evaluate the models on their ability to classify hate speech, but not the other tasks, as this is what we are interested in.

4.2 Experiments

This section describes the different experiments that have been conducted to answer the research questions. The first research question is: *What is the effect of different auxiliary tasks on different types of hate speech?* The first set of experiments aims to answer this question by implementing one of these auxiliary tasks at a time. In these models two tasks are learned at the same time; the target task hate speech detection (operationalized as one of three hate speech datasets) and one auxiliary task. These models are referred to as binary models. These experiments show the impact of each auxiliary task individually. For this thesis, we have selected four auxiliary tasks - sentiment analysis, emotion detection, sarcasm detection and irony detection - and aim to investigate their impact on three hate speech datasets. Consequently, there are twelve different binary multi-task models. Next, we build on the results of these experiments. The auxiliary tasks that improve the performance will be combined to find out if the performance can be improved even more. These will thus be ternary models, as three tasks are learned in parallel. All the models are tested on the hold-out test set of the respective primary training data.

The second research question is: *What is the effect of training on multiple datasets for hate speech at the same time?* To answer this question, we train on AbuseEval, TRAC and IHC at the same time. This model thus learns the following three tasks in parallel: abusive language detection, aggression detection and hate speech detection. Because there are multiple classifiers, we can predict these different phenomena separately from each other. There is the ability to learn from more data than when training on just one the datasets at once, but there is also still room for the subtle differences in the tasks. This should make the classification better than just combining all the data into one dataset, and thus one task. There is also the ability to place more importance on one or some of the tasks by introducing a weight to the loss (Waseem et al., 2018). This is not implemented in this thesis.

4.2.1 Baselines

To test the effect of multi-task fine-tuning, the models are compared to single task baselines. There are four baselines. There are three baselines trained and tested on only one of the tree datasets at a time: one for AbuseEval, one for TRAC and one for IHC. Comparison with these models will show the difference the auxiliary tasks make and what difference it makes to train on all of these datasets at once. There is a fourth baseline that is trained on the composite dataset of all three. All of the data is combined into one dataset and the model is trained on this in a single task manner. The model is tested on all three test sets separately to make comparison possible. This model will serve as a baseline for the second experiment to find out whether multi-task learning is beneficial outside of just having more data for training. The same initialization of the $BERT_{BASE}$ model as for the multi-task models is used as a starting point for the single task baselines. One classifier is added to predict the respective classes of the hate speech datasets by means of a linear layer. The training procedure is similar to the multi-task model in that the representation of the CLS token is taken and used for the classification. The randomly initialized parameters of the linear layer and the parameters in BERT are updated during training.

4.2.2 Hyperparameters and implementation

For all implementations, the $BERT_{BASE}$ uncased model was used. All models are trained with Adam optimizer with epsilon set to 1e-8 and dropout probability of 0.1. The maximum sequence length was set to 40. Messages with less than 40 wordpieces are padded with [PAD] at the right, and longer messages are truncated. Every model was trained for three epochs, with learning rate 2e-5 and batch size of 16. These hyperparameter settings are suggested in the original BERT paper Devlin et al. (2019) and were empirically confirmed to work well on a subset of the models in preliminary experiments. Each experiment was conducted five times with the same parameters, with the only variable value being the random seed. This influences the order the model sees the training examples in (the order of the batches and the order of the examples within a batch) and the initialization of the weights in the linear layers. The median of the five runs is reported. All experiments were implemented with the transformers and pytorch libraries in python.[1] The code was run in Google Colaboratory with a Tesla K80 GPU.

4.3 Evaluation

The models will be evaluated based on their performance on the hold-out test data. The models are only tested on their ability to predict hate speech. All the test data is in-domain. So the models trained on the AbuseEval data are tested on the AbuseEval test data, TRAC models are tested on the TRAC test data and IHC models are tested only on the IHC test data. The performance of all models is assessed with quantitative measures. In addition, a manual analysis is performed to gain insight into where the multi-task models gain or lose performance over the baselines.

4.3.1 Quantitative evaluation

The performance of the models on the hold-out test sets will be measured with commonly used metrics in NLP: precison, recall and macro F1-score. Precision aims to answer the question, out of all the times the model predicted this label, how many times was this correct? The correct predictions are the true positives, and the instances where the model did predict the label, but this is incorrect, are the false positives. Precision is calculated as follows:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

The recall score shows how much of the labels were found. It is the fraction of times the model predicted the label correctly over all the times it should have been predicted. The times it should have been predicted are the times the model did correctly predict the label (the true positives) and also the times it predicted the instance belonging to another class (the false negatives). Recall is thus:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

The F1-score is a combination of the two scores and thus sums up the performance of the model. F1-score is the harmonic mean of precision and recall:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

These three scores are calculated for all of the classes. The average of all three classes is also reported to give an overview of the models performance. For this the macro average is used. In this average all classes weigh equally. The choice to use macro F1 instead of weighted F1 is because of the class imbalance. The majority of the examples in all

[1] Transformers library: `https://huggingface.co/docs/transformers/index`. PyTorch library: `https://pytorch.org/`

three datasets is not hate speech. Because we are mainly interested in the performance of the explicit and implicit classes, not the ‘not hate speech’ class, we weigh these all equally.

If the performance of the model is better than the baseline, the significance of this finding will be tested with the McNemar test (McNemar, 1947):

$$\chi^2 = \frac{(b-c)^2}{b+c}$$

where b and c are:

		classifier	
		correct	incorrect
baseline classifier	correct	a	b
	incorrect	c	d

This test is thus a pairwise test of two models and compares the times either one of the two classifiers was incorrect while the other was correct in its prediction. Results where $p < 0.05$ are significant.

4.3.2 Qualitative evaluation

A qualitative analysis will be performed to gain deeper insight into the strengths and weaknesses of the multi-task models. The error analysis focuses on the false negatives and false positives of the multi-task models compared to the baseline. Moreover, attention will be payed to the true positives of the implicit classes. The predictions of the auxiliary classifiers will reveal if the implicit instances were in fact identified as having negative sentiment and negative emotions, as well as if these were sarcastic or ironic. An analysis of the true negatives (when no hate speech was found and this is correct) will also show if the auxiliary tasks might have helped the model decide between mere profanity (not hate speech) and hate speech.

Chapter 5

Results

In this section the results of all the experiments are presented. The first research question is First the experiments with binary models are presented. The results of these experiments will dictate what other experiments are done with the auxiliary tasks. The second set of experiments is aimed at improving hate speech detection by using multiple datasets. All models have been run five times and the models with median performance have been selected for this section.

5.1 Binary models

The first subquestion aims to find out what the effect of different auxiliary tasks on explicit and implicit hate speech is. First, each of the three datasets is combined with one auxiliary task at a time. Table 5.1 shows the performance of each of these twelve models and the single task baselines. The single task models are trained only on either the AbuseEval training data, TRAC training data or IHC training data. The multi-task models are trained on one of those training sets and the data from one other task. The models' performance are measured on the hold-out test data corresponding to the training data. For example, the scores in the 'AbuseEval' column and the 'Sentiment analysis' rows reflect the performance of the multi-task model trained on AbuseEval data and sentiment analysis data. The performance on each class is given and the average of these. The three different test sets all have three classes. The classes *explicit abuse*, *overt aggression* and *explicit hate* correspond to 1 in the table, *implicit abuse*, *covert aggression* and *implicit hate* correspond to 2. 0 stands for all the negative labels: *not abusive*, *not aggressive* and *not hateful*.

The baseline for AbuseEval scores 54.4 in terms of F1. Generally, *not abuse* (class 0) has the highest scores and *implicit abuse* (class 2) has the lowest scores. Especially the recall of *implicit abuse* is low (highest 13.9). This is in line with the amount of training data for these classes. There was the most training data for 'not abuse' and the least for 'not abuse'.

The baseline for TRAC scored F1 score of 53.1. None of the multi-task models succeeded in beating this score. All the models perform worst on *covert aggression* (class 2) and best on *not aggression* (class 0). Recall that 23% of the training data is *overt aggression* (class 1) and 35% of the training data is *covert aggression.* The low performance on *covert aggression* is not due to lack the training data, but points to an actual difficulty in learning this class.

The baseline model for the IHC dataset scores an F1 measure of 57.9. All multi-task

		AbuseEval			TRAC			IHC		
Auxiliary task		P	R	F1	P	R	F1	P	R	F1
None (STL baseline)	0	85.9	94.6	90.0	84.5	67.3	74.9	76.7	91.8	83.6
	1	62.0	53.8	57.6	44.3	64.6	52.5	50.0	22.9	31.4
	2	41.2	9.7	15.7	27.0	38.7	31.8	70.4	50.3	58.7
	avg	63.0	52.7	54.4	51.9	56.9	53.1	65.7	55.0	57.9
Sentiment analysis	0	86.6	92.5	89.4	85.2	55.6	67.2	78.0	89.3	83.3
	1	55.9	58.5	57.1	51.2	57.6	54.2	46.3	31.7	37.6
	2	35.0	9.7	15.2	23.0	55.6	32.6	68.4	53.3	59.9
	avg	59.1	53.6	53.9	53.1	56.3	51.4	64.3	58.1	**60.3**
Emotion detection	0	87.1	95.7	91.2	84.2	62.4	71.6	82.4	80.2	81.3
	1	67.0	59.4	63.0	46.7	59.0	52.1	40.6	29.8	34.4
	2	31.3	6.9	11.4	23.2	43.7	30.3	61.3	67.0	64.0
	avg	61.8	54.0	**55.2**	51.4	55.0	51.4	61.5	59.0	**59.9***
Sarcasm detection	0	84.8	95.0	89.6	85.7	60.6	71.0	81.4	82.5	81.9
	1	62.0	46.2	53.0	46.4	62.5	53.3	38.8	30.3	34.0
	2	29.4	6.9	11.2	25.4	49.3	33.5	62.9	63.5	63.2
	avg	58.8	49.4	51.3	52.5	57.5	52.6	61.0	58.7	**59.7**
Irony detection	0	86.2	93.4	89.7	85.1	62.5	72.1	82.1	80.1	81.1
	1	61.8	51.9	56.4	45.2	61.8	52.2	40.4	31.7	35.5
	2	31.3	13.9	19.2	24.6	44.4	31.7	61.0	65.8	63.3
	avg	59.7	53.1	**55.1**	51.6	56.2	52.0	61.2	59.2	**60.0***

Table 5.1: Performance of all binary models trained on the training set of the respective dataset and one auxiliary task. F1-scores that outperform the baseline are marked in bold. Statistically significant increase of F1-scores over the baseline according to the McNemar test ((McNemar, 1947) are marked with *

models improve on this. All models score lowest on *explicit hate* (class 1) and highest on *not hate* (class 0). Only 5% of the training data was labelled *explicit hate*, so this finding is not surprising. In the AbuseEval dataset, 5% was labelled *implicit abuse*, and the scores for this class are lower than the scores for class 1 of the IHC models. This seems to suggest that explicit instances of hate speech are easier to learn than implicit instances. The effect of each of the auxiliary tasks is described below.

