October 2022

Data Scarcity in Event Analysis and Abusive Language Detection

Sheikh Muhammad Sarwar
University of Massachusetts Amherst

DATA SCARCITY IN EVENT ANALYSIS AND ABUSIVE LANGUAGE DETECTION

A Dissertation Presented

by

SHEIKH MUHAMMAD SARWAR

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2022

Manning College of Information and Computer Sciences
University of Massachusetts Amherst

DATA SCARCITY IN EVENT ANALYSIS AND ABUSIVE LANGUAGE DETECTION

A Dissertation Presented

by

SHEIKH MUHAMMAD SARWAR

Approved as to style and content by:

James Allan, Chair

Hamed Zamani, Member

Vanessa Murdock, Member

W. Bruce Croft, Member

James Allan, Chair of the Faculty
Manning College of Information and Computer Sciences
University of Massachusetts Amherst

ACKNOWLEDGMENTS

I would like to thank my advisor Professor James Allan for supporting me throughout the Ph.D. I have learned about philosophies of information retrieval, leadership, and life by communicating with Professor Allan through numerous emails, meetings, and lab events. Looking back at myself six years ago, and considering how much I have improved, I thank him for his patience to let me grow as an independent researcher.

I would also like to acknowledge the support I received in my career from Dr. Vanessa Murdock. When I started my first internship with Dr. Murdock as my manager, I lacked the confidence to keep up with the pace of industry research. Dr. Murdock gave me confidence by wholeheartedly praising my work when it went well, and giving very helpful advice when I was stuck. I learned to be confident about my strengths and open about the areas I could improve on because of her. Both Professor Allan and Dr. Murdock have taught me leadership skills to the extent that I believe someday I can be a leader like them. As mentors, I would also like to thank Professor Brendan O'Connor, Professor Preslav Nakov, Professor Isabelle Augenstein, Professor Hamed Zamani, and Professor Bruce Croft. From professor O'Connor, I learned how to be self-critical and how to concretely define a problem in terms of input and output after critically looking at it through different lenses. Professor Nakov and Professor Augenstein gave me invaluable advice on paper writing. After working on the comments of Professor Croft on my thesis proposal, I learned how to be truly philosophical when writing the introduction and literature review for a thesis.

I enjoyed the work environment at CIIR and I thank my labmates and colleagues for making my time at CIIR wonderful. In particular, I would like to thank Hamed

Bonab, Youngwoo Kim, John Foley, Jiepu Jiang, Zhiqi Huang, Shahrzad Naseri, Hamed Zamani, Helia Hashemi, Ali Montazeralghaem, Daniel Cohen, Liu Yang, Myung-ha Jang, Lakshmi Vikraman, Tanya Chowdhury, and Qingyao Ai. Aside from my CIIR collaborators, I would like to thank several external collaborators who contributed to my research. Specifically, I would like to thank Dmitry Ignatov, Katie Keith, Sreyashi Nag, Danqing Zhang, Andrew Halterman, Momchil Hardalov, Dong-Ho Lee, Xiang Ren, Felipe Moraes, Dimitrina Zlatkova, and Yoan Dinkov. I would also like to thank the CIIR and CSCF staff including Jean Joyce, Kate Morruzzi, Dan Parker, Glenn Stowell, Michael Zarozinski, Gregory Brooks, Eileen Hamel, and Malaika, for their administrative support. Particularly, whenever I reached out to Dan Parker with any computing problem, he always smiled and made me feel better. I am yet to figure out how he remained so calm.

Finally, but most importantly, I would like to thank my friends and family for bearing with me on this difficult journey. I attribute all my successes including this Ph.D. to my parents: Md. Abu Hannan Mia and Fatema Rahman. My parents have allowed me to grow independently, supporting me when I was in trouble, but allowing me to make my own decisions. I would like to thank my best friend Azmir Haque. He was always there in my emotional turmoils. He once flew to the US to support me in my Ph.D. I do not have any siblings, but I do not regret it because of his presence. I thank Mahmood Jasim, Razia Siddiquee, Rafia Sultana, and Md. Arifur Rahman for inspiring me with surprising delicious weekend meals. I would like to thank Kirsten Nord for making the last few months of my Ph.D. enjoyable.

This work was supported in part by the Center for Intelligent Information Retrieval, in part under University of Southern California subcontract no. 124338456 under IARPA prime contract no. 2019-19051600007., and in part by the Air Force Research Laboratory (AFRL) and IARPA under contract #FA8650-17-C-9118 under subcontract #14775 from Raytheon BBN Technologies Corporation. Any opinions,

findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor

ABSTRACT

DATA SCARCITY IN EVENT ANALYSIS AND ABUSIVE LANGUAGE DETECTION

SEPTEMBER 2022

SHEIKH MUHAMMAD SARWAR

B.Sc., UNIVERSITY OF DHAKA

M.Sc., UNIVERSITY OF DHAKA

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

Lack of data is almost always the cause of suboptimal performance of neural networks. Even though data scarce scenarios can be simulated for any task by assuming limited access to training data, we study two problem areas where data scarcity is a practical challenge: *event analysis* and *abusive content detection*. Journalists, social scientists and political scientists need to retrieve and analyze event mentions in unstructured text to compute useful statistical information to understand society. We claim that it is hard to specify information need about events using keyword-based representation and propose a Query by Example (QBE) setting for event retrieval. In the QBE setting, we assume that there are a few example sentences mentioning the event class a user is interested in and we aim to retrieve relevant events using only

the examples as a query. Traditional event detection approaches are not applicable in this setting as event detection datasets are constructed based on pre-defined schemas which limits them to a small set of event and event-argument types. Moreover, the amount of annotated data in event detection datasets is limited that only allows us to build retrieval corpus for evaluation. Thus we assume that there are no relevance judgments to train an event retrieval model – except for the few examples of a specific event type. We create three QBE evaluation settings from three event detection datasets: PoliceKilling, ACE, and IndiaPoliceEvents. For the PoliceKilling dataset, where a relevant sentence describes a police killing event, we show that a query model constructed from the NLP features extracted from the few given examples are effective compared to event detection baselines. For the ACE dataset, where there are thirty-three types of events, we construct a QBE setting for each type and show that a sentence embedding approach effectively transfers for event matching. Finally, we conducted a unified evaluation of all the three datasets using the sentence-embedding based model and showed that it outperforms strong baselines.

We further examine the effect of data scarcity in abusive language detection. We first study a specific type of abusive language – hate speech. Neural hate speech detection models trained from one dataset poorly generalize to another dataset from a different domain. This is because characteristics of hate speech vary based on racial and cultural aspects. Our data scarcity scenario assumes that we have a hate speech dataset from a domain and it needs to generalize to a test set from another domain using the unlabeled data from the test domain only. Thus we assume zero target domain data in this scenario. To tackle the data scarcity, we propose an unsupervised domain adaptation approach to augment labeled data for hate speech detection. We evaluate the approach with three different models (character CNNs, BiLSTMs and BERT) on three different collections. We show our approach improves Area under

the Precision/Recall curve by as much as 42% and recall by as much as 278%, with no loss (and in some cases a significant gain) in precision.

Finally, we examine the cross-lingual abusive language detection problem. Abusive language is a super class of hate speech that includes profanity, aggression, offensiveness, cyberbullying, toxicity and hate speech itself. There are large collection of abusive language detection datasets in English such as Jigsaw. For other languages there exist datasets for abusive language detection but with very limited data. We propose a cross-lingual transfer learning approach to learn an effective neural abusive language classifier for such low-resource languages with help from a dataset from a resource-rich language. The framework is based on a nearest-neighbor architecture and is thus interpretable by design. It is a modern instantiation of the classic k-nearest neighbor model, as we use transformer representations in all its components. Unlike prior work on neighborhood based approaches, we encode the neighborhood information based on query-neighbor interactions. We propose two encoding schemes and show their effectiveness using both qualitative and quantitative analyses. Our evaluation results on eight languages from two different datasets for abusive language detection show sizable improvements in F1 over strong baselines.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

With the proliferation of internet-based applications content creation and access have become easier than ever. As a result, by 2025, some predict that there will be as much as 175 zettabytes of data in the global "datasphere" (Reinsel et al., 2018). This datasphere already contains a massive amount of text and there are a number of stakeholders who are interested in capturing useful information from this textual datasphere. One way the stakeholders generally achieve this is by employing a computer algorithm to extract the *meaning* of a textual data slice of interest – in a way that is useful to them. The textual data slice could be a sentence, passage, document or a collection of documents. The meaning of the slice depends on what is useful information to the stakeholders. For example, if we consider "hate speech" as the meaning, it is useful for the stakeholders to know whether a data slice is relevant to that meaning. This could be helpful if the stakeholders are interested to filter hateful content. Sometimes, depending on the use cases, the meaning of data is also obtained by aggregating the meanings from the individual slices of it. In order to evaluate if a computer algorithm can effectively identify the meaning of data, the stakeholders design *tasks*. For example, hate speech detection in Arabic tweets is a task.

A sampled collection is expected to be a representative sample of the data source, but this assumption rarely holds because the amount of data in a source is generally massive and noisy. As a result some form of selection bias is unavoidable in the

sampling process. Even if we obtain an unbiased sample, the data is likely to quickly evolve over time making the collection biased towards a specific time.

Once the task designers construct the collection using a sampling technique, they provide an annotation guideline to human annotators that describes the meanings they are interested in finding from the collection along with a label for each of those meanings. They generally provide long textual descriptions to describe those meanings. For example, for a retrieval task, the task designers provide a long textual description for a keyword query indicating its intent which annotators use to determine whether a document is relevant to the query or not. The responsibility of human annotators is to assign meaning to data through relevance in case of a retrieval task or labels in case of a classification task. However, the annotation cost is generally very high as it takes significant human effort. The annotation process might also be sensitive to the annotators if annotators are asked to observe contents that are harmful to their mental health. This is why the whole collection is not usually annotated. We refer to the annotated and un-annotated portions of the data as target-task data and unlabeled target-task data, respectively.

After the annotation process ends, the target-task data is used for evaluating computer algorithms. A typical approach is to at first teach the task to the computer algorithm by providing it a portion of the target-task data. Through this process the computer algorithm almost certainly learns the collection sampling bias, annotation guideline bias, and annotator bias. The more target-task data we provide the more it learns and the better it performs on the held out target-task data. This improvement could be attributed to the algorithm's ability to capture the bias within the task, and it might not necessarily show that the underlying problem associated with the task has been solved. For example, for solving the problem of hate speech detection, several researchers have constructed several tasks and showed high performance based on task-level evaluation metrics with task-level data as input for learning as well as

evaluation. Recent works on hate speech detection showed that computer algorithms, specifically neural networks, that learn from one hate-speech task and are evaluated on another task fail by a large margin because of learning the task bias (Arango et al., 2019). Thus if we want to solve the hate speech detection *problem*, we cannot conclude that a computer algorithm solves it even if it performs very well on all of the hate speech detection tasks. Generally, it is non-trivial to design a task that is representative of the problem.

Two tasks may focus on solving the same underlying problem, but they can be very different in terms of *data sources*, *language- and domain-specific sampling constraints*, *annotation guidelines*, and the *hired human judges*. As an example, for tasks such as abusive language detection, the annotation guidelines can be different based on how the task designers define "abuse". For an event analysis task, the annotation guidelines could vary based on the events that the task designers are interested in. For example, in one task the task designers may be interested in killing events, while in another one they might be interested in *police* killing events. The amount of high-quality human-annotated data needed to solve a task, e.g., detecting hate-speech about women in a Reddit – is not sufficient to solve a problem, e.g., hate speech detection in social media. Thus, the amount of data to solve a problem is generally scarce.

One way to evaluate if the underlying problem has been solved is to measure the performance of the algorithms across the tasks. This thesis provides directions towards improving cross-task performance by proposing novel techniques for augmenting data and learning from the augmented data. We assume that we have from zero (a zero-shot setting) to at best a thousand (a limited-data setting) annotated data instances from a task and we can borrow data from any related task to teach a computer algorithm.

To augment data, we identify a source task when we assume that we can estimate the target-task data distribution with the labeled data points from the source task. We assume that the source task is identified by a domain expert. Automatically identifying such an aligned source task is beyond the scope of this thesis.

Along with the source-task data for learning the distribution, when available, we also assume access to some unlabeled target-task data (chapter 4). In this scenario, we use a data generation module that takes the source-task data and target-task unlabeled data as input and generates synthetic labeled data that where the content is more similar to target-task content and the labeling knowledge is taken from the source-task. We also consider augmenting task-relevant rules in the form of knowledge to improve performance on the target task. We implement these techniques separately and show gains from them. A vision of this thesis is to use the source-task data, synthetic data, and task-relevant knowledge as augmentations to improve performance in data scarce scenarios.

Along with the augmentation techniques we explore and propose transfer learning techniques that can harness benefits of all the above mentioned augmentations to learn a model that effectively transfers the knowledge from augmentations to the target-task. On occasion we assume that a limited amount (generally in the range of hundreds) of labeled data is available from the target-task, and we also leverage that along with other augmentations at the time of transferring knowledge to a model. One simple way to transfer knowledge is to combine all the data sources and learn the computational model from them. However, this simple process often does not give optimal gain and we propose an approach in this thesis that learns to transfer knowledge from multiple sources (Chapter 3 and Chapter 5).

In summary, we propose novel data augmentation and transfer learning to handle data scarcity in two areas: *event retrieval* and *online content moderation*. We design the event retrieval task, create three evaluation corpora, and show that a sentence

embedding space learned from the dataset for the Natural Language Inference (NLI) (Bowman et al., 2015) task serves as a reasonable alternative to an event embedding space for retrieving similar events. Thus we show that it is possible to achieve a task-level transfer from NLI to unsupervised event retrieval (Sarwar and Allan, 2020, 2019; Halterman et al., 2021). We further boost the performance of our sentence-embedding based approach by segmenting sentences into events using a Semantic Role Labeling (SRL) approach Zhang et al. (2017). This indicates that it is important to augment task-specific knowledge along the transfer process.

We also address data scarcity issues in two online content moderation tasks that bear practical challenges – but that have not been addressed in existing content moderation literature. Specifically, we investigate hate speech detection and abusive detection tasks in zero-shot (Sarwar and Murdock, 2022) and limited-data settings (Sarwar et al., 2021), respectively. Data scarcity is a common and big challenge in content moderation because of the annotator disagreement, racial bias, and mental health issues that occur when annotating abusive contents (Schmidt and Wiegand, 2017; Waseem, 2016; Malmasi and Zampieri, 2018; Mathur et al., 2018).

1.1 Data Scarcity in Event Analysis

In Chapter 3, we provide a technical discussion on data scarcity in event analysis. The most extreme case of data scarcity that we tackle in this work is a Query by Example (QBE) scenario where there are only a few labeled data items from the target task and and an aligned source task does not exist. QBE is an effective alternative to keyword queries for identifying user information needs. It has been applied to retrieve entities and documents from unstructured text corpora (Smucker and Allan, 2006; Sarwar and Allan, 2019; Sarwar et al., 2019a), entities from knowledge graphs (Metzger et al., 2017), and tuples from relational databases (Fariha et al., 2018). QBE approaches are motivated by the fact that it is often easier for a user to express an

information need with examples rather than a natural language description (Fariha et al., 2018; Metzger et al., 2017). This is a realistic setting considering the needs of journalists and social scientists.

Journalists and social scientists are interested in mentions of specific types of events in unstructured text. Social scientists extract statistical information from text to answer substantive event-centered questions: How do actors respond to contested elections (Daxecker et al., 2019)? How many people attend protests (Chenoweth and Lewis, 2013)? Which religious groups are engaged in violence (Brathwaite and Park, 2018)? Why do some governments try to prevent anti-minority riots while others do not (Wilkinson, 2006)? How many civilians were killed by police (Keith et al., 2017)?. In the absence of official records, social scientists often turn to news data to extract the actions of actors and surrounding events (Halterman et al., 2021). These news-based event datasets are often constructed by hand, requiring large investments of time and money and limiting the number of researchers who can undertake data collection efforts.

In order to design a system that helps social scientists in finding answers to their substantive event-centered questions, it is important to find the relevant event mentions at first. For example, a first step to estimate the number of civilians killed by police is to find sentence- or document-level mentions of police killing events. Once such textual evidence is retrieved we can apply an automatic or manual aggregation process on it to find the number of civilians killed by police. We only focus on retrieving sentence-level mentions of target events specified by a query and leave the aggregation process as future work.

We explore the Query-by-Example (QBE) setting for retrieving sentences where a target event (e.g., arrest) or a target agent (e.g., police) or patient (e.g., civilian) type appears. We take the QBE paradigm because a keyword-query based sentence retrieval model Murdock (2007) is likely to fail in this retrieval context no matter

how sophisticated the model is; it will likely be an under-specified representation of the information need as it lacks the contextual information that is required to model the dependencies between an event trigger token span and the argument token spans. On the other hand, a statistical retrieval model based on keyword queries can still be an effective first step in finding documents containing relevant sentences representing a target event.

For example, consider the case where a social scientist wants to find all the *jail release* events from a corpus. The social scientist will expect a high-recall retrieval model that will find a large proportion of the relevant events at a ranking cut-off deeper than typical web search engines. This is because the social scientist is more interested in the prevalence of such events for computing useful statistics, and thus spending a few hours to get those events is still efficient compared to inspecting all the documents.