5.1.1 Sentiment analysis

Training for sentiment analysis has a different effect on the three datasets. When combined with AbuseEval, sentiment analysis improves the recall of *explicit abuse* (+4.7 points), but lowers the precision by 6.1 points. Performance on *implicit abuse* has not improved. The overall performance of this model is 0.5 lower than the baseline in terms of average F1 score. On TRAC, the performance of both *overt aggression* and *covert aggression* are improved. This is due to a higher precision for *overt aggression* and higher recall of *covert aggression*. The ability to predict *not aggression* decreased, so overall the model did not perform better than the baseline (F1=51.5). The effect of sentiment analysis on IHC is generally positive and the overall F1 score is 60.3 The average F1 score increased 2.4 points. The recall of *explicit hate* and *implicit hate* are increased, while the precision of these classes suffer.

5.1.2 Emotion detection

Utilizing emotion detecion as an auxiliary task has increased the performance of hate speech detection on AbuseEval and IHC. More specifically, the performance of *explicit abuse* on AbuseEval is increased (+5.6 recall, +5 precision points). However, the performance of *implicit abuse* went down, especially the recall (-9.9 points). The average F1-score (55.2) is higher than the baseline, but this finding is not significant. On TRAC, emotion detection increased the precision of *overt aggression* and the recall of *covert aggression.* However, the rest of the scores go down and the overall performance of the multi-task model (F1=51.4) is lower than the baseline. On IHC, a F1-score of 59.9 was achieved. This high score is due to an increase in the recall of *explicit hate* and *implicit hate.*

5.1.3 Sarcasm detection

On AbuseEval, sarcasm detection led to a F1-score of 51.3, which is lower than the baseline. The multi-task model performs worse on both *explicit* and *implicit abuse.* On TRAC, a F1-score of 52.6 is achieved, which is the highest among the multi-task models but still does not beat the baseline. Again, the recall of *covert aggression* is increased, this time by 10.6 points. The precision of *overt aggression* also improved slightly (+2.1 points). A non-significant increase in performance was observed on IHC, the F1-score increased from 57.9 in the baseline to 59.7. Especially the recall of *implicit hate* went up (+13.2 points). Recall of *explicit hate* also increased (+7.4 points). The precision of both these classes went down.

5.1.4 Irony detection

Irony detection, when used as an auxiliary task, had positive effect on all implicit types of hate speech. On AbuseEval the F1-score is 55.1, which is the highest score for AbuseEval. However, this result is not significant. The recall of *implicit abuse* increased 4.2 points compared to the baseline, but the precision lowered. The performance on *explicit abuse* also lowered. On TRAC a F1-score of 52.0 was achieved. Again, this model did not beat the baseline, but it did improve the recall of *covert aggression.* Irony detection for IHC has similar results as sarcasm detection: the recall of *explicit hate* and *implicit hate* increase, but the precision goes down. The overall F1-score of the model is 60.0, which is significantly better than the baseline.

5.2 Ternary models

The tasks that improved the performance of the models, are investigated further. We combine two auxiliary tasks at a time with the primary task. These models are thus trained on three different tasks in parallel, making them ternary models. Increase in performance was only observed on AbuseEval and IHC. The results of the experiments are presented below.

5.2.1 AbuseEval ternary model

Emotion detection and irony detection improved the performance on AbuseEval. Following these results, in the next experiment, these two auxiliary tasks are learned

Auxiliary tasks		P	R	F1
None (STL baseline)	0	85.9	94.6	90.0
	1	62.0	53.8	57.6
	2	41.2	9.7	15.7
	avg	63.0	52.7	54.4
Emotion detection + irony detection	0	85.8	95.5	90.4
	1	63.5	57.5	60.4
	2	20.0	1.4	2.6
	avg	56.4	51.5	51.1

Table 5.2: Performance of a ternary model on AbuseEval test set

alongside abusive language detection. This model is thus trained on three tasks. The resulting model is able to predict the abusiveness, the emotion and the ironic value of a message. The performance scores of this model can be found in Table 5.2.

Combining emotion detection and irony detection with hate speech detection did not yield better results than the baseline (F1=5.1 compared to baseline F1=54.4). Performance on *not abuse* and *explicit abuse* increased, but the performance on *implicit abuse* is lower. Especially the recall of *implicit absue* is very low (1.4). This model does worse than any of the other AbuseEval models. The optimal training data thus seems to be a combination of the irony detection dataset and AbuseEval itself.

5.2.2 IHC ternary models

All four auxiliary tasks increased the performance of hate speech detection on the IHC dataset. The target task and the auxiliary tasks were combined into different ternary models. Because all auxiliary tasks improved the performance, all combinations are explored.

All six ternary models beat the baseline in terms of F1-score, as is evident from Table 5.3. These models have the same tendencies as the binary models; they are best at predicting class 0 and worst at class 1.

The best model is trained on sarcasm detection, irony detection and IHC (F1=60.2).

The model trained on sentiment analysis and emotion detection improved the recall of *explicit hate* and *implicit hate*, resulting in a higher overall score than the baseline. This model is quite close in general performance to the binary models with sentiment analysis and emotion detection. However, the high recall that the model trained on emotion detection reached was not repeated by the ternary model (56.3 for sentiment analysis+emotion detection+IHC versus 67.0 for emotion detection). Training on sentiment analysis and sarcasm detection is also better than only training on IHC. It also improves upon training on sarcasm detection and IHC (F1=59.7), but not on sentiment analysis and IHC (F1=60.3). Combining IHC data with sentiment analysis and irony detection resulted in the highest recall of class 2 (75.4), and also increased the recall of class 1. This is at the cost of recall of class 0. Nonetheless, this model beats the baseline significantly. Training on emotion detection+sarcasm detection+IHC and emotion detection+irony detection improves upon the baseline, but not significantly. Like the other ternary models, these models were able to increase the F1 score of class 1 and 2, but not for class 0. The model trained on sarcasm detection+irony detection scored the highest among the ternary models, with a signicantly higher F1-score than

Auxiliary tasks		P	R	F1
None (STL baseline)	0	76.7	91.8	83.6
	1	50.0	22.9	31.4
	2	70.4	50.3	58.7
	avg	65.7	55.0	57.9
Sentiment analysis + emotion detection	0	78.8	87.9	83.1
	1	48.5	28.9	36.2
	2	66.6	56.3	61.0
	avg	64.6	57.7	**60.1**
Sentiment analysis + sarcasm detection	0	80.3	83.6	81.9
	1	44.1	30.7	36.2
	2	63.1	61.1	62.1
	avg	62.5	58.5	**60.1**
Sentiment analysis + irony detection	0	85.7	73.6	79.2
	1	46.2	27.5	34.5
	2	56.9	75.4	64.9
	avg	62.9	58.8	**59.5***
Emotion detection + sarcasm detection	0	79.3	85.7	82.4
	1	42.1	31.7	36.1
	2	64.7	57.3	60.7
	avg	62.0	58.2	**59.7**
Emotion detection + irony detection	0	79.6	85.3	82.3
	1	44.8	25.7	32.7
	2	63.8	59.4	61.6
	avg	62.7	56.8	**58.8**
Sarcasm detection + irony detection	0	80.6	83.4	82.0
	1	38.4	34.9	36.5
	2	63.8	60.5	62.1
	avg	60.9	59.6	**60.2**

Table 5.3: Performance of ternary models on IHC. F1 scores higher than the baseline are marked in bold. Statistically significant increase of F1-scores over the baseline according to the McNemar test ((McNemar, 1947) are marked with *

the baseline (60.2 versus 57.9). This is, however, not higher than the highest F1-score of the binary models (F1=60.3 for the sentiment analysis model).

5.3 Composite ternary model

The second subquestion is *what is the effect of training on multiple datasets for hate speech at the sentiment analysisme time?* To answer this question, the three datasets are treated like different tasks: abuse detection, aggression detection and hate speech detection. A multi-task model is trained on all three at the same time. Table 5.4 shows the results of this experiment. The STL Composite model is trained on all the data in a single task manner. The MTL Composite model is trained on all the tasks, but with a separate classifier for each task. The results are given for each of the three test sets.

The baseline of AbuseEval has an F1-score of 54.4. Neither composite models have beat this, the STL Composite model achieving an F1-score of 52.0 and the MTL Composite model 52.5. Multi-tasking increased the recall of *explicit abuse*, but lowered the precision. The recall of *implicit abuse* decreases, even though it was already very low. Overall, scores on *explicit abuse* are higher than on *implicit abuse.* This is expected for the single task baseline, because there is only a small amount of *implicit abuse.* However, it is not expected for the composite models. In the combined dataset, there is more implicit hate. The difficulty in identifying it is thus probably not due to the lack of data, but this points to an actual difficulty in detection this type of hate speech.

On TRAC, the best score is also achieved by the STL baseline (F1=53.1). The overall performance of the STL composite model is 48.5 in terms of F1 and 48.6 for the MTL composite model. There is a big increase in the detection of *covert aggression*: the recall increased from 38.7 to 65.5 in the MTL model. The recall of *not aggression* drops to 43.5 in the MTL model.

There is an improvement in performance observed on IHC. The F1-score goes from 57.9 (baseline) to 58.6 in the MTL Composite model. In contrast to the other two datasets, the scores on *implicit hate* are higher than on *explicit hate* on this dataset. IHC is the largest dataset among the three, so a bigger proportion of the training data came from this dataset. This might be why the performance increased but did not for the other two datasets. The recall of both *implicit* and *explicit hate* increased by learning these tasks in a multi-task setup. Again, we see that this goes hand in hand with a decrease in recall for the negative class, *not hate.*

5.3.1 Summary

None of the auxiliary tasks have proven to be generally useful for hate speech detection as they show different results on each of the three hate speech datasets. Irony detection increased the recall of all the implicit classes across the three datasets. This is the only auxiliary task that has such a clear effect. The other tasks had mixed results. Sentiment analysis and emotion detection improved the recall of *explicit abuse*, *covert aggression* and *implicit* and *explicit hate speech.* Sarcasm detection increased the performance on the latter three classes as well.

On the IHC dataset all auxiliary tasks improved the performance, but none did on the TRAC dataset. The recall of *covert aggression* was improved by all four auxiliary tasks, though. On the AbuseEval data, only emotion detection and irony detection made a positive difference, but this was not significant. Training on all three datasets

		AbuseEval			TRAC			IHC		
Model		P	R	F1	P	R	F1	P	R	F1
STL baseline	0	85.9	94.6	90.0	84.5	67.3	74.9	76.7	91.8	83.6
	1	62.0	53.8	57.6	44.3	64.6	52.5	50.0	22.9	31.4
	2	41.2	9.7	15.7	27.0	38.7	31.8	70.4	50.3	58.7
	avg	63.0	52.7	54.4	51.9	56.9	53.1	65.7	55.0	57.9
STL Composite	0	86.6	82.4	84.4	89.0	47.5	61.9	82.6	74.2	78.2
	1	56.0	48.1	51.8	48.1	52.1	50.0	35.3	29.8	32.3
	2	15.8	26.4	19.8	22.4	66.9	33.6	56.4	68.4	61.8
	avg	52.8	52.3	52.0	53.2	55.5	48.5	58.1	57.5	57.4
MTL Composite	0	86.8	91.2	88.9	90.7	43.5	58.8	81.2	79.8	80.5
	1	53.4	59.4	56.3	42.3	66.7	51.8	46.3	26.1	33.4
	2	24.0	8.3	12.4	24.0	65.5	35.2	59.0	64.9	61.8
	avg	54.7	53.0	52.5	52.4	58.6	48.6	62.2	56.9	58.6

Table 5.4: Performance of composite models. F1-scores that outperform the baseline are marked in bold. Statistically significant increase of F1-scores over the baseline according to the McNemar test ((McNemar, 1947) are marked with *

with a multi-task model could also only improve performance on IHC.

Chapter 6

Analysis and discussion

This chapter will provide a manual analysis of some of the models' predictions. The goal is to investigate how the auxiliary tasks affected the predictions and also to find out where the models still face challenge. In the previous section we found that most of the tasks have different effects on the different datasets. In this section, the most outstanding positive results per auxiliary task are investigated further through manual analysis. There are no gold labels for the auxiliary tasks on the test data. All the claims about whether the predictions are correct are according to the author of this thesis.

6.1 Sentiment and covert aggression in TRAC

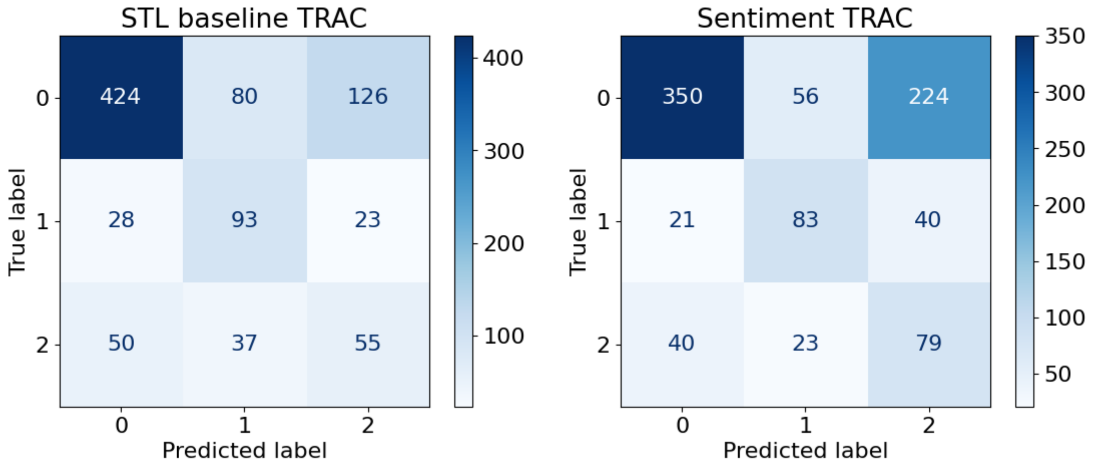


Figure 6.1: Confusion matrices of single task baseline and sentiment analysis multi-task model on TRAC.

On TRAC, covert aggression was detected more by the MTL-sentiment model than the baseline (55 to 79, see Figure 6.1). However, the number of false negatives increased by a large number too (126 to 224). The recall of *not aggression* went down; while the baseline found 424 of them, the MTL-sentiment model only correctly labelled 350 instances as *not aggression*.

Sentiment analysis increased the recall of covert aggression the most, so this section will focus on the covert aggression label.

Table 6.1 shows the times the predicted labels co-occur. *Covert aggression* was mostly predicted along with neutral sentiment (254 out of 343). *Overt aggression* is more associated with negative sentiment (154 out of 162) and *not aggression* is also associated with *neutral* mostly, but also with *positive*. This is in line with the definition of covert and overt aggression. Overt is direct and clear from the surface form of the text, thereby being also overtly negative. The aggression in covert instances is not direct, and therefore less often clearly negative.

		Sentiment		
		Negative	Neutral	Positive
Aggression	0	37	265	109
	1	154	8	0
	2	83	254	6

Table 6.1: Co-occurrence of aggression labels and sentiment labels predicted by MTL-sentiment on TRAC.

True positives True positives of the *covert aggression* class are instances where the model correctly identified covert aggression in texts.The number of true positives increased from 55 for the baseline to 79 for the MTL-sentiment model. 23 of the 79 tweets were labelled *not aggression* by the baseline. All 23 texts have been labelled as *neutral* in terms of sentiment. Two types of texts in the true positives stand out: criticism of politicians and political decisions and comments on a speech by former Prime Minister of India Manmohan Singh. Examples of both of these are test 1 and 2 and can be found in Table 6.2.

False negatives The MTL-sentiment model missed 40 covertly aggressive texts, and labelled them as *not aggressive*. 11 of these are labelled as having a positive sentiment. These texts have a sarcastic tone or are backhanded compliments to Prime Minister Modi for making the former Prime Minister speak out on demonetisation. These are unsurprisingly labelled positive, and thus it is not strange these are false negatives. The aggression is very indirect and not clear from the surface of the text (See example 3 and 4). 26 texts were labelled *neutral*. All of them seem to be not aggressive and the neutral label seems fitting. The author is not from India, so these might be unacceptable in the Indian culture. Texts 5 and 6 are examples of this. There are three false negatives with negative sentiment.

False positives There is a sharp increase of false positives from the baseline to the MTL-sentiment model (126 to 224). These texts have been labelled as *covert aggression* while they are not aggressive. Most of them are neutral in sentiment. It seems like the model has learned to associate neutral sentiment with covert aggression. Because a lot of not aggressive texts are also neutral, these have have also been labelled as *covert aggression*. Text 7 and 8 are such neutral messages that are falsely labelled as *covert aggression*.

	text	gold	STL baseline	MTL-sentiment	
				hate	sentiment
1	modi should learn from him how & what to communicate common people not to corporate people.	2	0	2	neutral
2	MM gave a speech, definitely he obeyed the someone's order as always..	2	0	2	neutral
3	ohh god he can speak now ohh really modi has some magic with him that our ex pm is speaking he got his voice back with demonetisation	2	0	0	positive
4	Wow....this guy can talk also....thanks modi ji for making mannequin mohan talk...	2	0	0	positive
5	Wait for a day.. there will be press release.. and usual panel discussions in the prime times. U will get the prescribed answer.	2	0	0	neutral
6	Mohammed Allaudin. Make your brain digital and win prizes.	2	1	0	neutral
7	Buddy we know you will deliver ..keep going , also stop watching cnbc tv 18	0	0	2	neutral
8	This is the time to stand united. Not for complaining.	0	0	2	neutral

Table 6.2: Examples from TRAC dataset.

6.2 Sentiment and explicit hate in IHC

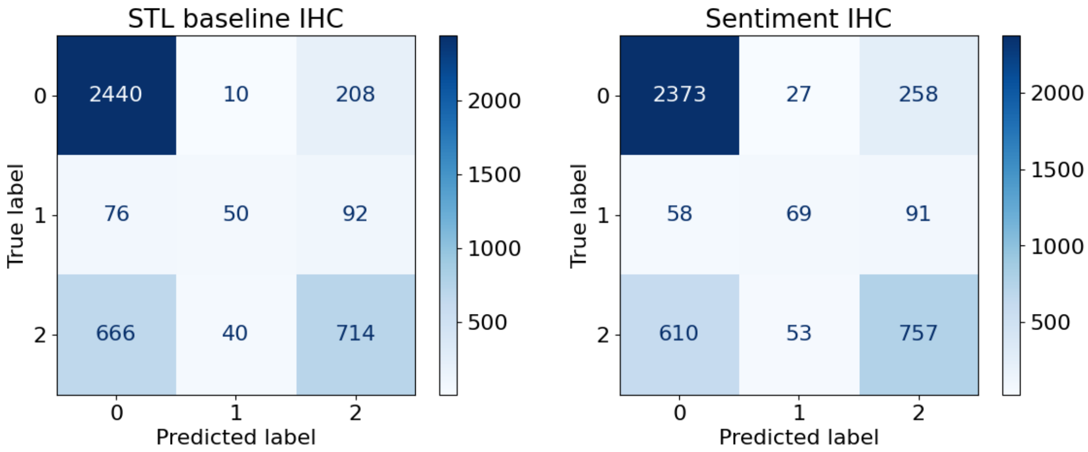


Figure 6.2: Confusion matrices of single task baseline and sentiment analysis multi-task model on IHC.

Utilizing sentiment analysis as an auxiliary task has a positive effect on IHC dataset. The confusion matrices of the baseline and sentiment model are shown in Figure 6.2. The number of detected instances of both explicit and implicit hate went up compared to the baseline. The baseline found 50 of the *explicit hate* messages and 714 *implicit messages*, the MTL-sentiment model found 69 and 757 instances of *explicit hate* and *implicit hate* respectively. Both of these class labels are predicted more often by the MTL model, thereby also decreasing the precision of these classes. The sentiment model associates *explicit hate* mostly with negative sentiment, as is evident from Table 6.3. 98% (146 out of 149) of the times it predicted*explicit hate*, it predicted a negative sentiment. The remaining 2% were labelled as neutral. The *not hate* label was assigned mostly along with neutral sentiment (around 63%). *Negative* and *positive* predictions co-occurred with *not hate* predictions around 30% and 7% respectively. This is in line with the expectation that explicit hate speech is negative in sentiment.

		Sentiment		
		Negative	Neutral	Positive
Aggression	0	902	1944	195
	1	146	3	0
	2	826	210	70

Table 6.3: Co-occurence of sentiment labels and hate speech labels predicted by MTL-sentiment on IHC.

The true positives will be looked at to see where the MTL-sentiment model improved the recall of explicit hate. The false negatives and false positives will be investigated to see where the model still struggles.

True positives There are 69 true positives for the MTL-sentiment model. Nine of these were not found by the baseline. All but one of these are labelled *negative* for sentiment. All nine texts are clear insults with a negative sentiment. Text 9-11 in Table 6.4 are examples of this.

False negatives False negatives are instances where the negative label (*not hate*) is predicted when it should have been labelled as *explicit hate.* So these are instances of explicit hate that the model missed. This happened 58 times. Eighteen tweets had clear insults and were labelled *negative* (see example 12). The rest of the tweets had more subtle insults and were labelled *neutral* or *positive* (example 13). It thus seems like the model picked up on the correlation between offensive words and negativity, but it still failed to label these as *explicit hate* as it should have.