To start the search process, the social scientist retrieves a ranked-list of documents with combination of keywords such as *jail*, *release*, *sentence*, etc., and manually finds event-sentences that are examples of what is desired. Although these sentences on their own could constitute a representation of her information need – i.e., a query in the form of examples (QBE) – keyword-based approaches do not provide support for such an event query except by using the set of example sentences as a bag-of-words query. This is because the keyword query does not provide the opportunity to leverage syntactic and semantic information that necessary to retrieve sentence-level event mentions. We explore different approaches in three different datasets to retrieve sentence-level event mentions in a QBE setting.

To solve QBE using data augmentation and transfer learning, we assume that a dataset for solving the Natural Language Inference (NLI) problem is our source task. A model that learns to measure semantic similarity between a pair of sentences – which is the NLI task – helps to score a pair of sentences based on their likelihood

of containing the same events. The augmentation and transfer phase are trivial in this case. However, we found that an NLI model under-performs in the transfer stage because even if a common event appears between a query sentence and a candidate sentence, the candidate sentence often contains other events along with the query event. It gives noisy input to the NLI model. We use PredPatt Zhang et al. (2017) that extracts events from candidate sentences and creates an unsupervised event-based segmentation of the candidate sentences. PredPatt is an unsupervised and rule based approach. This is how we augment task-relevant knowledge at the time of transfer.

Our contributions in the QBE setting for event retrieval are as follows:

- We create three settings to explore the QBE paradigm based on three different event detection datasets: PoliceKilling (Keith et al., 2017), ACE (Walker, 2006) and IndiaPoliceEvents (Halterman et al., 2021) to understand the challenges of QBE for event retrieval.

- In our PoliceKilling setting – with a few relevant sentences as a query – we retrieve sentences mentioning police-killing events where a person was killed by a police officer. To solve this, we propose SearchIE, a hybrid of IR and Natural Language Processing (NLP) approaches that indexes sentences represented using handcrafted NLP features. At query time, SearchIE samples terms from a Logistic Regression model trained with the few query sentences and uses them to query the retrieval index. We show that SearchIE outperforms state-of-the-art NLP models used to find civilians killed by US police officers – even with a single civilian name as a query (Sarwar and Allan, 2019). Given 20 examples SearchIE achieves 95% precision at top-5 which is an absolute improvement of 35% over the state-of-the art baseline from Keith et al. (2017).

- In our ACE setting, we propose a Semantic Role Labeling (SRL) based approach to identify event spans in sentences and use a state-of-the-art sentence matching model, Sentence BERT (SBERT) (Reimers and Gurevych, 2019a), to match event spans in queries and documents without any supervision (Sarwar and Allan, 2020). We show that given 10 examples our approach achieves a 5% absolute improvement for precision at top-10 over the RM3 baseline.

- For the third setting, we contribute a new dataset, IndiaPoliceEvents, with a goal to explore QBE approaches. We employ trained annotators to classify sentences from the Times of India into five event types. The new event detection dataset, IndiaPoliceEvents (Halterman et al., 2021), focuses on the needs of social scientists as the sentences are sampled from news articles published when the Gujarat riot took place. Social scientists are interested in monitoring influential political agents such as police and they want to retrieve sentence-level and document-level evidence for the activities of the police force.

- We perform an evaluation of our SRL and SBERT based sentence matching approach on QBE settings formulated from PoliceKilling, ACE, and IndiaPoliceEvents datasets. We find that in all the settings our approach outperforms strong retrieval baselines.

1.2 Data Scarcity in Online Content Moderation

Online content moderation has become an increasingly important problem – small-scale websites and large-scale corporations alike strive to remove harmful content from their platforms (Vidgen et al., 2019; Pavlopoulos et al., 2017; Wulczyn et al., 2017). This is partly in anticipation of proposed legislation, such as the *Digital Service Act* (Commission, 2022) in the EU and the *Online Harms Bill* (Government, 2022) in the UK. Even without such legislative efforts, it is clear that the lack of content

moderation can have significant impact on businesses (e.g., Parler was denied server space[1]), on governments (e.g., U.S. Capitol Riots[2]), and on individuals, e.g., because hate speech is linked to self-harm (Jürgens et al., 2019).

We address limited data issues for two tasks related to content moderation: *hate speech detection* and *abusive language detection*. In Chapter 4, we provide an approach for cross-domain hate speech detection that exploits the structure of hate speech. In Chapter 5, we provide a transformer-based k-nearest neighbor approach for cross-language abusive content detection.

1.2.1 Data Scarcity in Hate Speech Detection

Chapter 4 addresses the problem of zero-shot hate speech detection. In this chapter, we propose a new synthetic *data generation* approach.

Online harassment in the form of hate speech has been on the rise in recent years. A recent paper (ADL, 2020) from the Anti-Defamation League reports that nearly half (44%) of Americans report having experienced some type of online harassment, up from 41% in 2017. Of those 44%, 35% report having been harassed as a result of their sexual orientation, religion, race or ethnicity, gender identity, or disability.

The problem is exacerbated by machine learned systems that are trained using labeled data from online forums. With inadequate hate speech filtering, these systems themselves become vectors of hate. For example, YouTube (Tufecki, 2019) has been found to promote hate speech via its recommended videos simply by learning from user interactions. In 2016 Microsoft released a conversational agent "Tay" that learned from user interactions on Twitter, but had to take it down a short time later because it was generating racist content (Schwartz, 2019).

---

[1] `https://www.nbcnews.com/tech/tech-news/amazon-suspends-hosting-parler-its-servers-citing-violent-content-n1253648`

[2] `https://www.cbsnews.com/news/capitol-riot-arrests-2021-02-27/`

To filter hate speech, a machine learned system will need large amounts of training data with adequate coverage of the vocabulary. It is difficult to create a high-coverage vocabulary of offensive terms or phrases that occur in hate speech mentions because of regional and linguistic variants even within the same language, compounded by variety in the targets of hate speech. The terms directed at one target often have little or no overlap with terms directed at a different target. Furthermore, hate speech often does not contain any terms that are offensive in and of themselves. Rather it is contextually hateful, referring to offensive stereotypes, or alluding to or inciting violence against a target group.

Recent approaches to hate speech detection are based on supervised neural representation learning (MacAvaney et al., 2019; Glavaš et al., 2020; Pamungkas and Patti, 2019; Badjatiya et al., 2019; Agrawal and Awekar, 2018a; Arango et al., 2019; Waseem et al., 2018). These approaches require a large number of hate speech instances to achieve high recall in the hate speech class. Arango et al. (2019) found that the performance of neural models trained using data from Waseem (2016) drops significantly when tested on data from Basile et al. (2019), which is from a different domain. The failure of the models to generalize to a target task is due to user bias in the source-task data, where a small number of users generate the majority of hateful examples. Furthermore, since hate speech occupies a tiny proportion of data from a domain, test collections are often constructed by searching with a set of seed terms from a hate speech lexicon. This results in a data set with a domain-limited vocabulary which itself may have the same shortcomings. For example, a source data set seeded by anti-Muslim terms may be inadequate for detecting anti-woman content in target domain data.

One way to address the domain mismatch is to gather labeled data from the target domain. Since it is sensitive and costly to obtain annotations for hate speech (Schmidt and Wiegand, 2017; Waseem, 2016; Malmasi and Zampieri, 2018; Mathur

et al., 2018), it is desirable to utilize unlabeled data from the target domain to build a robust classifier. Thus, Unsupervised Domain Adaptation (UDA) – i.e., the problem of building a robust target domain classifier with labeled data from the source domain and unlabeled data from the target domain – is a realistic and important problem in the context of hate speech detection.

We contribute a method based on our Augment-Transfer framework that automatically generates a domain-adapted corpus to bridge the gap between source domain and target domain for hate speech detection. Although there are cross-domain studies for hate speech detection, to the best of our knowledge, this is the first study of UDA for hate speech detection.

We identify hate speech sentences where the hate speech content terms can be distinguished from their surrounding sentence context. For example[3] in the sentence "The problem with Honda CRVs is that they are boring", the content consists of the subject "Honda CRVs" and the negative descriptor "boring". The surrounding sentence context is "The problem with ... is that they are ...".

While all hate speech does not have this structure, leveraging examples that do provides a convenient template for domain adaptation. We can automatically identify the template in generic sentences with negative sentiment and slot in hate speech content to convert it to synthetic hate speech in a new domain. Note that the process does not have to be perfect because this type of training data can be generated in large quantities.

To create a domain-adapted corpus, we train a sequential tagger on the labeled data in the source domain so that the tagger is able to identify hate speech content terms, and surrounding sentence context templates. We apply the tagger to unlabeled

[3]In this thesis we intentionally use non-hate examples to limit the level of offensiveness in the thesis itself. In this example "Honda CRVs" (or by proxy, their owners) are not considered an at-risk or protected group.

data in the target domain to derive a lexicon of hate terms in the target domain. We also apply it to a large corpus of generic sentences with negative sentiment. This yields a large data set of sentence contexts that will serve as hate speech templates. In this work we use a collection of Twitter posts labeled with negative sentiment based on emojis (Go et al., 2009). As the posts are labeled using emojis this collection can be extended without any supervision meaning we can generate hate speech templates in abundance.

To adapt the generic hate speech templates to the target domain, we rank them according to their textual similarity to the target domain sentences, and select the top k for augmentation. This reduces noise in the domain-adapted data and increases the topical similarity between the generic templates and the target domain. Finally, we impute terms from the derived hate speech lexicon from the target domain into the generic templates. The result is a large corpus of negative sentences with hate speech content from the target domain. Our contributions in this work are:

- We propose an unsupervised domain adaptation approach to augment labeled data for hate speech detection. Specifically, we propose to convert a large collection of general domain negative emotion sentences into target domain specific hate speech using unlabeled data from the target domain along with a hate speech lexicon.

- We evaluate the effectiveness of the augmented data with three different models (character CNNs, BiLSTMs and BERT) on three different collections. We show our approach improves Area under the Precision/Recall curve by as much as 42% and recall by as much as 278%, with no loss (and in some cases a significant gain) in precision.

1.2.2 Data Scarcity in Cross-Lingual Abusive Language Detection

In Chapter 5 , from our experience with zero-shot hate speech detection and QBE for event retrieval we tackle a low-resource scenario for abusive language detection. In this thesis, abusive language includes personal attacks, hate speech, cyberbullying, sexual harassment, trolling, profanity, threats of violence, name calling, and discrimination (Wulczyn et al., 2017; Jigsaw, 2018). Hate speech is a specific type of abusive language that attacks and deprecates an individual or groups of people because of their race, ethnicity, gender, nationality, religion and any other characteristics (Nockleby, 1994; Parekh, 2012).

A core challenge in developing content moderation systems is the lack of available resources for languages other than English. Accordingly, our task here is to create an abusive language detection model for a target language with limited annotated data by transferring knowledge from another dataset in a different language, for which a large amount of training data is available.

There are existing approaches that could partially address this challenge. A popular approach is to fine-tune multilingual language models such as XLM-R (Conneau et al., 2020) or mBERT (Devlin et al., 2019) on the target dataset (Glavaš et al., 2020; Stappen et al., 2020). To incorporate the source dataset knowledge, a sequential adaptation technique (Garg et al., 2020) which first fine-tunes a multilingual LM on the source dataset, then on the target dataset, can be used. There are also existing approaches for mixing the source and the target datasets (Shnarch et al., 2018) in different proportions, followed by fine-tuning the multilingual LM on the resulting dataset. While sequential adaptation introduces the risk of forgetting the knowledge from the source dataset, such mixing methods are driven by heuristics that are effective, but not systematic. Crucially, as we argue here, this is because they do not model the relationship between source and target data.

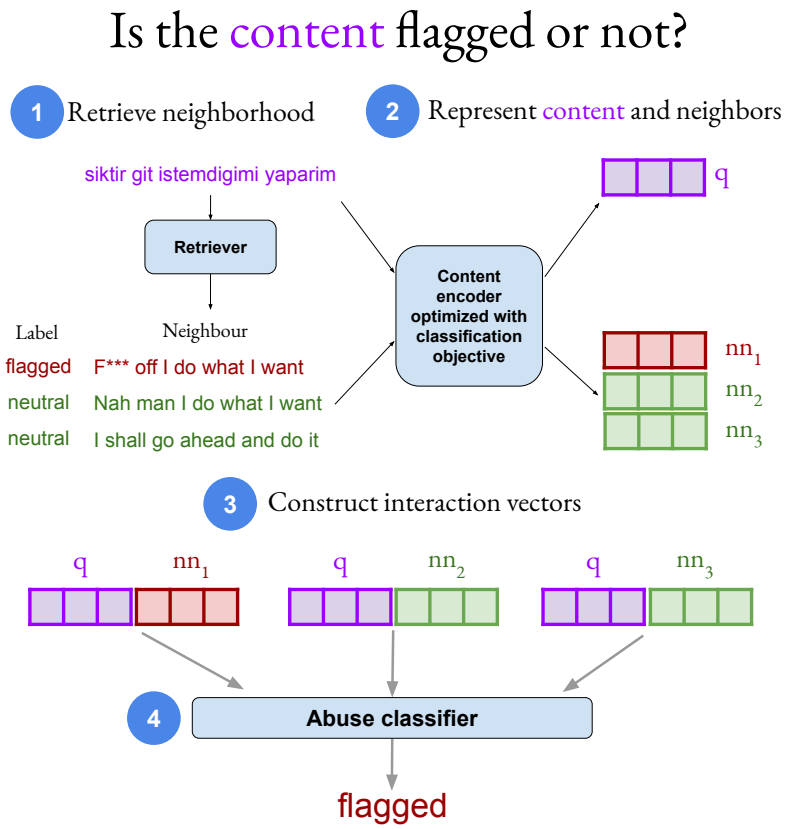


Figure 1.1: Conceptual diagram of our neighborhood framework. The query is processed using run-time compute, while the neighbor vector is pre-computed.

Another problem arises if we consider that new data cases with novel labels can be added to the source dataset. This is a specifically pertinent issue for content moderation, as efforts to create new resources often define their own label taxonomies (Banko et al., 2020). In that case, model re-training becomes a requirement in order to map the new label space to the output layer that is used for fine-tuning.

We propose a transformer-based k-Nearest Neighbor (kNN$^+$) framework,[4] a one-stop solution and a significant improvement over the classic k-NN model for the abusive language detection problem. Our framework addresses the above-mentioned challenges, which are not easy to solve via simple fine-tuning of pre-trained language

[4]We use + superscript to indicate that our kNN$^+$ framework is an improvement over the classic kNN model.

models. Moreover, to the best of our knowledge, our framework is the first attempt to use k-NN for transfer learning for the task of abusive content detection. Given a query, which is a training or an evaluation data point from the target dataset, kNN$^+$ retrieves its nearest neighbors using a language-agnostic sentence embedding model. Then, it constructs transformer representations for the query and for its neighbors. After that, the interactions between the query and each of the neighbors are modelled using *interaction features* computed from the transformer representation. The interaction features are indicative of the agreement or the disagreement between a query and its neighbors. The framework further uses a self-attention mechanism to aggregate the interaction features between the query and each of its neighbors, and to classify the input query. The conceptual framework is shown in Figure 1.1. Note that our framework is robust to neighbors with incorrect labels, as it can learn to disagree with them as part of its training process.

We instantiate two variants from our framework: Cross-Encoder (CE) kNN$^+$, and Bi-Encoder (BE) kNN$^+$. The CE kNN$^+$ concatenates the query and a neighbor, and passes that sequence through a transformer to obtain interaction features. The BE kNN$^+$ computes representations of the query and a neighbor by passing them individually through a transformer, and computes interaction features from those representations. BE kNN$^+$ is more efficient compared to CE kNN$^+$, but it does not yield the same performance gain. Both models outperform six strong baselines both in cross-lingual and in multilingual settings. Our contributions are summarized as follows:

- We demonstrate that neighborhood methods, such as kNN are a viable candidates for solving the content flagging task.
- We propose a novel framework, kNN$^+$, which, unlike a classic kNN, models the relationship of a data point and each of its neighbors to represent the neighborhood, using language-agnostic transformers.

- Our evaluation results on eight languages from two different datasets for abusive language detection show sizable improvements of up to 9.5 F1 points absolute (for Italian) over strong baselines. On average, we achieve 3.6 absolute F1 points of improvement for the three languages in the Jigsaw Multilingual dataset and 2.14 points for the WUL dataset.

CHAPTER 2

RELATED WORK

We present an Augment-Transfer framework to mitigate data scarcity in event analysis and abusive language detection. In section 2.1 we provide the definitions of events, hate speech, and abusive language based on existing literature. Then, in section 2.2 we provide a literature review to discuss progress on our tasks of interest: i) event retrieval, ii) abusive language detection, and iii) hate speech detection.

After a discussion on previous efforts on our tasks, in section 2.3, we discuss general data scarcity settings such as Query by Example (QBE), zero-shot, and limited-data. In section 2.4 we provide background on techniques such as semi-supervision, weak-supervision, data augmentation, and transfer learning to handle data scarcity in general. We later contextualize the discussion on these techniques and scarcity settings in sections 2.5 and 2.6 based on the literature analysis of data scarcity in our tasks. Based on our analysis of prior work, we conclude that these settings – in the context of our target tasks – are novel, challenging, and are motivated by real-world applications.