False positives False positives are instances of appropriate tweets that have been predicted to be hateful. There are 27 false positives for *explicit hate.* All but one are labelled *negative* for sentiment. 12 of these false positives are actually insults, so they have been mislabelled by the annotators. Text 14 and 15 are examples of this. There is an explicit wish for the target to die in text 14 and text 15 includes ‘disgusting’ and ‘vile’, both clearly insulting words. The rest of the tweets allude to hateful acts, but they do not promote or condone them (see examples 16 and 17). It is not surprising that the model predicted these to be negative and hate speech because these violent acts are also mentioned in hateful tweets. The model seems to have learned an association between negative sentiment and hate speech, thus even predicting hate speech when there is none, but there is a negative tone.

	text	gold	STL baseline	MTL-sentiment	
				hate	sentiment
9	the punchable fat white liberal face	1	0	1	negative
10	Theresa May is a traitor and needs to be shot like a traitor	1	0	1	negative
11	you globalist scumbag. one of soro's puppets lowlife	1	0	1	negative
12	I don't need someone to explain to me how the Dems are outright traitors. What I need explained to me is how do we get these fuckers hanged.	1	0	0	negative
13	sure there is. i'm finnish and german. i'm white. what are you? a half breed?	1	0	0	neutral
14	go die u racist asswipes	0	1	1	negative
15	i'm sure you'd love to kill anyone who is non-white or jewish lol. disgusting vile human you are.	0	1	1	negative
16	how about white catholic priests raping hundreds of thousands of white christian boys?	0	0	1	negative
17	and it was white non muslims who massacred 6 million jews and crusaders who massacred jews	0	0	1	negative

Table 6.4: Examples from IHC dataset.

6.3 Emotion and explicit abuse in AbuseEval

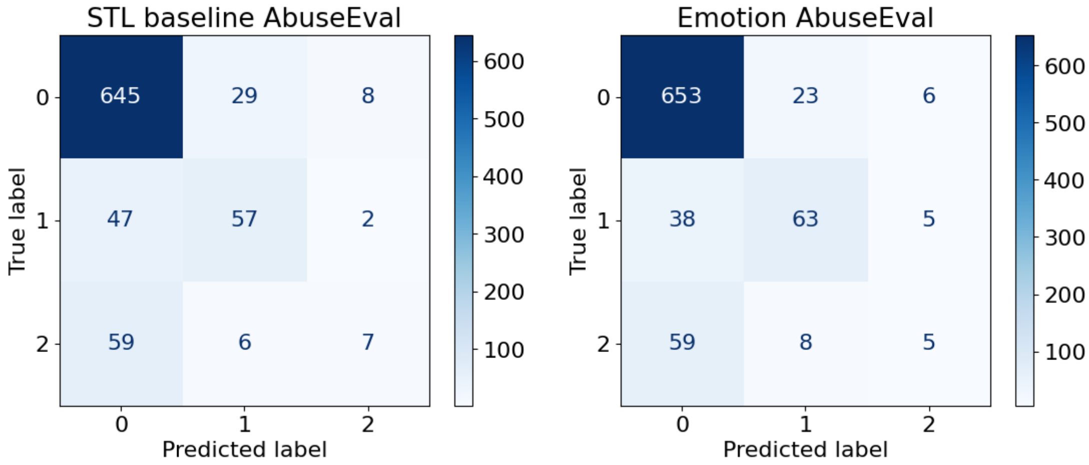


Figure 6.3: Confusion matrices of single task baseline and emotion detection multi-task model on AbuseEval.

The performance on AbuseEval was improved by emotion detection. This is due to better performance on *explicit abuse* class. Both the recall and precision improved: the number of found messages went from 57 in the baseline to 63 in the MTL-emotion model(see Figure 6.3. The MTL-emotion model also confused *not abuse* texts less often with *explicit abuse* than the baseline (47 to 38).

Table 6.5 shows how often the predicted labels occur on the same texts. *Explicit abuse* is mostly predicted together with *anger* and *disgust* (36 and 38 times). When *implicit abuse* is predicted, *disgust* is mostly predicted. Not abusive texts are predicted to be mostly *surprise*. *Joy* is also often predicted next to *not abuse*. *Joy* never co-occurs with *explicit abuse* and *implicit abuse*.

		Emotion					
		Anger	Disgust	Fear	Joy	Sadness	Surprise
Abuse	0	55	13	126	169	123	264
	1	36	38	10	0	9	1
	2	3	9	1	0	0	3

Table 6.5: Co-occurence of emotion labels and abuse labels predicted by MTL-emotion on AbuseEval.

True positives There is a small increase from 57 to 63 true positives from the baseline to MTL-emotion. 13 tweets were falsely labelled *not abuse* by the baseline and correctly labelled *explicit abuse* by the MTL-emotion model. Four of these are predicted to have *disgust* as emotion. Text 18 in Table 6.6 is an example of such a tweet. Anger and sadness also helped the model to find implicit abuse. However, it is not clear why the *sadness* label has been assigned to these tweets, as they do not convey sadness (see text 20).

False negatives The number of false negatives went from 47 in the baseline to 38 in the MTL-emotion model. The 38 false negatives are manually assessed to find out what type of mistakes the model makes. Three types of mistakes have been identified. 11 tweets contain only one bad word. On four of these, *anger* is correctly predicted, but they are still not found as *explicit hate speech*. See texts 21 and 22. On the other hand, two of these tweets have been labelled with positive emotion when that is not fitting (see example 23). The emotion detection task is thus not performing perfectly. The second type of mistake is due to the length of the tweets. The maximum text length is 40 wordpieces. For five tweets, the insults is at the end of the tweet and has been cut off. Example 24 shows one such instance. The original tweet and the tokenized version are shown. The insult 'a woman that gross' is not seen by the model. Ten tweets consist of criticism. There is a negative tone, but there is no explicit hate or abuse expressed. These have been mostly labelled as *fear* and *surprise*. Text 25 in an example of this.

False positives The number of false positives was reduced from 29 in the baseline to 23 in the MTL-emotion model. These tweets are mostly associated with *anger* (13 times) and *disgust* (6 times), and rightfully so. The model seems to have learned a link between anger and disgust and explicit hate speech. On one hand this is positive, as more abusive tweets were found. On the other hand, this means that angry or disgusted tweets that do not constitute as abusive language also get labelled as *abusive*. Three types of false positives have been identified. Six tweets are abusive and the gold labels are wrong. See examples 27 and 28. Five tweets include profane words such as 'fuck'. It is not surprising the model labelled these as *explicit abuse* as abusive language is often conveyed through profanity. See example 29. The rest of the tweets, such as example 30, have a negative or critical tone, much like some other tweets that are *explicit abuse* are.

6.4 Emotion and implicit hate in IHC

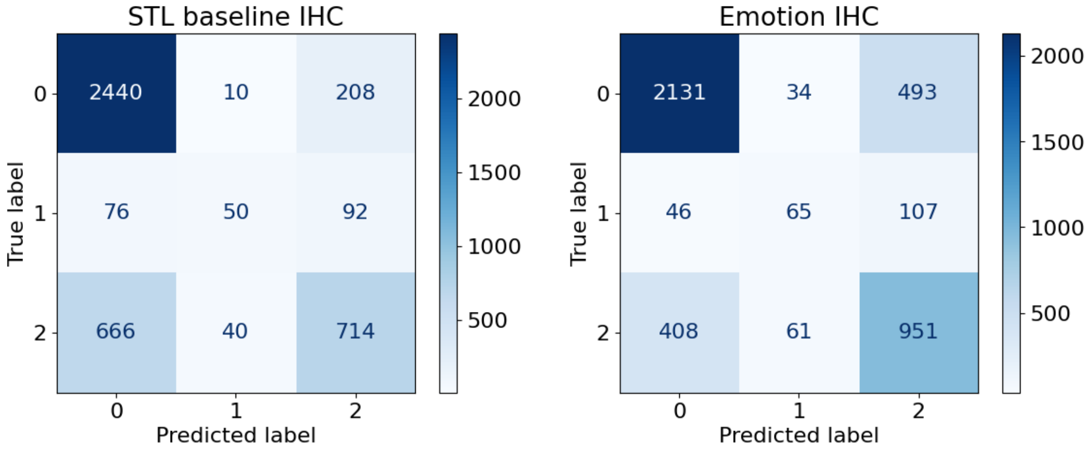


Figure 6.4: Confusion matrices of single task baseline and emotion detection multi-task model on IHC.

On IHC, the number of true explicit positives went from 50 to 69, as can be seen in Figure 6.4. The biggest improvement is observed in the *implicit hate* class. The number of true positives increased with 237, from 714 to 951. However, the MTL-emotion model also produced more false positives, decreasing the precision.

In the predictions by the MTL-emotion model for IHC *implicit hate* occurs most often with *fear* (788 times, see Table 6.7). *Disgust* co-occurs most often with *implicit hate* after *fear*. *Joy* is surprisingly often predicted when *implicit hate* is predicted (150 times). The analysis below will highlight some of these instances. *Anger* is least correlated with *implicit hate*. *Explicit hate* occurs most often with *disgust*. *Fear* and *surprise* are most often predicted when *not hate* is predicted.

True positives The number of true positives for *implicit hate* increased from 714 in the baseline to 951 in the MTL-emotion model. 286 tweets were labelled *not hate* by the baseline, but were correctly identified as *implicit hate* by the MTL-emotion model. Most of these are labelled as *fear* (160 tweets). The majority of these tweets mention violent acts or events such as war and attacks (see example 31 and 32 in Table 6.8), white power (example 33) and white pride (example 34). The *fear* label is expected for the tweets that mention violent acts, but are less intuitive for the white power and white pride tweets. The remaining tweets that do not fall into one of these categories are also not expressing fear. It is unclear why the model associates these tweets with *fear*.

The label *disgust* is assigned to 40 out of the 286 tweets the baseline missed but the MTL-emotion model caught and anger to twenty tweets. Similar to the fear label, the labels *disgust* and *anger* are assigned when acts like killing or raping are mentioned (example 35 and 36). There are a handful of instances where explicit disgust is expressed (see example 37 and 38). However, for 14 tweets there is no evidence of disgust in the tweets (for example in text 39). Some of the angry tweets also are not angry and seem more neutral. Interestingly, 25 tweets are labelled *joy*. Seven of these are praising the white race for its accomplishments and put them above other people. It is impressive the model found these tweets as *implicit hate*. Text 40 is an example of this. It is not clear why the rest are labelled *joy*. They do not express a positive emotion. The labels *sadness* and *surprise* also do not fit for many of the tweets (examples 41 and 42). We cannot conclude that there was a knowledge transfer from emotion to hate speech detection, because many of the predicted emotions are not accurate. Overall, it is not clear why the MTL-emotion model has improved the recall of *implicit hate*.