2.1 Backgrounds on Events and Abusive Language

In this section, we provide the definition of event, abusive language, and hate speech that we use throughout the technical chapters.

2.1.1 Events

An event described in a textual content indicates the occurrence of something and it generally includes mentions of entities such as people, object, etc. who take part in

George	Floyd	was	killed	by	police	on	May	,	25	2020	at	Minneapolis
Patient		-	Trigger	-	Agent	-	Date				-	Location
Argument		-	Predicate	-	Argument	-	Argument				-	Argument

Figure 2.1: Example of a sentence-level event mention, its participants along with the spatio-temporal aspects.

or are affected by that event. An event description often also includes spatio-temporal information indicating the time and location of the event. In this thesis, we focus on sentence-level mentions of events, while it is possible for an event mention to spread across a paragraph, document or multiple documents.

An example of a sentence-level mention of events with the participants and spatio-temporal information is shown in Figure 2.1. The sentence mentions a killing event. In the event mention, George Floyd is the *patient* or *direct object* because he was killed. *Police* is the *agent* or *actor* or *subject* as "police" carried out that event. The temporal aspect of the event indicates that the event happened in the past on May 25, 2020. The spatial aspect of the event indicates that it happened in Minneapolis. We refer to the agent, patient, data and location of an event as *arguments* and the keyword indicating the occurrence of the event as *predicate*. We use the terms "trigger" and "predicate" interchangeably to refer to keywords indicating an event occurrence. Understanding events and their descriptions in text has practical applications in news summarization, information retrieval, knowledge base construction etc (Yang and Mitchell, 2016).

2.1.2 Abusive Language

Presence of abusive language in online platforms is a serious and growing problem. It inhibits a platform user's active participation in online activities offered by the platform. Abusive language is an outcome of the negative online behavior of a group of platform users that surfaces through the text modality. In this thesis, we set

a broad definition of abusive language that includes personal attacks, hate speech, cyberbullying, sexual harassment, trolling, profanity, threats of violence, name calling, and discrimination (Wulczyn et al., 2017; Jigsaw, 2018). The key feature of online abusive language is that it can be harmful to a person or a group, to the online community where it occurs, or to the social platform hosting the conversation (Nakov et al., 2021). This spectrum poses challenges for clear labelling of training data, as well as for computational modelling of the problem.

2.1.3 Hate Speech

Hate speech attacks and deprecates an individual or groups of people because of their race, ethnicity, gender, nationality, religion and any other characteristics (Nockleby, 1994; Parekh, 2012). Hate speech is a specific type of abusive language that is directed towards a target. Hate speech in social media can cause tension between people and communities that might eventually lead to hate crime (Watanabe et al., 2018; Müller and Schwarz, 2020). This makes it essential to identify users who write and promote hate speech before the hate speech can agitate people enough to cause hate crime.

2.2 Literature Review on Application Areas

In this section, we provide a discussion on the problems we address and the progress the research community has made to solve them.

2.2.1 Event Detection, Extraction, and Retrieval

In this section, we discuss prior work on retrieving event information from unstructured text. To provide more context for understanding event retrieval we describe existing literature on event detection as well as extraction. Then we discuss prior work on event retrieval to motivate our QBE setting for event retrieval. We show how our setting is different from the existing ones.

2.2.1.1 Event Extraction

Event extraction comprises two sub-tasks: event detection and event-argument extraction. Event detection is the task of classifying a sequence of tokens in a text sequence into one or more event types (Nguyen and Grishman, 2015). Event extraction further includes identifying the token sequences representing the entities involved in the event. This generally includes the agents, patients, time and the location of the event. A typical event extraction pipeline based on the methodology proposed by Ahn (2006) is shown in Figure 2.2.

The first step is to identify the event triggers and assign them an event type. In Figure 2.2 a *killing* event is mentioned in the sentence and the sequence of tokens (i.e., the trigger that represent this event) is "shot dead" and the event type is "kill". The second step of event extraction is argument span detection. In this step, the token sequences representing the arguments of the event trigger are identified. In the example shown in Figure 2.2, there are two arguments to the killing event *Alton Sterling* and *two officers*. This step also includes assigning roles to each of the arguments. There are coarse-grained and fine-grained roles. In this example, the coarse-grained roles assigned to *Alton Sterling* and *two officers* are Patient and Agent, respectively, while their fine-grained roles are civilian and police.

An event extraction dataset such as ACE (Walker, 2006) contains many event mentions and they are annotated with an event schema, where the event schema include the event type (for example killing) as well as the argument roles (for example killer). The roles vary depending on the event type. For example, for a transportation event there is a role *destination*, which does not appear in a killing event. We claim that a schema to represent an event is a bottleneck to applying event detection algorithms in a query-by-example setting. This is because an event detection model trained from a dataset with a fixed set of schemas might not detect any event from the examples. Moreover, even if the event type can be detected from an example, the

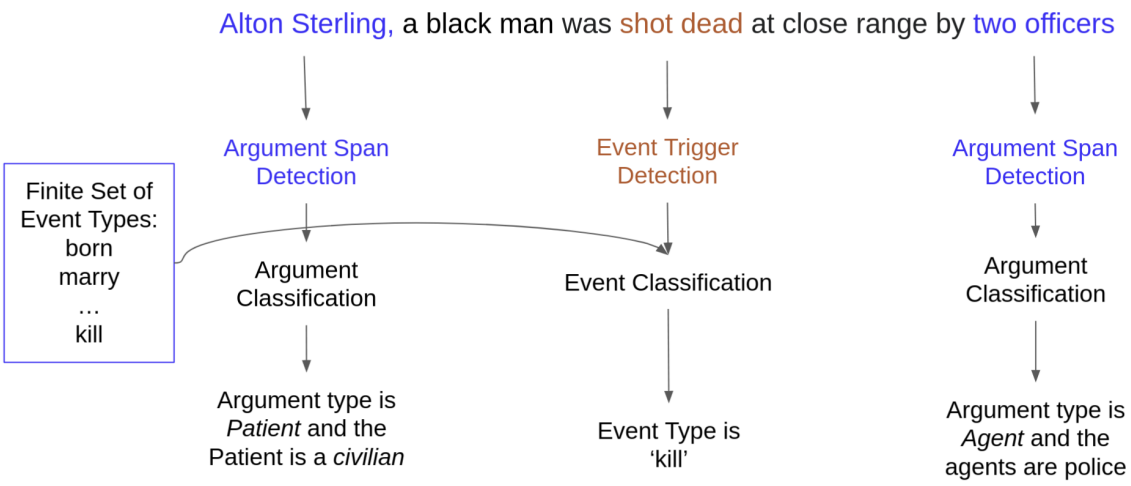


Figure 2.2: Event extraction pipeline

argument role might be unseen in the training data. Consider our example in Figure 2.2. Assume that the sentence is a an example in our query-by-example setting and an event detection model successfully identifies the kill event in the sentence. Now we apply the event detection model on our retrieval corpus and identify all the killing events. However, the sentence level semantics indicates that by using this sentence as a query a user is looking for *police* killing events and not just any killing event. Thus it is not sufficient to detect the event type; it is also necessary to identify the argument and its role type. Now, assume the dataset used to train the event detection model contained a large number of instances of the killing event, but that in none of those instances the agent was Police. This makes the schema-based event detection systems ineffective in a query-by-example setting.

2.2.1.2 Event Retrieval

There are a few works on event retrieval and those retrieval settings are different from our query-by-example setting. Metzler et al. (2012) proposed the *microblog event retrieval* task and used keyword queries to perform retrieval on a Twitter corpus constructed over a period of time. Their approach involved detection of time-spans

in which a target event occurred and summarization of the contents in that time-span for describing the event. Rudra et al. (2018) explored a similar approach to retrieve disaster-related information – e.g., about infrastructure damage, urgent needs of affected people. They identified sub-events using noun-verb pairs that closely occur in different tweets, for example "airport shutdown." Finally, they summarized the contents associated with the sub-events using an Integer Linear Programming approach. These approaches are fundamentally focused towards single keyword or phrase queries such as *earthquake* to detect events from Microblogs. In contrast, our event queries are constructed from example event descriptions.

Topic Detection and Tracking Topic Detection and Tracking (TDT) is similar to Query by Example (QBE) for event retrieval in spirit except that it begins with an empty set of examples for a specific event that eventually grows as a story on an event is identified Allan (2002). The input in TDT is treated as a stream of data that needs to be organized around an event-centered topic. There could be many other related events within the discourse of that event-centered topic. As a system observes more data from the stream, the number of examples for an event grows, with some false positive examples for each of the events. At some point, when a new data instance comes out of the stream it is used as a query to retrieve examples organized by the events. In that sense the retrieval collection is ever evolving.

TDT defines events as concrete instantiation of an event type whereas in our event retrieval task we focus on retrieving a more abstract instantiation of an event type. For example, for an event type "kill", a concrete instantiation of this event type will be an event that contains information about the agent (e.g., Police), patient (e.g., Alton Sterling), location (e.g., New Jersey) and time (e.g, 10 pm). In TDT, the hardness of the task lies in differentiating documents containing two concrete instantiation of the same type of events. This is because two documents with two concrete instantiations are generally very similar in terms of the distribution of the terms. Thus tf-idf based

approaches are not successful in TDT without the integration of named entity types, term surpriseness etc (Makkonen et al., 2004; Kumaran and Allan, 2004).

If we think about a pair of different sentence-level concrete event instances, they could be relatively easier to distinguish, as syntactic parsing of a sentence can provide important clues which are not trivial to obtain at document level. In this thesis we focus on sentence-level event mentions and try to discern between abstract instantiation of events such as *police killing* and *police failures*. Moreover, our problem focuses on a static collection compared to an evolving one in TDT.

2.2.2 Abusive Language and Hate Speech Detection

There have been several efforts to detect specific types of offensive content, e.g., hate speech, offensive language, cyberullying, and cyber-aggression. Hate speech detection is by far the most studied abusive language detection task (Ousidhoum et al., 2019; Kwok and Wang, 2013; Djuric et al., 2015; Chung et al., 2019; Burnap and Williams, 2015). Davidson et al. (2017) created one of the most widely used datasets for this task with over 24,000 English tweets labelled as *hate speech*, *profanity*, and *non-offensive*. Basile et al. (2019) organized a shared task for hate speech detection in English and Spanish.

There have also been numerous efforts to detect offensive language: OffensEval 2019–2020 (Zampieri et al., 2019; Zampieri et al., 2020) for English, Arabic, Danish, Greek, and Turkish, GermEval 2018 (Wiegand et al., 2018) for German, HASOC 2019 (Mandl et al., 2019) for English, German, and Hindi, TRAC 2018–2020 for English, Bengali, and Hindi (Fortuna et al., 2018; Kumar et al., 2020). Another popular and large-scale offensive language detection task came out as a part of The Toxic Comment Classification Challenge (Jigsaw, 2018) organized in the Kaggle platform. The organizers provided participants with almost 160K comments from Wikipedia organised in six classes: *toxic, severe toxic, obscene, threat, insult*, and

identity hate. The task was later extended to multiple languages (Jigsaw Multilingual, 2020),[1] offering 8,000 Italian, Spanish, and Turkish comments. There have also been datasets that cover various types of abusive language. Founta et al. (2018) addressed hate and abusive speech on Twitter, introducing a dataset of 100K tweets. Glavaš et al. (2020) targeted hate speech, aggression, and attacks in three different domains: Fox News (from GAO), Twitter/Facebook (from TRAC), and Wikipedia (from WUL). In addition to English, it further offered parallel examples in Albanian, Croatian, German, Russian, and Turkish. However, the dataset is small, containing only 999 examples.

2.2.2.1 Learning Approaches

Abuse detection is a special instance of text classification. Thus, it follows the recent trend of fine-tuning a pre-trained transformer with data from target-task. Typically, pre-trained language models such as BERT Devlin et al. (2019), RoBERTa Liu et al. (2019), ALBERT Lan et al. (2020), and GPT-2 Radford et al. (2019) are used for fine-tuning. In a multi-lingual setup, also mBERT Devlin et al. (2019) and XLM-RoBERTa Conneau et al. (2020) have shown to be useful. Other popular models include CNNs Fukushima (1980), RNNs Rumelhart et al. (1986), and GRUs Cho et al. (2014), including ELMo Peters et al. (2018). Older models such as SVMs Cortes and Vapnik (1995) are sometimes also used, typically as part of ensembles. Moreover, lexicons such as HurtLex Bassignana et al. (2018) and Hatebase[2] were also used.

2.3 Settings to Evaluate Data Scarcity

In the introduction, we mention and explain why data might be available to solve a task (e.g., Turkish hate speech detection in Wikipedia), but might be scarce to

[1] https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification

[2] `http://hatebase.org/`

solve a problem (e.g., hate speech detection). In this thesis, we simulate data scarcity in three settings: a query-by-example setting, a zero-shot setting, and a limited-data setting. We provide a background on them in this section. In section 2.4, we review existing techniques to address data scarcity in general. In section 2.5 and section 2.6, we discuss about how different data scarcity settings have been studied in our problem areas, and how different data scarcity solutions were adopted and applied in those settings.

2.3.1 Query by Example (QBE) Setting

In a QBE setting, a user provides a system examples of items they want to see in the retrieved ranked list. The examples constitute the query. QBE approaches are motivated by the fact that it is often easier for a user to express an information need with examples rather than a natural language description (Sarwar and Allan, 2020). It has been applied to retrieve entities and documents from unstructured text corpora (Geng et al., 2022; Smucker and Allan, 2006; Sarwar and Allan, 2019; Sarwar et al., 2019a), entities from knowledge graphs (Metzger et al., 2017), and tuples from relational databases (Fariha et al., 2018).

QBE is related to few-shot learning (Wang et al., 2020b). The few-shot learning setting assumes a classification scenario, where there are n classes and for each class there are k instances or shots. Given this setting any test item needs to be classified into one of these n classes. QBE is a more extreme setting compared to few-shot learning setting because it assumes that $n = 1$ and the number of examples are variable, meaning that k is not fixed.

2.3.2 Zero-shot Setting

In a zero-shot setting, we assume that no training data exists for a task and we have to borrow data from some other similar task, which we refer to as a source task. If we cannot find a similar task, we have to rely on heuristics to perform inference

on a test case. Xian et al. (2019) provide an overview of zero-shot learning from the perspective of image classification. They recommend that for the zero-shot setting it is not sufficient to evaluate performance across tasks because there could be common classes across tasks. For example, if images of dogs are available in both the tasks then classification of images of dogs does not remain a zero-shot problem. In this thesis, we address the binary classification problem, where the class of interest is hate speech or abusive language. The definition and annotation guideline for abusive language changes heavily across tasks. Thus, we assume that the zero-shot setting is implicit in a cross-task transfer as the label definition is subject to change.

2.3.3 Limited-Data Setting

The limited-data setting in this thesis assumes that we have hundreds of training instances from a task, and we can borrow data from other similar tasks. A limited data setting is not as restrictive as a few-shot setting where there are k labeled instances available for each of the n classes. This is because of the number of labeled instances for each class is variable and thus the learning techniques do not depend on a specific input schema. Recent studies on limited-data setting assumes hundreds of labeled data point for training (Du et al., 2021).

In all the above settings, we assume that we can borrow data from other similar tasks, borrow models trained from another task, or generate synthetic data to address data scarcity.

2.4 General Techniques to Tackle Data Scarcity

In this section, we provide a review of the techniques that researchers typically apply to tackle data scarcity – specifically from the neural machine learning perspective. Deep Neural Networks (DNN) are the most effective algorithms to solve supervised learning tasks where the goal of a model to align predictions on data with manu-

ally annotated ground truth (Devlin et al., 2019; Conneau et al., 2020; Raffel et al., 2020). However, the success depends on the amount of labeled data, because neural networks are prone to overfitting with limited data. This motivates deep learning researchers to solve the problem of learning without a large labeled dataset. There are techniques such as data augmentation, transfer learning, weakly-supervised learning etc. to achieve generalization on a task with no or very limited labeled data.

2.4.1 Data Augmentation

Data Augmentation is a process to generate synthetic labeled data to solve a task – from the labeled data from other tasks (Feng et al., 2021; Shorten et al., 2021). Data augmentation techniques aim to increase the diversity of available training data without using human effort in labeling additional data instances (Feng et al., 2021). Data augmentation for natural language processing is challenging in comparison to computer vision because of the discrete nature of language and the difficulty of designing perturbations so that the synthetic data does not have incorrect labels (Feng et al., 2021). On the other hand, because of the discrete nature of language, the augmentations are interpretable. Thus data augmentations are considered as interpretable regularization techniques to control overfitting of neural models. The non-interpretable regularizations include dropout, weight penalties etc (Kukacka et al., 2017).

Typical data augmentations techniques include token-level random perturbation operations including random insertion, deletion, and swap (Wei and Zou, 2019); combining available data instances to create new ones (Guo, 2020); translating the data to a different language and then translating it back to inject noise in data (Longpre et al., 2020); knowledge distillation (Thakur et al., 2021); and using pre-trained language models to generate more instances of labeled data using token replacement (Wu et al., 2019). Shorten et al. (2021) and Feng et al. (2021) provide a complete review on

data augmentation. All these data augmentation techniques are task-agnostic. In this thesis, we address data augmentation for event analysis and abusive content detection, and we provide task-specific augmentation techniques to improve performance on those tasks.