False negatives There are 408 tweets that the model predicted *not hate*, while they are *implicit hate*. Most are labelled *fear* again. The other negative emotions *anger* and *disgust* make up a smaller portion of the false negatives. The true positives labelled with these emotions often mention violent acts. This is less so the case for the false negatives. The tweets are more mild. So, even though a negative emotion was predicted, the hate was still too subtle to be caught by the model. 61 false negatives are labelled *joy*. These tweets range from actually joyous (example 43) to outright hateful (44). It is not surprising tweets labelled with *joy* are not identified as hate speech, because joy is not often associated with hate speech. Many of the false negatives have no surface level evidence of hate speech, such as examples 45 and 46.

False positives The number of false positives for *implicit hate* increased from 208 in the baseline to 493 in the MTL-emotion model. These tweets have been labelled *implicit hate* when the gold label is *not hate*. The majority is labelled as *fear*, just like the true positives and false negatives. The model has learned to associate *fear* with *implicit hate*. So even when there is no hate, when *fear* is predicted *implicit hate* is also predicted. Similar to the false positives in AbuseEval, the false positives here often describe violent events. They are not considered hate speech because they do not condone or promote it. See example 47.

6.5 Sarcasm and covert aggression in TRAC

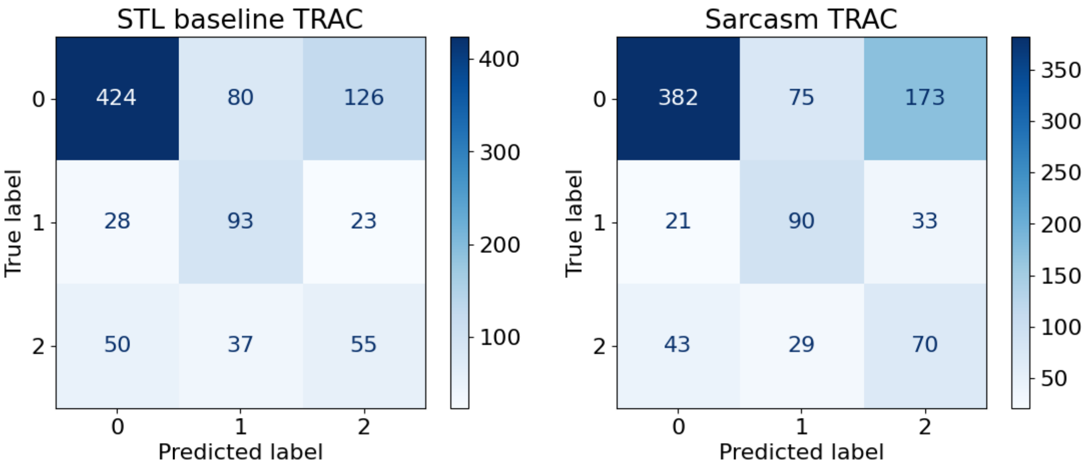


Figure 6.5: Confusion matrices of single task baseline and sarcasm detection multi-task model on TRAC.

The confusion matrices of the baseline and MTL-sarcasm model on TRAC are shown in Figure 6.5. Sarcasm detection had a positive effect on the recall of *covert aggression*, increasing the number of found messages from 55 to 70. However, the number of false positives for *covert aggression* also increased compared to the baseline (126 to 173). Overall, the MTL-sarcasm model did worse than the baseline. The MTL-sarcasm model predicted that 165 messages are sarcastic and 751 are not sarcastic, as can be inferred from Table 6.9. A little over a third of the messages labelled *overt aggression* are labelled with *sarcasm*. A quarter of the messages labelled *covert aggression* are labelled as being sarcastic. The messages predicted to be appropriate have the lowest proportion of sarcastic predictions. It is surprising that the overt label occurs more with *sarcasm* than the *covert* label, as covert aggression might be hidden with sarcasm. Overt aggression is clear from the surface form of the text, so these are unlikely to be sarcastic.

True positives The number of true positives for *covert aggression* went from 55 in the baseline to 70 in the multi-task model. Eighteen of those 70 messages were not detected by the baseline. Six are labelled *sarcasm*, twelve not. The tweets labelled *sarcasm* are not sarcastic, but some of them have a humorous tone and the aggression is very implicit. Texts 48 and 49 in Table 6.10 are examples of this. The aggression in all of the 18 tweets in incredibly hidden, if there at all, so it is not strange the baseline missed them. Knowledge about sarcasm does not seem to be the thing that makes the performance better than the baseline because the messages identified as sarcasm are not actually sarcastic.

False negatives There are 43 false negatives by the multi-task model. Three are labelled as sarcasm. One text can be considered sarcasm because of ‘ha ha ha’ (see example 50). These false negatives do not seem aggressive. When they do convey criticism it is very subtle and indirect. For example, text 51 is a question implying criticism and that criticism in itself is very mild.

False positives The number of false positives went up to 173. The baseline had 126 false positives. 41 messages are labelled as *sarcasm*, the rest as *not sarcasm.* None of them contain sarcastic statements. Eleven tweets contain an insult, so it is expected that these messages are labelled as being aggressive. Text 52 is an example of this. The rest of the texts do not have any evidence of abuse. Two topics stand out. 24 tweets are about the problem of black money in India (see example 53), and twelve tweets are about the speech by the former PM of India (see example 54).

6.6 Sarcasm and implicit hate in IHC

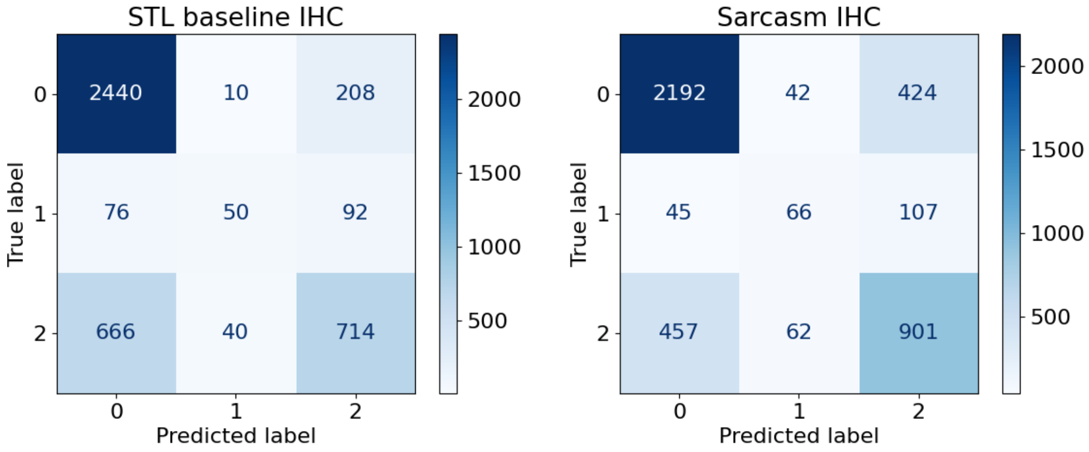


Figure 6.6: Confusion matrices of single task baseline and sarcasm detection multi-task model on IHC.

Figure 6.6 show the confusion matrices of the baseline and MTL-sarcasm on IHC. The multi-task model does better than the baseline on IHC, increasing the recall of both explicit (55 to 60 true positives) and implicit hate (714 to 901 true positives). The amount of detected sarcasm is different across the different predicted hate speech classes, as is clear from Table 6.11. *Explicit hate* is predicted 170 times, 140 of these tweets are labelled to not be sarcastic. The distribution of *sarcasm* and *not sarcasm* is roughly equal for the tweets predicted as *not hate*. Roughly two thirds of the tweets labelled *implicit hate* have been labelled as sarcasm. The implicit hate speech supposedly thus has a larger proportion of sarcastic tweets than the other two categories. This is to be expected, as explicit hate speech is direct and thus not sarcastic and implicit hate speech might be hidden in sarcasm.

True positives There are 901 true positives in total by the MTL-sarcasm model. 246 of those were not found by the baseline. Of those tweets, 164 are labelled *sarcasm*. Only four tweets are actually sarcastic (see texts 55-58 in Table 6.12). The other tweets are not sarcastic, even though they have been labelled so. The model has thus not learned to detect sarcasm accurately. The reason the MTL-sarcasm model is better than the baseline is therefore unclear.

False negatives The MTL-sarcasm model decreased the number of false negatives from 666 in the baseline to 457. 262 sarcasm. Seven tweets do in fact have a sarcastic tone (texts 59-62). These tweets have the same kind of sarcasm as texts 10-13, e.g. using a fake surprise (‘what a surprise’ and ’shocker’) and the phrase ’how dare’. However, even though these tweets were correctly identified as sarcasm, they still are not labelled *implicit hate*. The rest of the tweets labelled as sarcasm are not sarcastic. It is unclear why they have been labelled so. 195 tweets were labelled as ‘not sarcasm’. There is no clear difference between the tweets labelled as sarcasm and not sarcasm. Broadly, the false negatives can be divided into two groups: tweets that are understandably misclassified, and tweets that should have been found. Tweets that are understandably misclassified include tweets that are not hate speech at all (13 times, see example ...) and tweets that do not contain any sort of surface negativity (199 tweets). Fourteen of these are even very positive tweets praising white people. These tweets imply that other races are inferior to white, but this is left unsaid. See example 63. The majority of the false negatives do contain surface level evidence of implicit hate speech. 122 tweets mention hate groups such as Black Separatist and White Nationalist. 71 tweets explicitly mention violence such as rape and murder (see example 64), 23 tweets contain words like 'hate' and 'disdain' (example 65) and lastly, 29 tweets contain outright insults (see example 66). The model did not pick up on these, but should have.

False positives There are 424 tweets that are not hateful, but the MTL-sarcasm model has labelled them as *implicit hate.* None of the tweets are sarcastic. Nonetheless, 270 out of the 424 tweets are labelled as 'sarcasm'. There are five types of false positives. The first type are tweets that contain insulting language and should have been labelled as hate speech. There are six such tweets and text 18 is an example of this. 90 tweets contain references to known hate groups, that are often mentioned in the implicit hate tweets. 84 tweets contain references to violence and 23 tweets contain words that express hatred or contempt. These types of tweets were also observed in the false negatives. It is not clear what the model has learned as hate speech exactly, as violence, insults and hate groups are sometimes correctly found (true positives), missed (false negatives) and sometimes wrongly classified as *implicit hate.*

6.7 Irony and implicit abuse in AbuseEval

On AbuseEval, the highest overall score was obtained by the model trained on irony detection. The recall of *implicit abuse* increased from seven in the baseline to ten in the multi-task model. With this, the number of false predictions for *implicit abuse* also went up. *Not abuse* is especially often confused for *implicit abuse* (16 times compared to 8 in the baseline, see Figure 6.7). The number of detected instances of *explicit abuse* decreased from 57 to 55.