2.4.2 Weak-Supervision

Weak supervision is similar to data augmentation in spirit. While data augmentation focuses on generating different variations from the given labeled datasets, weak supervision is focused towards generating more labeled data from the abundant unlabeled data. Generally data augmentation approaches are applicable across different tasks, and weak supervision is more about leveraging task-specific knowledge from domain experts to label unlabeled data. It is about leveraging higher-level and/or noisier input from subject matter experts (SMEs) (Ratner et al., 2017).

2.4.3 Semi-Supervision

The semi-supervised learning paradigm assumes that there is unlabeled data available from the target task. It trains a model with the available data from the target task, and then uses that model to label unlabeled data from the target task to generate more data (van Aken et al., 2018). It is a special case of weakly supervised learning. In the case of weak supervision, we assume that there are subject matter experts who define functions, rules, and constraints to generate more labeled data. In the case of semi-supervision the model trained from target-task data does it rather than subject matter experts.

2.4.4 Transfer Learning

Transfer learning approaches provide techniques to transfer *useful* knowledge from other *labeled* or *unlabeled* datasets to the target task (Pan and Yang, 2010). The transfer of knowledge takes place through a computational model of the datasets

or the datasets themselves. A computational model parameterized to optimize the likelihood of generating the datasets could directly be applied to the target-task if we are interested in zero-shot transfer. Otherwise, that model might further be trained to generate the data of the target task, which is referred to as *fine-tuning* for the target task. Perhaps, the most popular instance of task-level fine tuning is training of pre-trained language models such as BERT (Devlin et al., 2019) with target-task data. Self-trained language models such BERT provide knowledge of the human language generation process to the target task in a compressed format using a parameterized neural network. When knowledge from such a model is combined with the knowledge about the target task in the form of labeled data, superior performance is achieved compared to only using the knowledge of the target task. We refer to pre-trained language models such as BERT as self-supervised as they obtain implicit supervision from human generated text using masked language modeling (Devlin et al., 2019; Raffel et al., 2020), rather than explicit supervision that requires humans to annotate data to solve a target task. Recent advances in transfer learning showed that it could often be beneficial to fine-tune a pre-trained language model with data from "intermediate" tasks before fine-tuning with target-task data (Phang et al., 2018; Pruksachatkun et al., 2020; Phang et al., 2020).

Another way to incorporate knowledge from different datasets to improve performance on the target task is to use those datasets when training a machine learning model for the target task. This training process is referred to as multi-task learning where a model learns other *auxiliary* tasks in conjunction with the target task. In multi-task learning the loss function is generally a weighted combination of the loss functions of individual tasks. A recent survey of multi-task learning with deep neural networks is provided by Vandenhende et al. (2021).

Unlike data augmentation, semi-supervision and weak-supervision, transfer learning does not focus on generating additional data. It focuses on learning the target

task from the available datasets from other tasks along with the target-task data, if available. Data generation techniques to combat data scarcity are essentially creating additional knowledge which can be fed to transfer learning algorithms. In other words, transfer learning does not take part in data labeling process – rather it consumes knowledge from data from additional tasks to improve on the target task. The additional tasks are often referred to as *source* tasks.

2.5 Techniques to Address Data Scarcity in Event Analysis

Human-labeled training data for event extraction is expensive to produce, has low coverage of event types, and is limited in volume (Wadden et al., 2019; Yang et al., 2019; Chen et al., 2017). Supervised machine learning approaches trained on such datasets are not suitable for large-scale event extraction for knowledge base population. Following distant supervision approaches for entity extraction and relation extraction (Mintz et al., 2009; Min et al., 2013), Chen et al. (2017) proposed an approach based on knowledge-base events (Bollacker et al., 2008) and Framenet (Baker et al., 1998) to automatically generate labeled data for event extraction. However, even though they could increase the number of training data instances for each of the event classes, they couldn't ensure high coverage. Yang et al. (2019) used pre-trained language model as a knowledge-base for event generation. They started with event templates from ACE and then masked event arguments and adjunct words to replace them with similar tokens to generate event sentences. Their approach has the risk of changing the roles of events and modifying event semantics. Hsi et al. (2016) leveraged event annotated data from a resource-rich language along with a parallel corpus to improve event detection on a specific language. Ferguson et al. (2018) applied semi-supervision to compensate for the lack of data. All these approaches improve detection of specific types of events, but do not provide any techniques to learn event similarity.

2.6 Techniques to Address Data Scarcity in Abusive Language Detection

Abusive language is a type of online harm. Even though it is different from clearly illegal activities such as child pornography, it is harmful. Small-scale to large-scale online forums strive to keep their platforms free of abusive language. They develop automatic content moderation systems to flag such contents. Hate speech is a sub-class of abusive language and this thesis addresses both abusive language and hate speech detection problems – from a data scarcity perspective. To be specific, we address cross-domain hate speech detection and cross-language abusive language detection. Our approach for cross-domain hate speech detection exploits the specific properties of a hate speech and proposes to convert negative emotion sentence to hate speech leveraging those properties. Thus we transfer knowledge from sentiment analysis to hate speech detection. Our approach for cross-language abusive language detection is a special type of cross-language text classification framework. In the literature review at first we discuss existing approaches for both the tasks and then provide a discussion on existing transfer learning-based solutions to these problems.

2.6.1 Cross-Domain Hate Speech Detection

Hate speech detection is a relatively recent research area. One of the early papers specifically focused on hate speech (Warner and Hirschberg, 2012) defines hate speech as that containing hateful content directed at a protected group, which is similar to the hate speech template employed in this paper. While there is a growing body of literature on approaches to hate speech detection (c.f. (MacAvaney et al., 2019) and (Schmidt and Wiegand, 2017)), we discuss the literature on data for hate speech detection and domain adaptation, as the focus of this work is data augmentation for hate speech, assuming there is only unlabeled data from the target domain.

The robust labeling approach proposed by Founta et al. (2018) focuses on fine-grained abusive behavior detection, treating it as a multi-class classification problem. They applied several techniques for obtaining robust labels from annotators, but did not apply any automatic approach specifically for hate speech detection. They used random boosted sampling to obtain a large collection of samples for human annotation. We propose an automatic method to generate labeled samples from a large collection of negative emotion sentences (Go et al., 2009), as we wish to reduce the reliance on expensive human annotation.

One of the public data sets labeled for hate speech was introduced in a pair of papers by Waseem (2016) and Waseem and Hovy (2016). This work provided a test bed and a methodology for studying hate speech. Because it is one of the first data sets, it is also one of the most studied, and subsequent work elucidated bias and other issues common in hate speech detection using this collection.

In general most hate speech datasets are biased because of the sampling procedure. Wiegand et al. (2019) demonstrated that a common method for sampling data for hate speech detection (focused sampling) results in datasets biased toward author and topic. Topic bias results in a domain specific dataset. The dataset provided by Waseem (2016) contains tweets mostly about women in sports with a focus on their competence as football commentators. Wiegand et al. (2019) showed that the data contains domain-specific keywords such as *announcer*, *commentator*, *football*, *sports*, occurring frequently in the data as a whole, and specifically in the abusive tweets.

Apart from topic bias, Wiegand et al. (2019) found that two authors **Male tears #4648** and **Yes, They're Sexist** generated more than 70% of the sexist tweets, while a single author **VileIslam** generated 90% of the racist tweets. Overall the authors reported that a focused sampling strategy made the Waseem data domain- and user-style specific. The authors suggested that it is imperative to perform cross-

domain classification to analyze the predictive power of a model constructed from any hate speech data.

To analyze the predictive power of the data set by Waseem (2016), Arango et al. (2019) performed a cross-dataset analysis having the Waseem data as the source and empirically demonstrated the effect of biased training data. They trained a BiLSTM model adopted from Agrawal and Awekar (2018a) on the Waseem data, tested the model on the Semeval dataset (Basile et al., 2019), and discovered an extreme drop in performance. The bias in the Waseem data arises because only 1,590 users write all the tweets in the collection. Moreover, a fine-grained analysis discovered that 491 users generated all the sexist tweets, while only 8 users generated all the racist tweets. Even worse, a single user generated 40% of all the sexist tweets, and another individual user generated 90% of all racist tweets. These findings were consistent with the findings of Wiegand et al. (2019). Waseem (2016) also mentioned that the inter-annotator agreement is $\kappa = 0.84$ and all disagreements occur in annotations of sexism. This suggests that the racist examples were very straightforward and therefore less valuable for training a model.

Arango et al. (2019) showed that cross-dataset performance can be improved by removing bias from the training data and adding data from another source (in this case the hate speech data provided by Davidson et al. (2019)). However, it is not clear whether the performance gain achieved by Arango et al. is caused by de-biasing or augmenting the data. Moreover, this cross-dataset experimentation was not complete. Typically domain-adaptation studies evaluate a model trained from a source domain across more than one target domain.

2.6.2 Cross-Language Abusive Language Detection

Most approaches for abusive language detection use text classification models, which have also shown to be effective for related tasks such as sentiment analysis. This

includes CNNs, LSTMs, BiLSTMs, with or without attention, Capsule networks, and fine-tuned transformers (Georgakopoulos et al., 2018; Badjatiya et al., 2019; Agrawal and Awekar, 2018b; MacAvaney et al., 2019; Arango et al., 2019; Srivastava et al., 2018; Sabour et al., 2017). Differently from what we are proposing, these approaches focus on single data points rather than on their neighborhoods.

Several papers studied the problem of bias in hate speech detection datasets, and have criticized the within-dataset evaluation process (Arango et al., 2019; Davidson et al., 2019; Badjatiya et al., 2019), as this is not a realistic setting, and findings about generalizability based on such experimental settings are questionable. A more realistic and robust evaluation setting was investigated by Glavaš et al. (2020), who show the performance of online abuse detectors in a zero-shot cross-lingual setting. They fine-tuned several multilingual language models (Devlin et al., 2019; Lample and Conneau, 2019; Conneau et al., 2020; Sanh et al., 2019; Wang et al., 2020a) such as XLM-RoBERTa and mBERT on English datasets and observed how those models transfer to datasets in five other languages. Other cross-lingual abuse detection efforts include using Twitter user features for detecting hate speech in English, German, and Portuguese (Fehn Unsvåg and Gambäck, 2018), cross-lingual embeddings (Ranasinghe and Zampieri, 2020), and using multingual lexicon with deep learning (Pamungkas and Patti, 2019).

While understanding the performance of zero-shot cross-lingual models is interesting from a natural language understanding point of view, in reality, a platform willing to deploy an abusive language detection system is almost always able to provide some examples of malicious content to be used for training. Thus, a few-shot or a low-shot scenario is more realistic, and we approach cross-lingual transfer learning from that perspective. We hypothesize that a nearest-neighbor model is a reasonable choice in such a scenario and we propose several improvements over such a model.

2.6.3 Cross-Domain and Cross-Language Text Classification Techniques

As abusive language detection is a text classification problem in general, we provide a survey on cross-domain and cross-language techniques for text classification.

2.6.3.1 Cross-Domain Transfer Learning

Machine learning models assume that the same underlying distribution generates the source and target domain data. However, this assumption is not true for all applications (Daumé III and Marcu, 2006). In fact, it has been shown that the source and the target domains come from different distributions for many tasks including named entity recognition (Lin and Lu, 2018; Tian et al., 2016), sentiment classification (Blitzer et al., 2007), and information retrieval (Cohen et al., 2018; Tran et al., 2019).

Domain adaptation techniques can be classified into *supervised* and *unsupervised* (Daumé III and Marcu, 2006). In terms of supervised approaches, Rizoiu et al. (2019) considered accessing 90% of the source and target domain data to predict 10% of the target domain data, which might not always be practical. Sharifirad et al. (2018) applied a text generation approach based on a knowledge-base to generate more source domain data. For example, their approach replaced a source domain keyword "girl" with the word "woman" using the "Is-A" relationship from ConceptNet. Their generation approach is lexical rather than topical. Moreover, their approach does not leverage unlabeled data from the target domain.

Unsupervised Domain Adaptation (UDA) considers labeled data in a source domain and unlabeled data in a target domain, which more closely reflects "real world" applications (Ruder, 2019). UDA techniques have been applied to many text classification tasks, but most relevant to the current work, sentiment analysis tasks (Xue et al., 2020; Hu et al., 2019; He et al., 2018; Chen and Cardie, 2018; Zhang et al., 2019; Qu et al., 2019). All these approaches focus on extracting domain-independent

features from both source and target domain data, using labels from source domain data to learn a sentiment classifier on the features.

He et al. (2018) devised a semi-supervised approach to use target domain data to train a sentiment classifier. Hu et al. (2019) proposed to distill domain-independent features by adding a domain-dependent task that strips out domain-dependent features. Qu et al. (2019) proposed a category alignment approach to avoid ambiguous target domain features near the decision boundary of the sentiment classifier and achieved state-of-the-art results for cross-domain sentiment classification. We adapted this approach to hate speech detection, and show the performance in experimental results.

While all these approaches focus on learning domain-invariant representations and calibrating classifier decision boundaries to perform better classification in the target domain for sentiment classification, there has been no study of their applicability to unsupervised cross-domain hate speech detection. There are a few studies that report cross-domain performance of different abusive content detection models, but they do not provide any direction to make these models adaptable using unlabeled data from the target domain (Glavaš et al., 2020; Pamungkas and Patti, 2019; Karan and Šnajder, 2018).

Karan and Šnajder (2018) discuss the difficulty of UDA for hate speech detection, in particularly that it is necessary to have some in-domain training data. They did not address the UDA problem and used the Frustratingly Simple Domain Adaptation (FEDA) technique from Daumé III (2007) with labeled data from the target domain.

Waseem et al. (2018) proposed a multi-task learning approach to integrate different datasets into a single training process to construct a generalized hate speech detection model. As this approach also uses labeled samples from all the datasets in both training and evaluation, it does not tackle the UDA problem, where no labeled data from the target domain exists. We create a UDA setting and propose

a data augmentation based UDA approach for hate speech detection that applies semi-supervision on a sentiment analysis data set and does not require learning of domain-invariant features.

2.6.3.2 Cross-Language Transfer Learning

In the cross-language abusive language detection task we have a target-language abusive language detection dataset with a limited number of training examples and a source-language abusive content detection dataset with a large number of training examples. A popular approach to tackle such a cross-lingual learning problem is to fine-tune multilingual language models such as XLM-R (Conneau et al., 2020) or mBERT (Devlin et al., 2019) on the target-language dataset (Glavaš et al., 2020; Stappen et al., 2020). To incorporate the source dataset knowledge, a sequential adaptation technique (Garg et al., 2020) which first fine-tunes a multilingual LM on the source dataset, then on the target dataset, can be used. There are also existing approaches for mixing the source and the target datasets (Shnarch et al., 2018) in different proportions, followed by fine-tuning the multilingual LM on the resulting dataset. While sequential adaptation introduces the risk of forgetting the knowledge from the source dataset, such mixing methods are driven by heuristics that are effective, but not systematic. Moreover, it is not clear how to align the label spaces of the source and of the target datasets. Another problem arises if we consider that new data cases with novel labels can be added to the source dataset (also considered in Chapter 5). In that case, model re-training becomes a requirement in order to map the new label space to the output layer that is used for fine-tuning.

CHAPTER 3

QUERY BY EXAMPLE FOR EVENT RETRIEVAL: A DATA SCARCE SETTING

Event analysis is a major component in computing statistics such as the number of civilians killed by police. If there is no existing database of that information, we gather instances from a large unstructured text collection. The first step is to retrieve target events. In this thesis, we focus on retrieving sentences that mention these events, such as police killing events. A keyword query for retrieving sentences mentioning police killings is insufficient, because bag-of-words features are inadequate for expressing event semantics and structure in queries and exploiting them in candidate sentences Sarwar and Allan (2019, 2020); Halterman et al. (2021). Moreover, we cannot use existing event detection datasets to find target events because of their lack of coverage of different types of events. This is why we believe a Query by Example (QBE) setting is appropriate for sentence-level event mention retrieval. In a QBE setting a query consists of one or more examples sentences that mention the target event type.

In this thesis, we propose the task of *event retrieval* in a QBE setting for the first time in literature. QBE has been applied to retrieve entities and documents from unstructured text corpora (Allan, 2002; Smucker and Allan, 2006; Sarwar and Allan, 2019; Sarwar et al., 2019a), entities from knowledge graphs (Metzger et al., 2017), and tuples from relational databases (Fariha et al., 2018). A QBE setting is data scarce as there are very few examples to understand the semantics and structure of a target event. Furthermore, it is not clear how we can learn an event matching model in an information retrieval setting, because a large collection of (text query, relevant event) or (example event, relevant event) pairs does not exist.

We create three QBE settings from three event annotated corpora: PoliceKilling (Keith et al., 2017), ACE (Walker, 2006), and IndiaPoliceEvents (Halterman et al., 2021). In the section 3.1, we discuss the PoliceKilling-QBE setting where the examples are patient names of police killing events – i.e., civilians killed by police. We show that a logistic-regression based retrieval model created from handcrafted features outperforms traditional retrieval approaches. In the section 3.2, we discuss the ACE-QBE setting where we create example queries using sentence-level mentions of thirty-three different types of events. We show that, surprisingly, a sentence embedding model learned from Natural Language Inference (NLI) corpus designed to compute the similarity between a pair of sentences, transfers to our sentence-level event matching task. To improve the transfer process we apply rule-based event extraction approaches that performs event based segmentation of a sentence which helps in matching a pair of similar events described in two different sentences. This indicates that transfer can become more effective with task-specific knowledge – which is one of our observations in this thesis.

In section 3.3 of this chapter, we describe the IndiaPoliceEvents corpus – an event detection dataset we create to detect five different events where police took part and caused them. We create a IndiaPoliceEvents-QBE setting from this corpus to investigate the effectiveness of different retrieval approaches including our proposed sentence-embedding based one.

Data scarcity is a problem in each of these settings because a vast majority of the event types are different across these settings and thus the knowledge about events is not intuitively transferable across the settings. At the end of the chapter, we provide a unified evaluation of our proposed QBE approach in all these settings. We find that a sentence embedding learned from NLI data is the most effective one when applied with event-specific knowledge.

The work described in this chapter is drawn from one publication at the International Conference on the Theory of Information Retrieval (ICTIR '19) (Sarwar and Allan, 2019), one publication at the Special Interest Group on Information Retrieval (SIGIR '20) conference (Sarwar and Allan, 2020), and one publication at the Association on Computational Linguistics (ACL '21) conference (Halterman et al., 2021).

3.1 QBE on PoliceKilling Dataset

Consider a user searching for *a list of civilians killed by Police*, who issues that query to a search engine. She lands on a web page where she finds the sentence: "*On March 1, 2000, just a few days after a jury acquitted the four police officers who killed* ***Amadou Diallo****, an undercover cop shot and killed 23-year-old* ***Malcolm Ferguson*** *at his Bronx home.*"

Now, the user has one sentence with a couple of positive instances and a query to express her information need. She wants to build a model that would be able to extract more entities like *Amadou Diallo* and *Malcom Ferguson*. Entities such as these typically do not have a Wikipedia page as they are not popular entities. Hence, we cannot adopt resource intensive entity retrieval approaches that depend upon searching through knowledge bases or articles on entities organized by entity categories (Vercoustre et al., 2008). Entity co-occurrence based models would suffer from lower precision if the co-occurring entity is too generic, such as *Bronx* that occurs in numerous contexts (Bron et al., 2010).

Another way to approach this problem is to construct a weakly supervised training dataset and estimate a statistical NLP model (e.g., feature-rich logistic regression, CNN, CRF) (Keith et al., 2017). A weakly supervised dataset is usually constructed by automatically labeling sentences with relevant entities from a knowledge base or a historical list. In the case of our example, the lack of a manually curated historical

database of police killing would make this process infeasible. We propose to construct a retrieval model using extremely limited data and rank sentences based on their likelihood of containing a police killing event and the entities involved with it. We score person entities from the top-k sentences in the ranked list to construct a ranked list of candidate entities.

3.1.1 Retrieval Approach

In this section, we describe **SearchIE**, our retrieval approach for Information Extraction (IE) with extremely limited data. A similar approach was explored by Foley et al. (2018), but it was focused on named entity recognition and did not index long-range features such as different length paths in a dependency parse tree of a sentence. Sarwar et al. (2018) approached a similar problem with term relevance feedback from users which is costly to obtain in practice. We require no feedback from the users in the pre-retrieval stage, and in contrast to Foley et al. (2018) make use of event-specific features. In the next subsections we describe the sentence retrieval and indexing as well as the entity scoring approach.

3.1.1.1 Sentence Indexing

We propose to index sentences by considering extracted NLP features as terms. Even though complex NLP features appear as a sequence of unigram, bigram, POS tag or Named Entity tags, we consider each part of the sequence as a term and index a sentence against them. For example, if a sentence contains two features: "family, NN, TARGET, NNP, shot, VBN", and "PERSON, speaks, to", the sentence is treated as a bag of terms, $B = \{$family, NN, TARGET, NNP, shot, VBN, PERSON, speaks, to$\}$ and the sentence is indexed against these terms. The sequence of these terms is preserved using a positional index that stores the positions of the terms in a document along with the terms themselves. A sample TREC style document with terms as features is shown in Figure 3.1.

The indexing approach is limited to entity types. This study assumes that we are searching for PERSON entities. At the time of indexing a sentence, all the person names in that sentence are replaced with the token PERSON. Finally, each PERSON token is replaced with a TARGET token in turn to create a mention. As a result, we have m mentions of a sentence if there are m person names in that sentence. For each mention in a sentence we extract features and by concatenating all the features from all the mentions in a sentence we create a large "document" from the sentence. We index that document against the DOCNO, and store the person's name against that DOCNO.

```
<DOC>
<DOCNO>1610174_77_0</DOCNO>
<NAME>Rodney Thomas</NAME>
<TEXT>Two years earlier , Officer Rodney Thomas was killed by a hit </TEXT>
<FEATURE>,,,,<punct,killed,VBN,>nsubjpass,TARGET,NNP was,<auxpass,killed,>nsubj
</DOC>
```

Figure 3.1: A TREC document created from a sentence. In this document, DOCNO is the sentence ID, NAME field contains a person name, TEXT field contains the original sentence, and FEATURE field contains the features extracted from the sentence using feature templates shown in 3.2.

3.1.1.2 Sentence Retrieval

Given lexical representations of k example entities $E = \{e_1, e_2, \ldots, e_k\}$, we find the set of sentences $X = \{x_{e_1}, x_{e_2}, \ldots, x_{e_k}\}$, where these entities appear. Note that an example entity can appear in multiple sentences. A mention, $M_{x^j_{e_i}}$ of entity e_i is constructed by taking a single sentence $x^j_{e_i} \in x_{e_i}$ and replacing the entity surface form e_i in that sentence with the token "TARGET". Now, mention $M_{x^j_{e_i}}$ becomes a positive training instance from which we can extract features. We extract the features mentioned in a study of identifying victims of police killing done by Keith et al. (2017). As we use their publicly available dataset, we compute the same features at indexing time and index sentences against those features.

Given the sentence set X we form the training dataset $D_{TR} = \bigcup_{i=1}^{k} \bigcup_{j=1}^{|x_{e_i}|} M_{x_{e_i}^j}$ and use the feature function $f: M_{x_{e_i}^j} \in D_{TR} \rightarrow F$ to generate features from a mention. Then we label all of these mentions as positive with probability $P(Q)$. The negative instances of our training set is also formed by considering all these mentions as negatives with probability $P(1-Q)$. We take this specific approach because our training data is weakly supervised *i.e.* an entity can appear in different contexts in different sentences. Then we learn a logistic regression model on D_{TR}. We use the following objective function that takes into account the weights of the samples:

$$L(\mathbf{w}) = \sum_{j}^{m} \log\Big(1 + e^{-y_j \mathbf{w}^T \mathbf{x}_j Q^{[y_j=1]}(1-Q)^{[y_j=-1]}}\Big) + \lambda \mathbf{w}^2$$

For binary classification, a trained logistic regression model is a vector of weights. We only select a subset of features ordered by their weights and use those features as query to our retrieval system. However, we again create a term based representation of a feature as discussed in Section 3.1.1.1 that turns a feature into a bag-of-words. However, sequences of these words are important as some of the features are generated by traversing a dependency tree. In this case, we take advantage of a widely studied proximity search approach that takes the number of words that can appear between the bag of words in a query as input (Rasolofo and Savoy, 2003).

3.1.1.3 Entity Scoring

For retrieving the entity list we first retrieve the top n sentences using our proposed IR model. Then we simply count the number of occurrences of each of the names in those sentences and rank those names by their frequency. It is easy for us to find those names because the target entities in our dataset are persons and NER taggers are quite accurate in annotating people. However, for arbitrary entity types this

approach cannot currently be applied as several entity type detection from free text is very challenging.

3.1.2 Experimental Setup

In this section we discuss the dataset, our example based query sampling process, and baselines.

3.1.2.1 Dataset

We evaluate our approach on cross-document entity-event extraction for police fatalities dataset created by Keith et al. (2017). The training examples of this dataset are Fatal Encounter (FE) knowledge base (human curated) entities collected from January, 2000 to August, 2016. The goal is to find the names of civilians killed by police in the period (September, 2016 - December, 2016) from Google News data. 258 entities from the FE knowledge base were found in Google news data in that period of time.

Mentions of training examples were found in Google News data (Jan, 2016 - Aug, 2016) and sentences with positive mentions were extracted. Sentences with negative mentions contained person entities that were not available in the FE knowledge base. Even though this approach does not take advantage of all the examples available in the history, it was shown to be sufficient for model training (Keith et al., 2017). As a result, the historical database contained 17,219 civilians and the training example set could only cover 916 of them. A full description of the dataset can be obtained from the work of Keith et al. (2017). The test example set covered 258 entities and their mentions are found from the news corpus of September, 2016 to December, 2016. Sentences that did not contain mentions from the FE database became the negative training data for both train and test splits.

To take the SearchIE approach, we constructed a corpus of 164,871 sentences as the union of all the training and test sentences. We indexed those sentences using the

	Features
$D1$	length 3 dependency paths that include TARGET: word, POS, dep. label
$D2$	length 3 dependency paths that include TARGET: word and dep. label
$D3$	length 3 dependency paths that include TARGET: word and POS
$D4$	all length 2 dependency paths with word, POS, dep. labels
$N1$	n-grams length 1, 2, 3
$N2$	n-grams length 1, 2, 3 plus POS tags
$N3$	n-grams length 1, 2, 3 plus directionality and position from TARGET
$N4$	concatenated POS tags of 5-word window centered on TARGET
$N5$	word and POS tags for 5-word window centered on TARGET

Figure 3.2: Feature Templates (Keith et al., 2017)

Indri Search Framework Strohman et al. (2005). We index both the original sentence and the feature based representation of the sentence. In fact a sentence becomes a large "document" of features and we index sentences against those features (see Section 3.1.1.1 for details on feature index construction). Feature extraction templates are listed in Figure 3.2, taken from Keith et al. (2017).

The index contained approximately 146 million terms among which there were only 87 thousand unique terms. We also constructed a text-only index containing 5 million terms with 76 thousand unique terms. The reason behind constructing a text-only index is to compare the performance of corresponding feature based index in terms of extraction performance.

3.1.2.2 Query Construction

Our queries are examples – names of civilians in the context of this dataset. We randomly sample 30 names from a set of all the civilian names in the training (916) and test (258) data. Then we create 50 k-example queries by a random selection from $\binom{30}{k}$ possibilities. As a result, we have 50 queries for number of examples ranging from 1 to 30 – resulting in 1500 queries.

At the time of evaluation, for SearchIE and all other baselines, no credit was given to a system for retrieving entities belonging to the set of examples since the examples are already known. In this work, we only consider entity level novelty.

3.1.2.3 Baselines

We experiment and compare the effectiveness of SearchIE with both ad-hoc IR (Information Retrieval) and IE (Information Extraction) baselines. We considered Query Likelihood (QL) (Ponte and Croft, 1998) and Relevance Model 3 (RM3) (Abdul-Jaleel et al., 2004) as IR baselines, we also used the model proposed by Keith et al. (Keith et al., 2017) as our IE baseline. For convenience, we refer to this model as **Weak-LR**: a logistic regression model that is trained on weakly supervised data. The performance of Weak-LR is driven by a soft labeling approach, which assumes a mention sentence to be positive with some confidence. Even though Weak-LR is the state-of-the-art for this dataset, it was not designed for and has not previously been tested in the limited examples scenario.

Our baseline models take different types of inputs based on their solution approach. IR models take user-specified keywords concatenated with examples as query. We used three keywords for the user-specified query: *civilians*, *police*, *killed*. Weak-LR and SearchIE takes only examples as input. The output of SearchIE and other IR approaches is a ranked list of sentences, from which a ranked list of entities is computed using the approach of Section 3.1.1.3. Weak-LR outputs probabilities for all the mentions generated from a sentence and we perform mention level aggregation to generate a score for that sentence. Given m mentions generated from a sentence, the probability for each of those mentions is computed, and the maximum of those probabilities is selected as the score for that sentence. Finally, sentences are ordered based on scores and an entity ranked list is constructed using the same frequency

based aggregation approach we used for SearchIE and all other baselines to ensure fair comparison.

3.1.3 Experimental Result

Feature Effectiveness We ranked the features based on their weights estimated from our Logistic Regression model. Some of the highest ranked features resulted from training with 30 examples are: (TARGET, TARGET O, police, TARGET NN, shot, TARGET NNP, police NN, officers NNS, killed VBN). Some of the lowest ranked features from the same model are: (PERSON NN Talks NNS TO, county NNP courthouse NN, supporters NNS, of cumberland county, supporters 18 on 17, talks to supporters, PERSON talks to, vigil NN case following VBG, steps NNS det the DT). The highest ranked features are more general – recall oriented. The lowest ranked features, which we reject at the time of forming the search query, are very specific and comprise long sequence of nodes in dependency path trees. Though they might be useful for making decision about a mention they are not useful for ranking.

Effect on the Number of Examples Figure 3.3 shows the effect of adding more examples with SearchIE and other baselines. SearchIE supersedes the baselines both for very limited number of examples and as the number of examples increase. Please note that we only used the 200 highest weighted features regardless of the number of examples to generate this figure. The SearchIE approach has top performance and it generally becomes better as more examples are provided. The Weak-LR approach is surprisingly unstable, varying substantially with different numbers of examples. We have shown 95% confidence interval for the performance metrics, illustrating that Weak-LR is has wider intervals in general, also supporting the hypothesis that it is more sensitive to the specific set of examples selected.

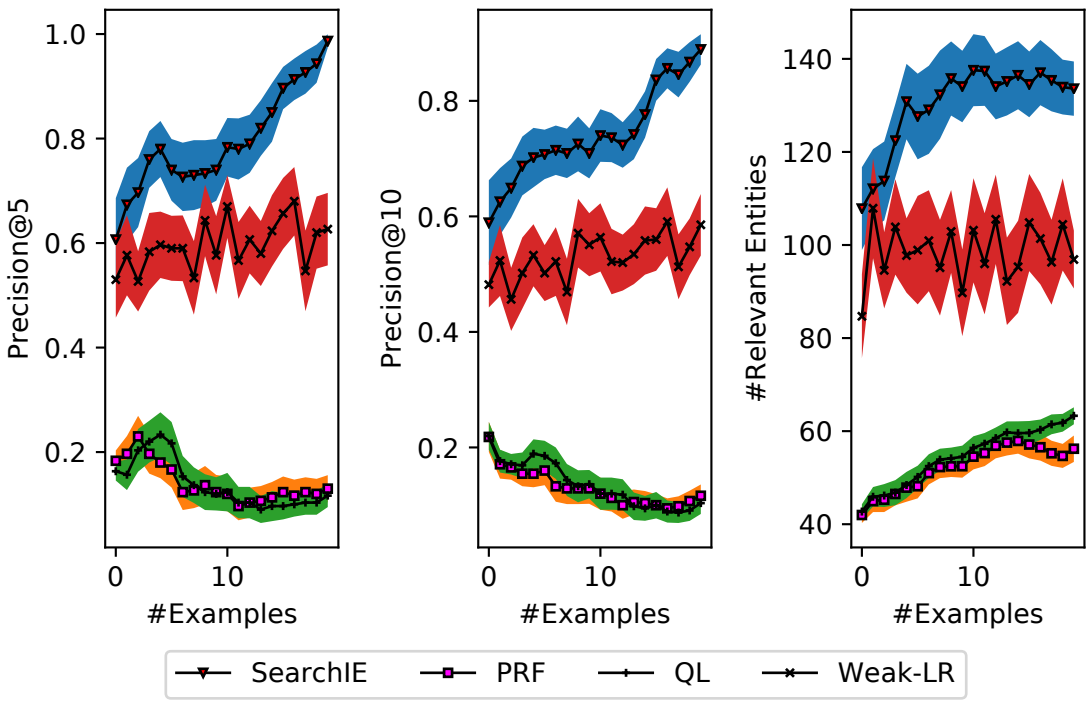


Figure 3.3: Effect of including more examples

3.2 QBE on ACE Dataset

In the PoliceKilling dataset, we created queries with examples of police killing events by using the names of the killed civilians to find those examples. On the other hand, the ACE dataset Walker (2006) contains event annotations based on thirty-three types of events. We create queries with event examples by directly sampling sentences from these classes. Thus the ACE setting is directly based on events rather than entities. Below we describe an example of QBE setting from the ACE dataset and the associated retrieval challenges.

Consider the case where a user wants to find all the *jail release* events from a corpus. To start this process, she retrieves a few documents with combination of keywords such as *jail*, *release*, *sentence*, etc., and finds sentences from those documents that mention a jail release event. Although these sentences constitute a representation of her information need (query), traditional retrieval approaches do not provide

support for such an event query. A sentence matching model that computes similarity between a pair of sentences can be a remedy to this problem. However, our experiments suggest that the performance of a state-of-the-art unsupervised sentence matching model is sub-optimal for event matching – without the integration of event-specific knowledge.

We study the event matching problem in a cross-lingual setting – i.e., we assume that the language of example sentences and corpus sentences are different. Although Cross-Lingual Information Retrieval (CLIR) is a well-studied problem, CLIR studies are targeted towards document retrieval (Sarwar et al., 2019b; Galuscáková et al., 2021; Nie, 2010). To the best of our knowledge, there is no study or available testbed for studying CLIR or even mono-lingual IR for example-driven event retrieval. Such a setting would be very useful for journalists, security agency personnel, and political scientists. This motivated us to create a testbed and evaluate standard retrieval approaches for our task, Cross-Lingual Event Retrieval with Query by Examples (CLER-QBE).