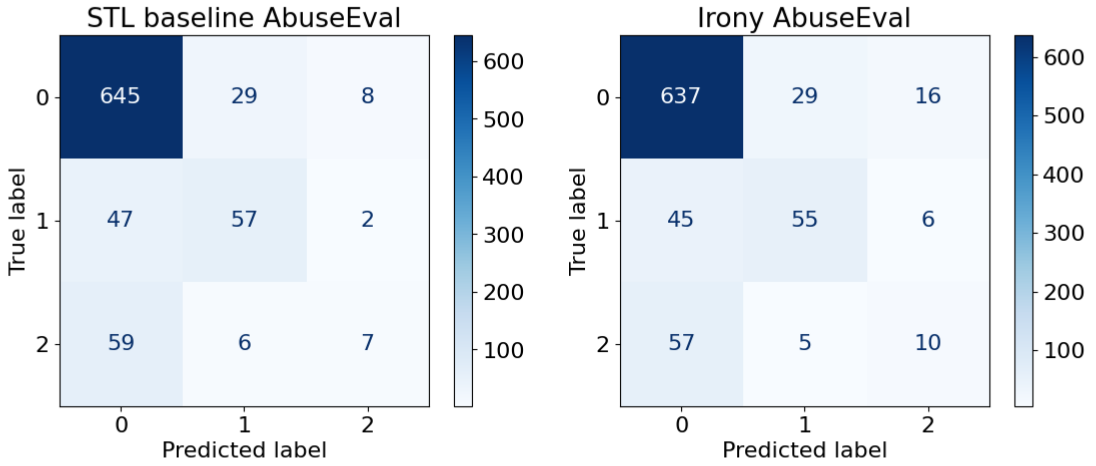


Figure 6.7: Confusion matrices of single task baseline and irony detection multi-task model on AbuseEval.

Table 6.13 shows the co-occurrence of the hate speech labels and irony labels predicted by the MTL-irony model trained for AbuseEval. The majority of tweets labelled as *not abuse* and *explicit abuse* have been labelled as 'not irony'. In contrast, just over half of the tweets labelled *implicit abuse* have been labelled as being ironic in nature. Just like with sarcasm, a higher number of ironic tweets is expected in the *implicit abuse* class. It is not surprising that the model has learned to associate irony most with *implicit abuse.*

True positives There are ten true positives. Six of those are labelled *irony.* Only one of them contains ironic statements ('cool', 'that's awesome' in example 67 in Table 6.14). Six of the true positives were missed by the baseline. All of them are different and have different insults (examples 67-72) It is unclear what the MTL-irony model has learned differently than the baseline.

False negatives The MTL-irony model has 57 false negatives. They can be divided into three groups. The first group of false negatives have some surface-level evidence of negativity and insultiing language (30 tweets). However, this negativity is mild and often refers to specific actions or events. Text 73 is an example of this. The second group of errors does not have any negative words, the insult is implied. 23 tweets are of this type. See example 74. The last group is ironic statements. There are four ironic false negatives. All of them have been correctly identified as irony, but still were not found as *implicit abuse.* Text 75 is an example of irony: 'nice work'.

False positives There are sixteen false positives by the MTL-irony model, where *implicit abuse* is predicted, when the gold label is *not abuse.* Nine are labelled n*ot irony* and seven are labelled *irony.* None of them are ironic. Of the sixteen false positives, four tweets have explicit language (see example 76) and three tweets contain words like 'terrorism' and 'racist' which often occur in abusive tweets in this dataset as accusations. Text 77 is an example of this. The remaining nine tweets talk about gun control and politics, but are not abusive. Such tweets are prevalent in this dataset and occur in all three classes.

The MTL-irony model is the best model out of all AbuseEval models. However, it is unclear why this is the case. It does not seem to be the irony detection task because the prediction on only one ironic tweet was improved, and many of the tweets predicted to be ironic are in fact not ironic.

6.8 Irony and implicit hate in IHC

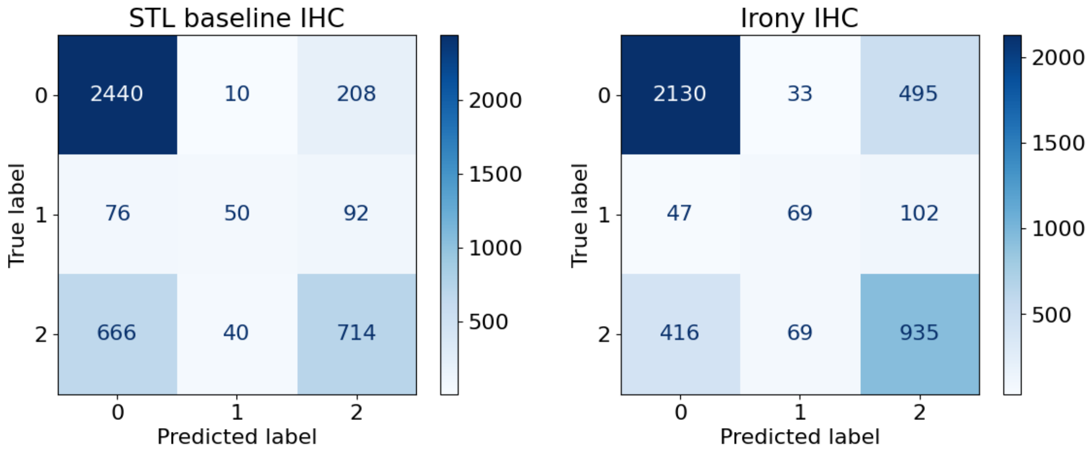


Figure 6.8: Confusion matrices of single task baseline and irony detection multi-task model on IHC.

Figure 6.7 shows the confusion matrices for the baseline and MTL-irony model on IHC. Especially the performance on *implicit hate* was increased. The baseline found 714 of these, while the multi-task model found 935. The recall of *explicit hate* also improved, going from 50 to 69. On IHC, just over half of the tweets are labelled as *irony* (see Table 6.15. *Irony* is predicted most often on tweets labelled *not hate*. Tweets labelled as *explicit hate* are also more often predicted to be ironic. This is unexpected, as explicit hate is direct and thus cannot be ironic. For tweets labelled as *implicit hate*, the *not irony* label is more often predicted.

True positives There are 935 true positives by the MTL-irony model. 272 of those were missed by the baseline. 97 are labelled *irony*. There are ten instances of actual irony. The MTL-irony model had thus improved the performance on ironic tweets. See examples 78 and 79 in Table 6.16. The remaining tweets are not ironic.

False negatives There are 416 false negatives in the MTL-irony model. This is a decrease from 666 false negatives by the baseline. There are five instances of irony. All but one have been correctly identified as irony by the model. However, they still are not identified as *implicit hate*. The remaining false negatives have the same error types as identified in the sarsasm detection model above.

False positives The MTL-irony model has 495 false positive predictions for the *implicit hate* class. This is a high increase from 208 false positives in the baseline. Most of these tweets have been labelled as *not irony* (322 tweets). In the remaining 173 tweets irony was detected. There are two instances of verbal irony here (see text 80 and 81). Again, the same mistakes were made as by the MTL-sarcasm model.

6.9 Summary and discussion

The manual analysis suggests that knowledge transferred from sentiment analysis to hate speech detection. The models learned to associate covert aggression with neutral sentiment and explicit hate with negative sentiment. Covert aggression is subtle and indirect compared to overt aggression. A neutral, as opposed to negative, sentiment is expected. The MTL-sentiment model classified many tweets as neutral, and therefore classified many tweets as covert aggression. This increased the recall of the *covert aggression* class. The high co-occurrence of *explicit hate* and negative sentiment is also expected. Above, we have showed examples of the MTL models improving upon the STL models by identifying texts as hate speech that have these neutral and negative sentiments. This evidence supports the claim that there was knowledge transfer from sentiment analysis to hate speech detection. The same observations regarding negative sentiment and hate speech were made by Plaza-Del-Arco et al. (2021).

There is some evidence for knowledge transfer from emotion detection to hate speech detection. Especially the identification of fear, disgust and anger seemed to help detect more hate speech. This is in line with the findings of Rajamanickam et al. (2020), Plaza-del Arco et al. (2021) and Plaza-Del-Arco et al. (2021) who also showed examples of fear, disgust and anger being helpful for hateful texts. However, the manual analysis also revealed that the emotions predicted are not always accurate. This undermines the claim that there was a knowledge transfer, because the model seems to not have precise knowledge of emotion. The main reason for the incorrect emotion predictions is the fact that the model can only predict emotions, not the absence of emotions. All the texts in the three test sets are thus predicted to express one of six emotions, even when they are neutral. Besides that, it is not unexpected that emotion detection performs worse than sentiment analysis. Emotion detection is more complex and is usually more difficult to learn than sentiment analysis (Nandwani and Verma, 2021).

The analysis found some evidence of knowledge transfer from sarcasm detection to hate speech detection on IHC, but none for TRAC. *Covert aggression* was associated more with *not sarcasm*, while *implicit hate* occurred more often with *sarcasm.* There is some improvement on sarcastic texts, but overall it is not clear what the influence of sarcasm detection is.

There is also some evidence of knowledge transfer from irony detection to hate speech detection. The manual analysis pointed out that hate speech predictions on tweets with ironic tone did improve. However, like sarcasm, some were still misclassified. Also similar to sarcasm detection, *irony* was predicted more than is correct. The most likely reason sarcasm and irony detection were not learned as well as sentiment analysis (and emotion detection) is the distribution of the training data. The datasets contain 50% sarcasm and irony, while only a small number of texts in the hate speech test sets are sarcastic or ironic. Therefore, too many texts were predicted to be sarcastic and ironic.