To solve CLER-QBE, we follow a popular CLIR approach that uses two stages: query translation and retrieval (Nie, 2010). We translate the example sentences that constitute our event query using a commercial Machine Translation (MT) system and focus on the retrieval problem. It is challenging to retrieve sentences containing a target event with translations of examples sentences for two reasons: i) translated example sentences are noisy because of MT error; ii) only a sub-sequence of tokens in the translated example sentences describes the target event that holds for corpus sentences too. Both these issues make it challenging to understand user intent and match event mentions in translated examples and corpus sentences. They result in a phenomenon we refer as *noisy matching*.

To alleviate the effect of the noisy matching problem, we assume we have event trigger annotation for our example sentences. Consider the sentence: "Pasko, whose

sentence included time served, was **released** in January for good behavior after serving more than two-thirds of the sentence." Note the mention of three events: sentence, jail release, and sentence serving completion. We assume that a user interested in the jail release event would provide us with the trigger keyword *released* along with the example sentence so that we can extract appropriate context around the trigger to understand the user intent. Note that knowing trigger words in the examples does not solve the problem because we still need to isolate the target event from all other events in the document that could contain more than one event.

To extract event extents from documents and match them with query context we use PredPatt, an unsupervised technique for Semantic Role Labeling (SRL) (Zhang et al., 2017). PredPatt identifies the predicates and their corresponding arguments from a sentence. We use that information to predict event spans in documents. Once the document event spans are identified, we match them with query context using a recently proposed Sentence-BERT (SBERT) model (Reimers and Gurevych, 2019a). The original BERT model does not provide effective out-of-the-box sentence embeddings without fine-tuning (Reimers and Gurevych, 2019a). SBERT is fine-tuned with Natual Language Inference (NLI) data and it is able to create sentence embeddings that significantly outperforms other state-of-the-art models on semantic textual similarity tasks. Finally, to describe our contributions concisely, we propose the task of CLER-QBE, construct a standard testbed, evaluate classical retrieval approaches on that, and propose an effective SRL-based technique to predict document event spans as well as an unsupervised matching model to match query context with the predicted spans.

3.2.1 Problem Formulation

$Q_e = \{s^1_{src}, s^2_{src} \ldots, s^n_{src}\}$ is an event query that consists of n example sentences mentioning a target event, $e = \{s^1, s^2, \ldots, s^n\}$ in *src* language. For example, $Q_{\text{jail release}}$

$= \{s^1_{Arabic}\}$ indicates that a user has provided an example sentence describing a *jail release* event in Arabic and wants to retrieve sentences describing *jail release* events in another language. Q_e is issued against a corpus, $D_{trg} = \{d^1_{trg}, d^2_{trg}, \ldots, d^m_{trg}\}$ of m sentences written in *trg* language. There is a relation, $Event(d^i_{trg}) \subset E = \{e_1, e_2, \ldots e_l\}$ that maps a sentence d^i_{trg} to a set of events, E. We assume query event $e \in E$ for the sake of evaluation. $Event(x) = \emptyset$ indicates that x does not mention any event. The task is to retrieve a ranked list $R = (d^1_{trg}, d^2_{trg}, \ldots, d^k_{trg})$ of k sentences mentioning e. A sentence d^i_{trg} in the ranked list is relevant if $e \in Event(d^i_{trg})$; otherwise it is non-relevant.

Our problem assumes that the user has annotated example sentences with event *triggers*, based on event detection literature where an event mention contains a main word or phrase that evokes the event (Lai and Nguyen, 2019; Reimers and Gurevych, 2018). To illustrate this we provide an example from our dataset: "Pasko, whose **sentence** included time served, was **released** in January for good behavior after **serving** more than two-thirds of the sentence." This example actually describes three events: i) *Pasko* was sentenced, ii) he was released from jail, and iii) he served time in a jail. If the user annotates the example sentence with the keyword *released* it probably means that she is looking for jail release events. As we have example sentences as well as user annotated triggers, we use $Q_e = \{s^1_{src}, s^2_{src}, \ldots, s^n_{src}\}$ and $Q^t_e = \{t^1_{src}, t^2_{src}, \ldots, t^n_{src}\}$ as sentence query and trigger query, respectively. Sentence and trigger queries based on the above example would be $Q_{\text{jail release}} = \{\text{Pasko, whose ... released ... sentence.}\}$ and $Q^t_{\text{jail release}} = \{\text{released}\}$.

3.2.2 Approach

Our approach consists of four components: *Query Translation*, *Document Scoring*, *Matching Model* and *Event Span Detection*.

Query Translation One common practice in cross-lingual information retrieval is to translate a search query using an off-the-shelf MT model, and perform mono-lingual retrieval using the translated query (Nie, 2010). We take the same approach – i.e., we translate Q_e and Q_e^t into target language using a Google's online MT model to obtain $\widetilde{Q}_e = \{\tilde{s}_e^1, \tilde{s}_e^2, \ldots, \tilde{s}_e^n\}$ and $\widetilde{Q}_e^t = \{\tilde{t}_e^1, \tilde{t}_e^2, \ldots, \tilde{t}_e^n\}$, respectively.

Document Scoring Now that our sentence and trigger queries are translated into the target language, we use a mono-lingual sentence matching model, M_s, to compute similarity between our queries and documents. Given M_s, a sentence matching model we compute the score of a document in the target language as, $score(d_{trg}^i) = \sum_{\tilde{s}_e^j \in \widetilde{Q}_e} M_s(\tilde{s}_e^j, d_{trg}^i)$. Similarly, we use a model M_t to match triggers with corpus sentences and compute similarity scores using $score(d_{trg}^i) = \sum_{\tilde{t}_e^j \in \widetilde{Q}_e^t} M_t(\tilde{t}_e^j, d_{trg}^i)$. Sorting the documents using the scores computed by each model results in two ranked lists that we combine using the reciprocal rank fusion approach (Cormack et al., 2009). The intuition behind combining lists is that they capture different aspects of matching. The trigger matching model does not include context while the sentence matching model includes it.

Matching Model Our trigger matching model, M_t, is query likelihood approach. As triggers do not contain any contextual information, unigram statistics are sufficient to establish matching. As sentence matching model, M_s, we use Sentence BERT (SBERT) (Reimers and Gurevych, 2019a). SBERT adds a pooling operation to the output of BERT to derive a fixed sized sentence embedding. Similar to the authors we use the mean pooling strategy to compute a fixed size representation for sentences. With a fixed size representation of a pair of sentences we use cosine similarity to compute the similarity between them. However, one problem with event retrieval that is a sentence usually mentions more than one event, which holds for both query and document sentences in our setting. To match the query event with the document

event accurately we focus on the relevant part of the example sentence and the corpus sentence. The next section describes how we find these relevant parts.

Event Span Detection Given $\widetilde{Q}_e$ we compute matching scores of each $\tilde{s}_e^j \in \widetilde{Q}_e$ with each $d_{trg}^i \in D_{trg}$ using M_s. Before doing that we need to consider that a target event e is usually mentioned by a subsequence of tokens in the example sentence $\tilde{s}_e^j$. Considering the entire sentence as the search intent would result in noisy matching. To alleviate this problem we locate the trigger $\tilde{t}_e^j$ in $\tilde{s}_e^j$ and take a window of information around $\tilde{t}_e^j$. As $\tilde{t}_e^j$ and $\tilde{s}_e^j$ are translations of t_e^j and s_e^j, sometimes $\tilde{t}_e^j$ cannot be located in $\tilde{s}_e^j$ even if t_e^j appears in s_e^j. In that case we compute word embedding similarity of $\tilde{t}_e^j$ and all others tokens in $\tilde{s}_e^j$ and select the location of the highest scored token. Assuming the location is l, we consider a token span starting from $l - w$ to $l + w$ to capture a window w of tokens around the translated event trigger. We refer to this token span as query context.

In order to find event spans in a document we use a Semantic Role Labeling Approach (SRL) to find predicate argument structure from a sentence. Given a sentence, SRL is used to answer basic questions about sentence meaning, including "who" did "what" to "whom," etc (Carreras and Màrquez, 2005). We use an unsupervised SRL approach, Predictive Patterns (PredPatt) (White et al., 2016), to find predicate and arguments and use those to predict event spans from documents. PredPatt is lightweight, fast, and unlike other supervised SRL approaches, it does not need to adapt to a target domain with further training (Hartmann et al., 2017; Zhang et al., 2017). It uses a set of non-lexicalized, extensible and interpretable patterns on the Universal Dependency (UD) (de Marneffe et al., 2014) parse of a sentence to extract predicates and arguments. PredPatt with UD is able to extract predicate and arguments in almost any language.

To illustrate how we use PredPatt to predict event spans, consider the example provided in our problem definition section: "Pasko, whose **sentence** included time

served, was **released** in January for good behavior after serving more than two-thirds of the sentence." The predicates and their corresponding arguments found by running PredPatt on the example are shown in Table 3.1. We predict event spans by considering the minimum size token window that covers a predicate and all its arguments. As a result, a document d^i_{trg} is decomposed into f token spans i.e. $d^i_{trg} = \{d^{i1}_{trg}, d^{i2}_{trg}, \ldots, d^{if}_{trg}\}$. In order to compute the score of d^i_{trg} with respect to example sentence $\tilde{s}^j_e$ we take the maximum of the scores of the token spans.

Table 3.1: Event Span Prediction Using PredPatt (Zhang et al., 2017)

Predicate	Arguments	Predicted Event Spans
included	{sentence, time}	sentence **included** time
released	{Pasko}	Pasko , whose sentence included time served , was **released**
serving	{two-thirds}	**serving** more than two-thirds

3.2.3 Experimental Setup and Results

3.2.3.1 Dataset Construction

We adopt the ACE 2005 multilingual event detection dataset provided by the Linguistic Data Consortium (Walker, 2006) to evaluate CLER-QBE. ACE 2005 provides sentences in *English*, *Arabic*, and *Chinese* and each sentence is human annotated with zero or more event types from thirty-three event types defined in ACE guideline. Trigger words or phrases are also provided along with the corresponding event type annotations. We pre-processed the original ACE 2005 dataset[1] with the help of English, Arabic and Chinese lanuage processing libraries from Stanford CoreNLP (Manning et al., 2014). Table 3.2 provides a few frequent event types along with the number of sentences mentioning them from our processed version of ACE.

[1] `https://github.com/nlpcl-lab/ace2005-preprocessing`

Our processed version of ACE contains 16249, 1458, and 2088 sentences in English, Chinese, and Arabic, respectively. Among them 3884, 487, and 2059 sentences are annotated with at least one event type. As English has the largest number of sentences, we construct our retrieval corpus from English. To create queries, we assume each event type as a query and randomly draw Arabic and Chinese example sentences for that event type. Relevance judgments for English sentences for any query event type are created using event type annotations provided by ACE.

3.2.3.2 Experimental Setting

We use the *Indri* search framework to index our English corpus. We use existing implementations of PredPatt[2] for SRL and SBERT[3] for matching. We use TrecTools[4] to evaluate our retrieval runs and perform reciprocal rank fusion. We use a window size of five around the trigger words in example sentences to determine query context. Our adopted ACE dataset and source codes to generate all the experimental results are available [5].

Table 3.2: Highly occurring events in ACE with the number of sentences describing them in different languages

Event Type	English	Chinese	Arabic
Movement:Transport	713	99	392
Conflict:Attack	1510	74	455
Contact:Meet	280	44	190
Transaction:Transfer-Money	187	24	42
Life:Die	584	34	213

[2]https://github.com/hltcoe/PredPatt

[3]https://github.com/UKPLab/sentence-transformers

[4]https://github.com/joaopalotti/trectools

[5]`https://github.com/sarwar187/multilingual-event-retrieval/tree/predpatt-integration`

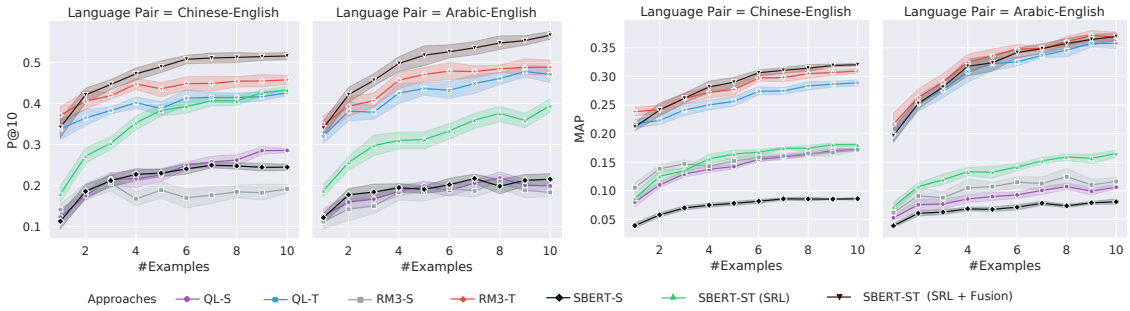


Figure 3.4: Retrieval performance in terms of Precision@10 and MAP for two language pairs with increasing number of examples. We randomly sample ten sets of k-examples query and plot the mean with 95% confidence interval.

3.2.3.3 Experimental Results

We report retrieval performance in terms of Precision@10 and Mean Average Precision (MAP) on the ACE English retrieval corpus using Chinese and Arabic queries containing different number of example sentences. We use three retrieval approaches: QL (Query Likelihood), RM3 (Relevance Model 3) and Sentence BERT (SBERT) (Reimers and Gurevych, 2019a) and three different example query types: sentences (S), triggers (T), combined (ST). The process of constructing a combined (ST) query is illustrated in section 3.2.2 and we use it with SBERT matching model. As our proposed query construction method includes an SRL component, we refer to this approach as SBERT-ST (SRL). Thus we have five baseline approaches: QL-T (QL with Trigger Query), QL-S (QL with Sentence Query), RM3-T, RM3-S, SBERT-S, along with two proposed approaches SBERT-ST (SRL) and SBERT-ST (SRL + Fusion). SBERT-ST (SRL + Fusion) is the reciprocal rank fusion of RM3-T and SBERT-ST (SRL). Note that QL-S and RM3-S do not directly support sentence queries. Hence, we construct a bag-of-words query from the example sentences by extracting unique terms from them. All the Chinese and Arabic sentences as well as trigger queries were translated by Google MT [6].

[6]https://cloud.google.com/translate

Figure 3.4 reports the precision@10 and Mean Average Precision (MAP) for retrieval with Chinese and Arabic Queries with increasing number of examples. One important thing to note that trigger queries (QL-T, RM3-T) result in much better performance than sentence queries (QL-S, RM3-S). It happens because we have a small retrieval corpus and we do not lose precision by matching ambiguous triggers. For example, there is less chance of matching a *sports attack* event than a *military attack* event with keyword *attack* as a query. The failure of the baseline sentence query approaches (QL-S, RM3-S, SBERT-S) is explainable by the noisy matching phenomenon that happens when the entire example and document are considered for matching. Our proposed approach SBERT-ST (SRL) outperforms all the baseline approaches with sentence queries in terms of Precision@10 for any number of examples. We observe gain in MAP for Arabic queries, while for Chinese queries this gain is achieved with more than four examples. Finally, to combine the strength of trigger and paragraph queries, our propose SBERT-ST (SRL + Fusion), which is a reciprocal rank fusion of RM3-T and SBERT-S (SRL), outperforms all the baselines in terms of P@10. Improvement in MAP is also observed but not for Arabic queries.

3.3 QBE on IndiaPoliceEvents Dataset

In this section, we describe the construction of a new dataset IndiaPoliceEvents that we use for evaluating our QBE approach. This dataset is collected with a motivation to create an evaluation benchmark for social scientists so that they can evaluate total recall for event retrieval task. The corresponding paper has been published in the Association of Computational Linguistics (ACL '21) conference (Halterman et al., 2021).

This dataset contains sentence level relevance judgments for five questions about events in which police took part or was an agent. One query is borrowed from the PoliceKilling dataset and it is *Did police kill someone?*. For the annotation, we fixed

a corpus, hired annotators to annotate each sentence in the corpus given the context of the document from which we draw the sentence. The annotators were asked to judge the relevance of a sentence based on each of the five questions about police activity. We describe the annotation process and the dataset in the next section.

Given a collection of sentences judged based on five queries, we randomly select five relevant sentences for each of the queries and remove them from the collection. From a set of five sentences for a query, we randomly select one-, two- and three-example sentences to constitute our QBE setting. For each length we repeat the process to get five samples per length. For example, for the police killing query, we sample, five one-length queries and use them to retrieve police killing sentences. The results are reported in Table 5.2 along with the results from all the other event retrieval settings we explored.

3.3.1 Annotations and Dataset

We curate our corpus with a substantively motivated specification: it is restricted to a single authoritative news source, over a defined span of time, with articles that mention one of two locations involved in or related to the 2002 Gujarat violence.

From the website of *Times of India*, an English language newspaper of record in India, we first download all news articles published in March 2002. During this period, widespread communal violence occurred in India, following the death of 59 Hindu pilgrims in a train fire in the state of Gujarat. In the subsequent months, reprisal attacks were directed at mostly Muslim victims across the state (Human Rights Watch, 2002; Subramanian, 2007). In creating our annotations, we specifically focus on the actions of police during these events, since a large body of evidence points to the importance of police intervention and non-intervention in quelling or permitting ethnic violence (Human Rights Watch, 2002; Wilkinson, 2006; Subramanian, 2007). We focus on the first month of the violence in order to fit within our annotation

budget. This month saw the greatest levels of violence, though violence continued for a period of months afterward.