Even though the evidence presented in the manual analysis supports some transfer of knowledge, it is difficult to say whether the higher performance is actually due to knowledge transfer. The BERT models trained in this thesis have many random elements that influence what optimum is reached, such as random dropout and the initialization of parameters. Sellam et al. (2021) have shown that the same model architecture trained with different random seed produces different predictions. The difference in predictions by the multi-task models and the single task models might therefore in part be due to the random aspects in the training procedure. Moreover, BERT is a deep learning model and the decisions it makes cannot be traced easily. This is called the black box problem (von Eschenbach, 2021). Without probing of the parameters, such as was done in (Kapil and Ekbal, 2020), we cannot say what is happening inside the model that leads it to its predictions.

	text	gold	STL baseline	MTL-emotion	
				hate	emotion
18	#DespicableDems lie again about rifles. Dem Distorted the Law to Push Gun-Control at Kavanaugh Confirmation URL via @USER	1	0	1	disgust
19	All these sick ass ppl from school gave me something and now I have to chug down this nasty drink so it can go away	1	0	1	anger
20	@USER #Gutierrez has always been nothing more than a #racebaiter	1	0	1	sadness
21	#StopEtchecopar? Fuck you all Que florezcan mil Baby Etchecopar	1	0	0	anger
22	#Bakersfield is why we need gun control! Sorry for the victims other than the gunman himself I hope he rots in hell!	1	1	0	anger
23	@USER #Rosie makes me nauseous! URL	1	0	0	joy
24	@USER @USER Put DeLauro in a police lineup identifying the bag lady" - she would be picked everytime! She has to be proof either Conn voters are incompetent **to vote or she is part of a vast voter fraud conspiracy! No one votes for a woman that gross!"**	1	0	0	surprise
25	#BlaseyFord It's so comical that liberals always want it both ways first women need to be treated as equals but when it comes to Ford testifying they say it's unfair to have more men asking her questions. Which is it Dems are women equal or should we receive special treatment?	1	0	0	surprise
26	...what about death _metal_? I want more death METAL... Tell me, you hateful Keebler Elf, WHERE CAN I HAVE MORE DEATH METAL???? URL	0	1	1	anger
27	@USER @USER @USER She is just nasty	0	1	1	disgust
28	@USER nigga are you stupid your trash dont play with him play with your bitch	0	1	1	anger
29	Alright let me get right with God bc Mother Nature is like fuck humans URL	0	1	1	anger
30	#FauxProgressives I've been done with these #CorruptDemocrats for years now and here's more proof!	0	1	1	anger

Table 6.6: Examples from AbuseEval dataset.

		Emotion					
		Anger	Disgust	Fear	Joy	Sadness	Surprise
Hate speech	0	162	283	823	373	364	580
	1	18	112	24	4	1	1
	2	118	237	788	150	130	128

Table 6.7: Co-occurrence of emotion labels and hate speech labels predicted by MTL-emotion on IHC.

	text	gold	STL baseline	MTL-emotion	
				hate	emotion
31	u white people won a battle but your going to lose the war .	2	0	2	fear
32	muslims attack - a common theme around the world ! !	2	0	2	fear
33	you guys seem really cool . how can i join your white power party ?	2	0	2	fear
34	white pride worldwide hooded sweatshirt — more : via	2	0	2	fear
35	funny how you never hear about white men raping women in black neighborhoods . i know 2 white women who ’	2	0	2	disgust
36	where were the protests in milwaukee when five blacks were killed just prior to the young man who the cops killed ?	2	0	2	disgust
37	it honestly makes me sick to think about what the white liberals have done to the black family in america . they deserve to be forced out	2	0	2	disgust
38	this is disgraceful makes me sick that it white on black crime that’s mostly reported ! black on white happens more t repoted	2	0	2	disgust
39	official : 75 % of berlin criminals are arabs sw rm hmm via	2	0	2	disgust
40	the white race has an amazing history filled with incredible accomplishments . through our achievements the whole world has benefitted .	2	0	2	joy
41	it’s ok to be white	2	0	2	sadness
42	if you are a person of european non-amerindian heritage send us an application .	2	0	2	surprise
43	the amazing white race	2	0	0	joy
44	everywhere y’all touch equal hell,disease & death..	2	2	0	joy
45	the french are not white ? :d have you ever seen a french ? !	2	0	0	surprise
46	where the white women at ? via	2	0	0	surprise
47	charlottesville wasn’t about blacks . the main fighting was white on white .	0	0	2	fear

Table 6.8: Examples from IHC dataset.

		Sarcasm	
		Sarcasm	Not sarcasm
Aggression	0	26	420
	1	70	124
	2	69	207

Table 6.9: Co-occurrence of sarcasm labels and aggression labels predicted by MTL-sarcasm on TRAC.

	text	gold	STL baseline	MTL-sarcasm	
				hate	sarcasm
48	Quick .....tell them the size of Mr Modi ji's chest	2	0	2	sarcasm
49	Ha ha ha ha mr. singh able to speek	2	0	2	sarcasm
50	Modi is great..But he did not gave enough time to politicians to convert their black money..All politicians want black money to be recovered but from public not theirs..Ha ha ha..	2	0	0	sarcasm
51	Why was The Economist correspondent barred from attending the Governor's press conference.?	2	2	0	not_sarcasm
51	It means that coward attack was planned by RAW... loosers	0	2	2	not_sarcasm
51	All the black money got white in 50 days Dont u think the policy failed.	0	2	2	not_sarcasm
51	And the literate has spoken..... such poise and well informed Man- our former PM. And kept dignity throughout. True Indian!	0	0	2	not_sarcasm

Table 6.10: Examples from TRAC dataset.

		Sarcasm	
		Sarcasm	Not sarcasm
Hate speech	0	1318	1376
	1	140	30
	2	981	451

Table 6.11: Co-occurrence of sarcasm labels and hate speech labels predicted by MTL-sarcasm on IHC.

	text	gold	STL baseline	MTL-sarcasm	
				hate	sarcasm
55	how dare white people want to exist as a race ! that is racism unless it's a non-white race wanting to think tribally ! those bigots !	2	0	2	sarcasm
56	black lives matter and again . black on white crime ! what a surprise ...	2	0	2	sarcasm
57	cool so when black people kill each other you can't blame the white man !	2	0	2	sarcasm
58	i assume it's about how the africans freed themselves from evil whites owners . with . no . help . at . all .	2	0	2	not_sarcasm
59	oh look the anti-gay anti-woman is in bed with white supremacists . shocker .	2	0	0	sarcasm
60	how dare people of european decent want their civilisation to live on & flourish ! ! so unreasonable !	2	0	0	sarcasm
61	" how dare you be white in public ! reeeeee ! " " good for you ! "	2	0	0	sarcasm
62	How dare I suggest boys' suffering needs attention too	2	0	0	sarcasm
63	white power fam .	0	2	2	sarcasm
64	and white cops kill our black men every day .	0	2	2	sarcasm
65	what fuels your disdain for the jewish people ?	0	0	2	not_sarcasm
66	only to white uneducated stupid people think its not about race .	0	0	2	sarcasm

Table 6.12: Examples from IHC dataset.

		Irony	
		Irony	Not irony
Abuse	0	165	574
	1	22	67
	2	17	15

Table 6.13: Co-occurrence of irony labels and abuse labels predicted by MTL-irony on AbuseEval.

	text	gold	STL baseline	MTL-irony	
				hate	irony
67	28, 27, 25 and 21 but like,, it's still really miserable and unpleasant for us?? And like they even told me how they weren't happy and would have got divorced before I was even born so I'm like cool cool I was literally born into hatred that's awesome no wonder I'm such a fuckup	2	0	2	not_irony
68	@USER @USER @USER Okay and? Anybody can go to Eredivise and score 15+ goals. Shit league. Bundesliga teams are competitive in euro competitions. Mexico hasn't produced any talent like Pulisic in a long time. It's okay, maybe you guys will one day.	2	0	2	irony
69	#Democrats #Liberals you are being #threatened by #Armed #Nazis and you talk about #disarmament of yourselves and of #Disconnection from the #Military #Lifestyle ? You should #broadcast your #Armory and #killSkills #RightFuckingNOW URL	2	0	2	irony
70	#ConstitutionDay It's very odd for the alt right conservatives to say that we are ruining the constitution just because we want #GunControlNow but they are the ones ruining the constitution getting upset because foreigners are coming to this land who are not White wanting to live	2	0	2	irony
71	@USER @USER Who the hell does he think he is?	2	0	2	not_irony
72	@USER @USER You do not like the Jewish people because you ALL claims they are holding all the money. Sound familiar? Your left wing libtard family is the family of the KKK. You do not hate Jews? That's funny. Gun control; even funnier. Good luck	2	0	2	irony
73	#BelieveAllWomen just when I think things can't get dumber.. this happens.. liberals probably don't realize this because they are children but this isn't equality.	2	0	0	not_irony
74	#auspol I don't know why he is still in his job. Seriously. URL	2	0	0	not_irony
75	. a grown ass woman, probably 10 years older than me is currently spreading rumors rather than talking to me about it, nice work you got there satan	2	0	0	irony
76	@USER I BET she is a lot safer when she has sex now tho ? Or she wouldn't have just one but a few kids . Why cant moms get dicked down too ?	0	2	2	irony
77	#StopTheBias #RedCensorship My account was suspended for apparently promoting Hate Speech" , yet #Antifa Terrorist Organizations are actively up and running. Since your Customer Service refuses to reply to my emails, care to explain how this makes sense? @USER URL	0	0	2	irony

Table 6.14: Examples from AbuseEval dataset.

		Irony	
		Irony	Not irony
Hate speech	0	1659	934
	1	91	80
	2	484	1048

Table 6.15: Co-occurrence of irony labels and hate speech labels predicted by MTL-irony on IHC.

	text	gold	STL baseline	MTL-irony	
				hate	irony
78	oh look the anti-gay anti-woman is in bed with white supremacists . shocker .	2	0	2	irony
79	” hey kids wanna replace white people ” ” ”	2	0	2	irony
80	its not racism - how dare a white person get on a bus - what did rosa parks die for ?	0	0	2	irony
81	of course ! how dare he try to escape from their slavery for the democrats !	0	0	2	irony
82	the white teach did the smart thing for once	0	0	2	irony

Table 6.16: Examples from IHC dataset.

Chapter 7

Conclusion

7.1 Summary

This thesis aimed to explore multi-task fine-tuning on BERT for implicit and explicit hate speech detection. In multi-task learning various tasks are learned by one model, at the same time. This was done by fine-tuning BERT for two or three tasks at a time. The BERT encoder is shared among all the tasks and thus learns representations from all tasks.

Two different approaches to multi-task learning were taken. The first approach is using auxiliary tasks - sentiment analysis, emotion detection, sarcasm detection and irony detection - with the aim to leverage information of these tasks for hate speech detection. We tested the effect of these auxiliary tasks on AbuseEval, TRAC and IHC, which are all datasets annotated for explicit and implicit hate speech. We found that all of the tasks had at least some positive effect on either the recall or precision of implicit or explicit hate speech. Through a manual analysis of the predictions of the multi-task models we found that sentiment and emotion information helped the models identify hate speech better than the baseline without this information. Sarcasm and irony detection also increased the performance of the models, especially that of the implicit class. There is some evidence of improved predictions on sarcastic and ironic texts. However, the models still failed to identify a number of sarcastic and ironic tweets that express hate speech and generally were not accurate predicting sarcasm and irony.

The second way in which we tested the potential of multi-task fine-tuning is by learning from multiple datasets as if they are different tasks. This way the model can learn from more diverse data and a bigger amount of data. We found that this approach only increased the performance on IHC. This suggests that the datasets might be too different and the model was unable to learn accurate representations from it.

7.2 Limitations and future work

We have identified four major limitations of this research and suggest future research based on each of them.

This thesis used existing datasets to train for the auxiliary tasks. However, the manual analysis showed that emotion detection was not accurate due to the lack of a *neutral* or *no emotion* label. Moreover, sarcasm and irony detection were inaccurate because of the large amount of sarcasm and irony in the training data and the small amount of sarcasm and irony in the test data. The detection of emotion, sarcasm and irony might be improved when the training data better reflects the test data. n future research, this should be taken into account.

The second limitation of this study is the lack of optimization. All experiments were conducted with the same hyperparameters. Higher performance might be achieved when the hyper-parameters are tuned for each individual experiment. Moreover, in this thesis all tasks in multi-task setups were treated as equal. However, the interest in the performance of the tasks is not equal. We are more interested in the performance of hate speech detection than any of the auxiliary tasks. A bias could be introduced through a weighing of the cross entropy loss to make one task's loss count more than the others'.

Another interesting direction for future research would be to focus on the multi-task model's ability to generalize across different types of hate speech. In this thesis, we trained a multi-task model on three different hate speech datasets in order to improve the performance on the test sets of these datasets. Waseem et al. (2018) conducted similar experiments with the aim to improve generalizability. To test how well a model generalizes to other data is usually done through cross-domain testing, where the model is trained on one dataset and tested on a dataset with a different distribution. A way to test the generalizability of the multi-task approach would be to train on two of the three datasets, and test on the third.

The analysis of the effect the auxiliary tasks had on hate speech detection focused only on the predictions made by the models. From this, we inferred whether a transfer of knowledge happened. However, we did not investigate the shared representations the models learned. Future research into what happens in multi-task learning can be done by taking layers of these representations and conducting experiments with them to find out what knowledge they represent, as was done in (Kapil and Ekbal, 2020).

Bibliography

F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*, 2(7):e12189, 2020.

P. Badjatiya, S. Gupta, M. Gupta, and V. Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*. ACM Press, 2017. doi: 10.1145/3041021.3054223. URL `https://doi.org/10.1145%2F3041021.3054223`.

V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, and M. Sanguinetti. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2007. URL `https://aclanthology.org/S19-2007`.

S. Bauman, R. B. Toomey, and J. L. Walker. Associations among bullying, cyberbullying, and suicide in high school students. *Journal of adolescence*, 36(2):341–350, 2013.

A. Brown. What is hate speech? part 1: The myth of hate. *Law and Philosophy*, 36: 419–468, 2017.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. doi: 10.1023/a:1007379606734. URL `https://doi.org/10.1023/a:1007379606734`.

T. Caselli, V. Basile, J. Mitrović, I. Kartoziya, and M. Granitzer. I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6193–6202, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.760`.

T. Caselli, A. Schelhaas, M. Weultjes, F. Leistra, H. van der Veen, G. Timmerman, and M. Nissim. Dalc: the dutch abusive language corpus. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH)*, online, Aug. 2021. Association for Computational Linguistics.

D. S. Chauhan, S. Dhanush, A. Ekbal, and P. Bhattacharyya. All-in-one: A deep attentive multi-task learning framework for humour, sarcasm, offensive, motivation, and sentiment on memes. In *Proceedings of the 1st conference of the Asia-Pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing*, pages 281–290, 2020.

T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515, 2017.

W. De Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*, 2019.

P. Delobelle, T. Winters, and B. Berendt. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.292. URL `https://aclanthology.org/2020.findings-emnlp.292`.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

T. Dreisbach. How extremists weaponize irony to spread hate, 2021. URL `https://www.npr.org/2021/04/26/990274685/how-extremists-weaponize-irony-to-spread-hate`.

P. Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

M. ElSherief, C. Ziems, D. Muchlinski, V. Anupindi, J. Seybolt, M. De Choudhury, and D. Yang. Latent hatred: A benchmark for understanding implicit hate speech. *arXiv preprint arXiv:2109.05322*, 2021.

B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1169. URL `https://aclanthology.org/D17-1169`.

P. Fortuna and S. Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018.

D. Ghosh, A. R. Fabbri, and S. Muresan. Sarcasm analysis using conversation context. *Computational Linguistics*, 44(4):755–792, 2018.

D. Ghosh, A. Vajpayee, and S. Muresan. A report on the 2020 sarcasm detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 1–11, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.figlang-1.1. URL `https://aclanthology.org/2020.figlang-1.1`.

P. Kapil and A. Ekbal. A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, 210:106458, 2020.

M. Khodak, N. Saunshi, and K. Vodrahalli. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://aclanthology.org/L18-1102`.

D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*, 2015.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari. Aggression-annotated corpus of Hindi-English code-mixed data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://aclanthology.org/L18-1226`.

J. Lemmens, I. Markov, and W. Daelemans. Improving hate speech type and target detection with hateful metaphor features. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 7–16, 2021.

S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152, 2019.

K. Maity, P. Jha, S. Saha, and P. Bhattacharyya. A multitask framework for sentiment, emotion and sarcasm aware cyberbullying detection from multi-modal code-mixed memes. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1739–1749, 2022.

S. Malmasi and M. Zampieri. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202, 2018.

T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, and A. Patel. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th forum for information retrieval evaluation*, pages 14–17, 2019.

I. Markov, N. Ljubešić, D. Fišer, and W. Daelemans. Exploring stylometric and emotion-based features for multilingual cross-domain hate speech detection. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 149–159, 2021.

R. Martins, M. Gomes, J. J. Almeida, P. Novais, and P. Henriques. Hate speech classification in social media using emotional analysis. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 61–66. IEEE, 2018.

Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, June 1947. doi: 10.1007/bf02295996. URL `https://doi.org/10.1007/bf02295996`.

W. Medhat, A. Hassan, and H. Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

R. S. Michalski. A theory and methodology of inductive learning. In *Machine learning*, pages 83–134. Elsevier, 1983.

S. Mohammad. # emotional tweets. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, 2012.

M. Mozafari, R. Farahbakhsh, and N. Crespi. A bert-based transfer learning approach for hate speech detection in online social media. In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*, pages 928–940. Springer, 2020.

V. Nahar, S. Al-Maskari, X. Li, and C. Pang. Semi-supervised learning for cyberbullying detection in social networks. In *Databases Theory and Applications: 25th Australasian Database Conference, ADC 2014, Brisbane, QLD, Australia, July 14-16, 2014. Proceedings 25*, pages 160–171. Springer, 2014.

P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1001. URL `https://aclanthology.org/S16-1001`.

P. Nandwani and R. Verma. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11(1):81, 2021.

J. T. Nockleby. *Hate Speech*, pages 1277–1279. Macmillan, Detroit, MI, 2000.

G. T. Patrick. The psychology of profanity. *Psychological Review*, 8(2):113, 1901.

J. Pavlopoulos, J. Sorensen, L. Laugier, and I. Androutsopoulos. Semeval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*, pages 59–69, 2021.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

G. K. Pitsilis, H. Ramampiaro, and H. Langseth. Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48:4730–4742, 2018.

F. M. Plaza-del Arco, S. Halat, S. Padó, and R. Klinger. Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language. *arXiv preprint arXiv:2109.10255*, 2021.

F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia. A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9:112478–112489, 2021.

R. Plutchik. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier, 1980.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

S. Rajamanickam, P. Mishra, H. Yannakoudakis, and E. Shutova. Joint modelling of emotion and abusive language detection. *arXiv preprint arXiv:2005.14028*, 2020.

J. Risch and R. Krestel. Bagging bert models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, 2020.

J. Risch, A. Stoll, M. Ziegele, and R. Krestel. hpidedis at germeval 2019: Offensive language identification using a german bert model. In *KONVENS*, 2019.

B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*, 2017.

S. Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

M. Safi. Sri lanka accuses facebook over hate speech after deadly riots. *The Guardian*, Mar 2018. URL https://www.theguardian.com/world/2018/mar/14/facebook-accused-by-sri-lanka-of-failing-to-control-hate-speech.

K. Saha, E. Chandrasekharan, and M. De Choudhury. Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of the 10th ACM conference on web science*, pages 255–264, 2019.

N. S. Samghabadi, A. Hatami, M. Shafaei, S. Kar, and T. Solorio. Attending the emotions to detect online abusive language. *arXiv preprint arXiv:1909.03100*, 2019.

S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright. Sarcasm detection using machine learning algorithms in twitter: A systematic review. *International Journal of Market Research*, 62(5):578–598, 2020.

A. Schmidt and M. Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain, Apr. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1101. URL https://aclanthology.org/W17-1101.

T. Sellam, S. Yadlowsky, J. Wei, N. Saphra, A. D'Amour, T. Linzen, J. Bastings, I. Turc, J. Eisenstein, D. Das, et al. The multiberts: Bert reproductions for robustness analysis. *arXiv preprint arXiv:2106.16163*, 2021.

S. Stecklow. Why facebook is losing the war on hate speech in myanmar. *Reuters*, Aug 2018.

C. Sun, X. Qiu, Y. Xu, and X. Huang. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer, 2019.

I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL `https://aclanthology.org/P19-1452`.

L. Torrey and J. Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

C. Van Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. De Pauw, W. Daelemans, and V. Hoste. Detection and fine-grained classification of cyberbullying events. In *Proceedings of the international conference recent advances in natural language processing*, pages 672–680, 2015.

C. Van Hee, E. Lefever, and V. Hoste. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

B. Vidgen, A. Harris, D. Nguyen, R. Tromble, S. Hale, and H. Margetts. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3509. URL `https://aclanthology.org/W19-3509`.

W. J. von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34(4):1607–1622, 2021.

S. Wachs, M. Gámez-Guadix, and M. F. Wright. Online hate speech victimization and depressive symptoms among adolescents: the protective role of resilience. *Cyberpsychology, Behavior, and Social Networking*, 25(7):416–423, 2022.

W. Warner and J. Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, Montréal, Canada, June 2012. Association for Computational Linguistics. URL `https://aclanthology.org/W12-2103`.

Z. Waseem. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142, 2016.

Z. Waseem and D. Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-2013. URL `https://aclanthology.org/N16-2013`.

Z. Waseem, T. Davidson, D. Warmsley, and I. Weber. Understanding abuse: A typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899*, 2017.

Z. Waseem, J. Thorne, and J. Bingel. Bridging the gaps: Multi task learning for domain transfer of hate speech detection. *Online harassment*, pages 29–55, 2018.

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.

D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, 2(0): 1–7, 2009.

M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1144. URL `https://aclanthology.org/N19-1144`.

M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/S19-2010. URL `https://aclanthology.org/S19-2010`.

M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and Ç. Çöltekin. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online), Dec. 2020. International Committee for Computational Linguistics. doi: 10.18653/v1/2020.semeval-1.188. URL `https://aclanthology.org/2020.semeval-1.188`.

Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

Z. Zhang, D. Robinson, and J. Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 745–760. Springer, 2018.

O. Ștefăniță and D.-M. Buf. Hate speech in social media and its effects on the lgbt community: A review of the current research. *Romanian Journal of Communication and Public Relations*, 23(1):47–55, 2021.