Our final corpus consists of the subset of scraped documents published in March 2002 that include either the name of the state (*Gujarat*) or a city related to the beginning of violence (*Ayodhya*).[7] Selecting on geographical and temporal metadata is a high recall way to filter the corpus without biasing the dataset by filtering to topic or event-related keywords, thus giving a better view of the true recall of an event extraction method.

Event Class	Pos. Sents.		Pos. Docs.	
KILL	96	(0.45%)	50	(3.98%)
ARREST	299	(1.40%)	128	(10.17%)
FAIL TO ACT	207	(0.97%)	114	(9.05%)
FORCE	222	(1.04%)	90	(7.15%)
ANY ACTION	2,073	(9.69%)	457	(36.24%)

Table 3.3: IndiaPoliceEvents number and percentage of positive sentences (sents.) and documents (docs.) after the adjudication round. In total, the dataset contains 21,391 sentences and 1,257 documents.

3.3.1.1 Annotations via natural language

To collect annotations, we give annotators an entire document for context, and then ask them *natural language questions* about semantic event classes anchored on the actions of police for each sentence in that document:

- KILL: "Did police kill someone?" Lethal police violence is an important subject for social scientists (Subramanian, 2007). Example sentence: *"In Vadodara, one person was killed in police firing on a mob in the Fatehganj area."*

[7]Selecting documents using location-based keywords is a standard first step in political science text analysis (Mueller and Rauh, 2017). This filters to 18% of the total articles in March 2002. The precipitating event for the March 2002 violence was the burning of a train of pilgrims returning from Ayodhya.

- ARREST: "Did police arrest someone?" Knowing when and where police made arrests and who was arrested is an important part of understanding police response to communal violence. Example sentence: *"Police officials said nearly 2,537 people have so far been rounded up in the state."*
- FAIL TO ACT: "Did police fail to intervene?" In the 2002 Gujarat violence, police were often accused of failing to prevent violence or allowing it to happen. Knowing when police were present but did not act is important for understanding the extent of this phenomenon and its potential causes (Wilkinson, 2006). Example sentence: *"The news items [...] suggest inaction by the police force [...] to deal with this situation."*
- FORCE: "Did police use force or violence?" Political scientists are interested not only when police kill but the level of force they use. Example sentence: *"Trouble broke out in Halad [...] where the police had to open fire at a violent mob."*
- ANY ACTION: "Did police do anything?" We collect annotations on all police activities, so that social scientists could, in the future, label more fine-grained event classes. Example sentence: *"In the heart of the city's Golwad area, the army is maintaining a vigil over mounting tension following [...]"*

Figure 3.5 shows the interface annotators see. While the first three classes each correspond to a single annotation question, we create FORCE and ANY ACTION by taking the union of several different questions posed to annotators, which made it easier for annotators to distinguish between different subtypes. FORCE is the union of "Did police kill someone?" and "Did police use other force or violence?". ANY ACTION is the union of four questions: "Did police kill someone?", "Did police arrest someone?", "Did police use other force or violence?", and "Did police do or say something else (not included above)?".

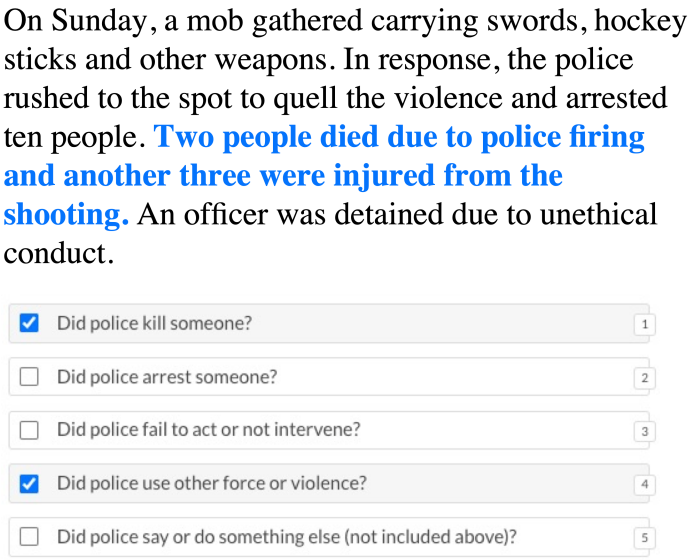


Figure 3.5: We present annotators with a highlighted sentence (blue) and its document context. Their task is to click a check-mark for the event-focused questions for which there is a positive answer in the highlighted sentence.

Following the guidelines of Pustejovsky and Stubbs (2012), we first assign each document to two annotators and then follow with an *adjudication round* in which items with disagreement are given to an additional annotator to resolve and create the gold standard. For annotators, we select undergraduate students majoring in political science (as opposed to crowdworkers) in order to approximate the domain expertise of social scientists.[8] We initially recruited and selected 12 students. After a pilot study and two rounds of training, in which we provided individual feedback to annotators via email, we selected 8 final annotators based on their performance. Each student annotated around 330 documents (∼5,500 sentences).

Table 3.3 shows the prevalence of the event classes after the adjudication round. Note that some of the classes are relatively rare: of all documents, only roughly 4%

[8]Our annotation protocol (no. 2238) was reviewed as exempt by the University of Massachusetts Amherst's IRB office. Annotators were paid $25 per training session and a lump sum for document annotations; we expected this to exceed $14 USD per hour based on a generous (conservatively high) estimate of completion time. All annotators reported their work time was less than this estimate.

Dataset Name	Approach	1-example	2-example	3-example
PoliceKilling-QBE	QL	0.22	0.18	0.17
	RM3	0.22	0.17	0.16
	SBERT	**0.32**	**0.26**	**0.22**
	SBERT-ST (SRL)	0.26	0.2	0.18
IndiaPoliceEvents-QBE	QL	0.04	0.03	0.05
	RM3	0.06	0.08	0.07
	SBERT	0.28	0.48	0.54
	SBERT-ST (SRL)	**0.37**	**0.58**	**0.7**
ACE-QBE (Ch-En)	QL	0.12	0.18	0.22
	RM3	0.14	0.17	0.22
	SBERT	0.11	0.19	0.23
	SBERT-ST (SRL)	**0.18**	**0.28**	**0.31**
ACE-QBE (Ar-En)	QL	0.13	0.16	0.17
	RM3	0.12	0.14	0.15
	SBERT	0.13	0.18	0.19
	SBERT-ST (SRL)	**0.19**	**0.26**	**0.29**

Table 3.4: Comparison of lexical and semantic event-retrieval approaches in terms of precision@10 on the retrieval settings created from three event-detection datasets. In all the datasets our proposed approach SBERT-ST (SRL) (details in 3.2.2 and 3.2.3.2) outperforms the baselines.

have KILL and 7% have FORCE. Our annotators had fairly high inner-annotator agreement for KILL and ARREST, with Krippendorff's alpha values of 0.75 and 0.71 respectively. Other questions, such as FAIL TO ACT and "Did police use other force?" had lower agreement ($\alpha < 0.4$), indicating more difficulty and ambiguity.

3.3.2 A Unified Evaluation Three QBE setting

We conduct a unified evaluation our three QBE settings: PoliceKilling-QBE, IndiaPoliceEvent-QBE, and ACE-QBE. The ACE-QBE setting has two different sets of queries: Ch-En and Ar-En. In the Ch-En setting, we create QBE examples from the Chinese annotated events from ACE and use that to retrieve events from the English corpus. For the Ar-En setting, we create QBE examples from ACE Arabic language pack and retrieve from English corpus. This is because the English anno-

tated corpus contains the largest number of annotated sentences to be a challenging retrieval corpus.

The experimental results for all the QBE settings are shown in Table 3.4.

3.4 Summary

We proposed three QBE settings. For the PoliceKilling setting, we show we can effectively construct a query from a few examples of police killing events by extracting and weighting handcrafted NLP features. For the ACE setting we show that a sentence-embedding based approach based on SBERT (Reimers and Gurevych, 2019a) transfers to event retrieval – when we segment each of corpus sentences using Pred-Patt to obtain events from sentences. We have created the IndiaPoliceEvents dataset for social scientist by annotating event mentions in sentences into five classes. We constructed QBE-IndiaPoliceEvents setting and provided evaluation of our sentence-embedding based approach on all the three QBE settings. We compared our approach with strong baselines and showed that our approach outperforms them.

CHAPTER 4

ZERO-SHOT HATE SPEECH DETECTION

Online harassment in the form of hate speech has been on the rise in recent years. Addressing the issue requires a combination of content moderation by people, aided by automatic detection methods. As content moderation is itself harmful to the people doing it, we desire to reduce the burden by improving the automatic detection of hate speech. Hate speech presents another challenge as it is directed at different target groups often using a completely different vocabulary. Further the authors of the hate speech are incentivized to disguise their behavior to avoid being removed from a platform. This makes it difficult to develop a comprehensive data set for training and evaluating hate speech detection models because the examples that represent one hate speech domain do not typically represent others, even within the same language or culture.

We propose a novel data augmentation approach as an Unsupervised Domain Adaptation (UDA) technique for hate speech detection. We assume a zero-shot setting for the target task – i.e., we assume labeled data for the tasks is not available. However, we assume that we have access to unlabeled data from the target task and labeled data from a source task. In the literature, this particular setting has given rise to a number of UDA techniques where researchers use the unlabeled data from the target task to make the distribution of the source-task data and the target-task data closer (Ganin and Lempitsky, 2015). Note that existing methods in literature use the term domain adaptation rather than task adaptation to refer to UDA. In the introduction, we discussed that two tasks can differ in terms of domain from where

the data is sourced, language of data and annotation guidelines. Thus, by using the term "task" we create a space for new problems in the UDA literature. But, to compare with existing solutions, and to limit the scope of our study, we only focus on domain difference in this chapter. As a result, we use the terms *domain* and *task* interchangeably in this chapter. Generally, we use the terminologies in the UDA literature to be consistent with existing work.

Our UDA approach for hate speech detection augments labeled data that is close to the data distribution of the target domain. As our target domain is data-scarce, we propose a synthetic data generation approach that considers labeled data from the source domain and unlabeled data from the target domain to generate more labeled data that is similar to the target domain. The unlabeled data from the target domain helps us to capture the distribution of hate speech vocabulary in the target domain and use that for more data generation. We contribute a novel data generation method, while we assume a simple transfer approach based on fine-tuning. Note that this data generation approach is specific to hate speech detection, because we exploit certain characteristics of hate speech sentences to create hate speech templates for instantiating hate speech. One observation of this thesis is that it is crucial to provide problem-specific treatment to the data generation and transfer learning and this chapter is an instance of this observation from the generation perspective.

We evaluate the effectiveness of our data augmentation approach with three different models (character CNNs, BiLSTMs and BERT) on three different collections. We show that our approach improves Area under the Precision/Recall curve by as much as 42% and recall by as much as 278%, with no loss (and in some cases a significant gain) in precision. The work described in this chapter is drawn from an accepted publication at the International Conference on Web and Social Media (ICWSM) (Sarwar and Murdock, 2022).

4.1 Proposed Cross-Domain Adaptation Technique

As mentioned in the introduction, hate speech detection has a bias problem where a classifier might learn the hate speech vocabulary and usage patterns of a very small number of people, and be unable to generalize to hate speech in a new domain, directed at other groups. One solution is to limit the contribution of any given individual to the dataset as shown by Arango et al. (2019). We found that increasing the amount of training data is also effective even without limiting the contribution of an individual (further discussed in Section 4.3.2). However, neither solution solves the problem of adapting to a new domain. We propose an Unsupervised Domain Adaptation (UDA) approach that both augments the training data, and adapts to the target vocabulary.

Problem Setting We have a source domain hate speech dataset D^s with labeled examples, and unlabeled data D^t_u from the target domain. The task is to train a hate speech detection model using D^s and D^t_u. We evaluate it on the labeled data from the target domain D^t_l.

We augment the source domain dataset, D^s, with domain-adapted hate speech in the target domain. We describe the process in detail below, and an example sentence transformation for each step is shown in Table 4.1. In the example, we did not want to use any actual hate speech so the text does not become disturbing to the reader. In place of a profane of hateful term we use the term *boring* which expresses an opinion. Such cases do not appear in actual data processing. We also use "Honda CRV" instead of a race, gender, ethnicity or any target-group related term so that we do not end up offending a reader by chance.

4.1.1 Learning a Tagger From the Source Domain Data

We define *context carriers*, which contain useful patterns from which a variety of hate speech can be generated. For example in the sentence "The problem with Honda

Symbol	Explanation	Example
D^s_{hate}	Hate Speech in the source domain	The problem with Honda CRV's is that they are boring
H^s	Source domain (external) hate lexicon of OTG tokens	honda, crv, boring
	Context carrier	The problem with ... is that they are ...
$\widetilde{D}^s$	Templatized sentence used to train an OTG tagger	The problem with REP is that they are REP
D^t_u	Unlabeled data from the target domain	Bananas are very yucky!
H^t	Target domain lexicon of OTG tokens derived from tagging	bananas, yucky
$\widetilde{D}^t_u$	Templatized target domain sentence (for similarity scoring)	REP are very REP!
D^{weak}	Negative emotion sentence	I hate Sundays – they are so dull
$\widetilde{D}^{weak}$	Negative emotion sentence after tagging and templatizing	I hate REP – they are so REP
	Negative emotion sentence, domain adapted	I hate bananas – they are so yucky

Table 4.1: Example sentences from each stage of the domain adaptation. The hate speech lexicon used to derive token-level labels in the source data is from an external source, whereas the hate lexicon for the target domain is the result of applying the tagger to the unlabeled target domain data. The negative emotion sentences are generic and are not related to either the source or the target domains. They are adapted to the new domain first by selecting the sentences that are most topically similar to the target domain, and then imputing target domain hate speech tokens into the sentences.

CRVs is that they are boring" the *context carrier* is "The problem with ... is that they are ...". We also define *Offensive or Target Group* (OTG) tokens as combination of offensive keywords and keywords indicating a specific race, gender, religion, etc. that are the target of the offense. These are the hate speech *content* of a sentence. We learn an OTG token tagger, T_{OTG}, from the source data D^s, that outputs hate speech content and context carriers from a sentence input.

The data D^s is labeled for sentences rather than tokens, but almost all the hate speech datasets are retrieved from social media or blog search systems with queries from a hate speech lexicon. In this paper we used the lexicon from hatebase.org[1] as the hate speech lexicon, H. Entries in H are unigrams (such as "criminal") and phrases that mention offensive terms and a target group. We tokenize the phrases and consolidate them with the unigrams to create a lexicon of OTG tokens, H^s.

To create training data for T_{OTG}, we select examples from D^s that have been labeled as hate speech at the sentence level, D^s_{hate}. We iterate over the tokens in

[1] `https://hatebase.org/` visited May 2021

D^s_{hate}, and label tokens as "OTG" that have a match in the hate lexicon H^s. Other tokens are labeled as "O". We did not use non-hate examples from D^s for training the model even if OTG tokens appear in that part of the data, because the appearance of OTG tokens in a neutral sentence is not necessarily indicative of offensiveness or hate. For example, a sentence might mention the International Criminal Court, matching the hate term "criminal" and not be in any way offensive or hateful.

Once we label the sequence tagging data set from the source hate speech data set, we learn the sequence tagger, T_{OTG}. We used both character and word level representations in the model. The character-level representation captures terms that have been encoded[2] to avoid automatic detection.

The tagger T encodes character vectors using convolutions, and then max pooling obtains the character-based representation of a word. The word embedding representation is concatenated with it. A Bidirectional Long Short Term Memory (BiLSTM) layer is applied on top of the concatenated representations to obtain a contextual word representation. Finally, a Softmax layer is applied on the word representation to obtain a probability distribution over the label set. An example is shown in Figure 4.1. The tokens "honda", "CRVs" and "boring" are tagged as OTG tokens.

To create a weakly-labeled data set in the target domain, we apply the tagger T to the (unlabeled) target domain dataset, D^t_u. This produces two outputs: a new hate speech lexicon comprised of OTG tokens in the target domain, H^t, and the set of target domain context carriers $\widetilde{D}^t_u$. We replaced the OTG tokens with the token "REP" to templatize the sentences. Note that the context carrier now represents the topic of the sentence, minus the hate terms.

We also apply the tagger to the noisy negative emotion data set D^{weak} to obtain the negative emotion context carriers $\widetilde{D}^{weak}$, which we also templatize with the token

[2]Encoding substitutes numbers and special characters for letters in words to evade lexical pattern matching.

"REP". We discard the tokens tagged as "OTG" in the negative emotion data because they are more likely to be generic nouns and adjectives.

4.1.2 Adaptation of Weakly Labeled Data to the Target Domain

The above process yields a large weakly-labeled corpus of synthetic hate speech $\widetilde{D}^{weak}$ candidate sentences, which are unrelated topically to either the source or target domains. We adapt this corpus to the target domain as follows. We represent the sentences in both $\widetilde{D}^{weak}$ and $\widetilde{D}^{t}_{u}$ as tf-idf vectors. For each sentence in $\widetilde{D}^{weak}$ we compute the cosine similarity to each sentence in $\widetilde{D}^{t}_{u}$. This produces a vector of similarity scores for each sentence in $\widetilde{D}^{weak}$, which we sum to produce a single score which represents the topical similarity of the sentence to the target domain. Note that this similarity is computed in the absence of OTG tokens.

We select the top 10,000 sentences according to the similarity score that contain at least two "REP" tokens. We replace the "REP" tokens with tokens from the target domain hate lexicon H^t, uniformly and at random. We label these sentences as hate speech. Random sampling is a reasonable strategy here because it reduces bias towards any specific OTG term. Note that although the tagger was trained entirely on hate speech sentences, there is no guarantee a whether a specific term in H^t is offensive or target group indicative. This work is focused towards creating robust out-of-domain hate speech detectors without any additional labeled data.

We also select the top 10,000 sentences that contain no more than one "REP" token, and replace all "REP" tokens with tokens randomly sampled from H^t. We label these sentences as non-hate speech to allow the learner to distinguish between hate speech (directed at a target) and speech which is merely offensive. The final data set is comprised of the labeled source dataset D^s, and the domain adapted training sentences, containing both hate and non-hate examples.

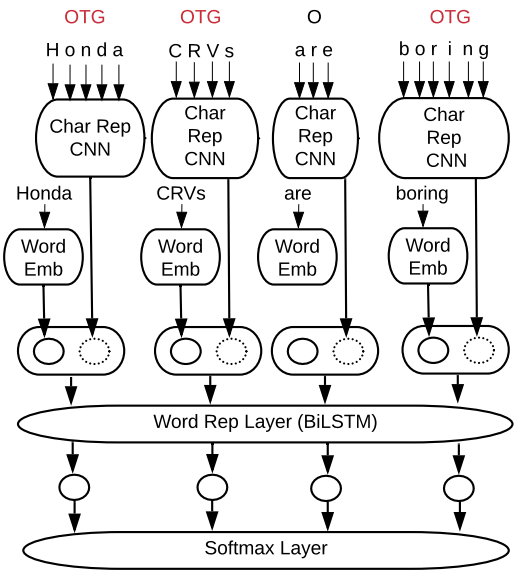


Figure 4.1: The Offensive or Target Group (OTG) tagging model. The model makes use of character-level and word-level information. In this example "Honda" and "CRVs" are the Target, "boring" is offensive, and "are" is neutral. Tokens are labeled "OTG" and "O" accordingly.

Dataset name	Number of examples	Hate Speech	Source of Data
WA (Waseem, 2016)	14949	4839	Tweets
DBW (Davidson et al., 2019)	24783	4993	Tweets
SE (Basile et al., 2019)	9000	3783	Tweets
GI (De Gibert et al., 2018)	10944	1196	Forum posts
HA (Majumder and Patel, 2019)	5852	1143	Facebook posts and tweets
AR (Arango et al., 2019)	7006	2920	Unbiased WA and DBW hate speech

Table 4.2: Description of the hate speech datasets

4.2 Hate Speech Datasets

We consider the datasets provided by Waseem (2016) and Davidson et al. (2019) as source-task data following Arango et al. (2019). We include two more data sets as target-task data sets provided by De Gibert et al. (2018) and Majumder and Patel (2019) along with the only dataset provided by Basile et al. (2019) that Arango et al. (2019) used in their task adaptation experiments. Note that we create a UDA setting from all these data sets, which we describe in Section 4.3.3 and this section only provides a summary of the original data sets. Table 4.2 provides the collection statistics for the data sets.

4.2.1 Source Domain Data

WA: Waseem (2016) collected 136,052 tweets, from two months of Twitter[3] data, focusing on entities likely to engender hate speech. They annotated 16,914 of the tweets. A tweet is annotated as hate speech if it uses a sexist or racial slur, or attacks a group of people on the basis of their religion, gender, ethnicity or sexuality, or if it defends xenophobia or sexism. Their specific approach to collection and annotation ensured that non-hate speech in this corpus contains offensive terms. These offensive examples that are not hate speech present a challenge to hate speech detection because it is difficult for a classifier to distinguish the hateful tweets from those that are merely offensive.

DBW: Davidson et al. (2019) queried twitter using a hate speech lexicon from `hatebase.org` and retrieved 85.4 million tweets written by 33,458 users. From this large collection they randomly selected 25k tweets and crowd-sourced the annotations as one of three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech. They defined hate speech as a language used to express hatred towards a targeted group or intended to be derogatory, to humiliate, or to insult the

[3]www.twitter.com visited May 2021

members of the group. Although the tweets were retrieved using offensive keywords, only 5% of the randomly sampled tweets were coded as hate speech, while a majority of them were identified as offensive.

AR: Arango et al. (2019) de-biased WA and added hate speech tweets from DBW to create a combined dataset that outperformed models trained on the biased WA by a large margin using SE (Basile et al., 2019) as the test set. Because of this improvement over the previous data sets, and its focus on domain bias, AR is our baseline dataset, and the base upon which we augment the data.

4.2.2 Target Domain Data

SE: Basile et al. (2019) released this dataset for the "Multilingual detection of hate speech against immigrants and women in Twitter" task at SemEval. The task organizers defined hate speech as "any communication 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 characteristics." Tweets were collected using multiple strategies including monitoring the accounts of people known to use hate speech, as well as sampling tweets containing terms from a lexicon of offensive keywords. The dataset is multi-lingual (Spanish and English). The English training set consists of 10,000 tweets among which roughly 40% represent hate speech.

GI: De Gibert et al. (2018) sampled sentences published between 2002 and 2017 collected from Stormfront, a white supremacist forum. It contains 10,568 sentences classified into hate speech and non-hate speech. The annotators define hate speech as "a deliberate attack directed towards a specific group of people motivated by aspects of the group's identity."

HA: Majumder and Patel (2019) created a labeled collection of posts from Twitter and Facebook in Indo-European Languages: German, English, and Hindi. The organizers created evaluation benchmarks for three sub-tasks, and we use labeled

Training set	PRAUC	AUC	PR	REC	F1	TP	FP
WA	0.583	0.673	0.654	0.307	0.417	1160.7	616.7
DBW	0.566	0.648	**0.664**	0.166	0.265	627.3	317
Unbiased WA + hate speech from DBW (AR)	0.605	0.674	0.533	**0.684**	**0.598**	**2588.9**	**2283.7**
all WA + hate speech from DBW	**0.645**	**0.716**	0.659	0.49	0.562	1855.75	961.13

Table 4.3: Addition of more examples of hate speech is comparable to unbiasing the data set. PRAUC values reported for WA and AR are slightly different from the ones reported in Table 4.5, because we perform in-domain cross validation in that table.

data for the binary classification task that requires a model to classify a post as hate speech or non-offensive. We use the training dataset for English in our evaluation. After manual inspection we found that sentences from the English dataset are often code-mixed with Hindi, which makes this dataset challenging and different from all other datasets. Table 4.5 indicates that a Word-BiLSTM model struggles to achieve a reasonable PRAUC on this dataset, even when it is trained with labeled instances from the same dataset.

4.3 Experimentation

We consider three different models based on text representation techniques. The first one, *Word-BiLSTM*, is a BiLSTM based model proposed by Agrawal and Awekar (2018b) and used by Arango et al. (2019). The second, *Char-CNN*, is a Convolutional Neural Network (CNN) that applies convolution over character representations. The third model, *Subword-BERT*, is a fine-tuned BERT (Devlin et al., 2019), which uses subwords to convert text to vectors. For all the models, the validation set was 10% of the training set (source domain data + weakly labeled data).

We show that the domain adaptation approach described above improves results across a variety of models and data sets, even when the text is a mixture of languages and uses character-level substitutions. All the results in this paper are produced

by running the same algorithm 10 times with the same hyper-parameters using 10 different random seeds and averaging performance.

4.3.1 Model Details

The focus of this work is on the domain adaptive data generation, not on the models themselves. We show in the experimental results section that a different model performs best in each target domain because of the token representation. Character attacks are very common in hate speech and BERT fine-tuning also fails with character level adversarial attacks. We do not propose or advocate any specific model in this paper, as the focus is the data generation, and it is model-agnostic by design.

Word-BiLSTM follows Agrawal and Awekar (2018b), who proposed a deep learning model for the detection of cyberbullying, which often involves hate speech. They explored CNN, LSTM, BiLSTM, and BiLSTM with attention architectures with the underlying Glove word embedding representation. The results for all architectures were similar. As we compare our results with Arango et al. (2019), we also use the BiLSTM model. The sequence of layers in this architecture is word embedding, then a BiLSTM, then fully connected layers, and finally softmax. The authors used 50-dimensional word vectors and LSTMs (both directions makes it 100 dimensional). We apply Dropout after the BiLSTM and word embedding layers. Even though Arango et al. (2019) trained the BiLSTM model with the Adam optimizer for 10 epochs, we further create a validation set and follow an early stopping strategy with patience value of 3.

Char-CNN is an implementation of the model proposed by Zhang et al. (2015). This model looks at the input text as a sequence of characters. Given the sequence of character embedding, this model applies six layers of convolution with max-pooling. Then it applies three fully connected layers with two dropout modules in between

them for regularization. The early stopping mechanism was used for this CNN with patience value of 3.

Subword-BERT uses the $\text{BERT}_{\text{base}}$ model to encode text Devlin et al. (2019). We apply a special token [CLS] at the beginning of the text and another token [SEP] at the end of the text. We take the representation of the [CLS] token from the 12^{th} layer of BERT, which is a 768-dimensional vector and pass it through a Fully Connected (FC) layer. Finally, we apply a softmax activation function on the representation computed by the FC layer to classify. We used a batch size of 32, with a learning rate of 2e-5, and trained the model for three epochs. Devlin et al. (2019) mentioned that 2-4 epochs of fine-tuning is quite effective for the GLUE (Wang et al., 2018) tasks. We found that training for 3 epochs works best in our setting.

4.3.2 Preliminary Experiments

The selected datasets provide a platform for creating a challenging domain adaptation setting. We demonstrate this by showing the drop in PRAUC (Area Under the Precision-Recall Curve), when the training and test set are from different datasets compared to when they are from the same dataset, as shown in Table 4.5. Note that the diagonal represents testing on a held out set of 10% of the data, and training on the other 90%. We used the word-BiLSTM model described in section 4.3.1 for these experiments.

We replicate the results of Arango et al. (2019), and further add hate speech examples from DBW without limiting the number of tweets from a single user. We run the word-BiLSTM model using the hyper-parameters from Arango et al. (2019). We report PRAUC and AUC, True Positives and False Positives, alongside precision, recall, and F1 scores reported by Arango et al. (2019). The result is shown in Table 4.3. The first two rows show that using WA and DBW alone results in poor performance when adapting to SE. The third and fourth rows show that limiting tweets from users

is less effective if we consider PRAUC and AUC, as shown by comparing WA to the unbiased version of WA when adding hate speech examples from DBW to both sets.

4.3.3 Unsupervised Domain Adaptation Setting

Unsupervised domain adaptation assumes that only unlabeled data exists in the target domain. To create such a setting, we randomly sample 10% data from each of the target datasets to create unlabeled data. This resulted in 900, 1095, and 586 randomly selected sentences from SE, GI, and GA datasets, respectively. We do not use the labels of these sentences but use the sentences themselves in the noisy data generation process. The remaining data is used as test set. In the SE, GI, and HA test sets there are 3409, 1097, and 1040 hate speech examples, and 4691, 8752, and 4226 non-hate speech examples, respectively. The data is not truly a uniform random sample from the unlabeled data of the target corpus as it is a part of the original labeled data. However this is a typical limitation of UDA settings.

To show the effectiveness of our proposed approach, we use AR as the baseline training data, and show the improvements that we obtain by augmenting domain adaptive weakly supervised data with AR. Our technique involves training an Offensive or Target Group (OTG) tagger from AR, and we adapt the sequence tagger implementation of Yang and Zhang (2018) for this task.

Note that AR consists of unbiased WA and DBW. While the DBW dataset is sampled using a hate speech lexicon taken from `hatebase.org`, WA was not sampled in that way. Following our approach described in Section 4.1.1, we require to match tokens from a hate speech lexicon to hate speech data for generating training data for the OTG tagger. We only use the DBW portion of the AR dataset for this purpose. We use an n-gram based matching technique to map the tokens from the hate speech lexicon to the 4993 hate speech in DBW. Once we train the OTG tagger with this data, we run the tagger on a large scale *weakly supervised* sentiment analysis

dataset provided by Go et al. (2009). This dataset contains 800,000 negative emotion sentences that we convert to hate speech templates using the OTG tagger, as described in Section 4.1.1.

Following the approach described in section 4.1, we rank these templates by their similarity to the target domain, select top 10,000 hate and non-hate templates, and convert them to hate and non-hate examples. The value of 10,000 was determined empirically, by tuning it as a parameter on a held out set.

Experiments described in the previous section indicated that data augmentation from the hate speech class is one of the key factors in reducing bias and adapting to a new domain. Results of the experiments in table 4.4 show the effectiveness of adding domain adapted, weakly labeled data to the AR data, evaluated on the SE, GI, and HA test sets, respectively.

Table 4.4 shows that the addition of weakly labeled data improves PRAUC, AUC and F1 metrics for all types of models for the hate speech class. The per-class metrics can be inferred from the True Positives and False Positives and the total number of examples in the data set. In particular recall has a larger gain for character and subword models compared to the word-based model. This is especially notable for the HA data which includes examples that are a code-mix of Hindi and English.

Another important observation is that although BERT fine-tuning is a strong baseline for text classification tasks, it performs worse than the word embedding BiLSTM model on the SE data. This does not hold for the GI data, where we find that BERT fine-tuning supercedes all the other approaches by a large margin. This could be accounted for by the fact that the GI data is sampled from white-supremacists' forum posts which includes complete grammatical sentences, whereas the SE data is from Twitter. As BERT has been trained on Wikipedia, it models this type of content better.

Target Domain	Model	Training Data	PRAUC	AUC	PR	REC	F1	TP	FP
SE	Char-CNN	AR	0.549	0.591	0.460	0.590	0.517	2012	**2358**
		AR + SE_{weak}	**0.558**	**0.646**	**0.496**	**0.748**	**0.597**	**2549**	2585
	Word-BiLSTM	AR	0.605	0.674	0.533	**0.684**	0.598	**2588.9**	2283.7
		AR + SE_{weak}	**0.653**	**0.729**	**0.611**	0.652	**0.631**	2222	**1415**
	Subword-BERT	AR	0.599	0.675	**0.551**	0.637	0.591	2170	**1765**
		AR + SE_{weak}	**0.613**	**0.697**	0.541	**0.740**	**0.625**	**2521.5**	2140
GI	Char-CNN	AR	**0.174**	**0.628**	0.153	0.478	0.232	524	2905
		AR + GI_{weak}	0.167	0.613	**0.166**	**0.500**	**0.249**	**548**	**2750**
	Word-BiLSTM	AR	0.151	0.514	0.151	0.297	0.200	326	1832
		AR + GI_{weak}	**0.225**	**0.660**	**0.213**	**0.442**	**0.288**	**485**	**1787**
	Subword-BERT	AR	0.291	0.758	0.234	0.644	0.343	706	2309
		AR + GI_{weak}	**0.331**	**0.786**	**0.260**	**0.644**	**0.369**	**706.5**	**2019.5**
HA	Char-CNN	AR	0.216	**0.519**	0.203	0.225	0.213	234	**921**
		AR + HA_{weak}	**0.307**	0.514	**0.203**	**0.845**	**0.327**	**879**	3461
	Word-BiLSTM	AR	0.205	0.510	0.203	0.474	0.283	541.4	**2130.3**
		AR + HA_{weak}	**0.217**	**0.533**	**0.209**	**0.555**	**0.304**	**577**	2183
	Subword-BERT	AR	**0.209**	0.525	**0.218**	0.254	0.234	264	**948**
		AR + HA_{weak}	0.208	**0.526**	0.205	**0.851**	**0.331**	**885**	3434.5

Table 4.4: The UDA approach improves over training with source domain dataset, AR, taken from Arango et al. (2019). AR is a combination of unbiased WA and hate speech from DBW. SE_{weak}, GI_{weak} and HA_{weak} indicate the domain-adapted weakly labeled data as described in Section 4.1. The results are average of 10 runs and the best results are boldfaced.

		Testing Set				
		WA	**DBW**	**SE**	**HA**	**GI**
Training Set	**WA**	**0.768**	0.199	0.561	0.198	0.103
	DBW	0.390	**0.465**	0.525	0.191	0.079
	SE	0.390	0.226	**0.725**	0.195	0.133
	HA	0.396	0.213	0.421	**0.240**	0.062
	GI	0.384	0.275	0.455	0.172	**0.404**

Table 4.5: Cross-dataset performance represented using PRAUC. The same 90/10 train/test split was used in each comparison. In most cases, the results are significantly worse on out-of-domain test data.

4.3.4 Model Adaptation vs. Data Augmentation

The model improvements presented in this paper are data-driven, as we increase the model effectiveness by augmenting weakly labeled data with source domain data in the training process. Model-driven approaches, such as ACAN (Qu et al., 2019) take advantage of the unlabeled target domain data in the training process for learning domain-agnostic representations, but they do not use any external data. As ACAN is a strong baseline for UDA for sentiment analysis, we investigate its performance for hate speech detection. ACAN uses Glove word embeddings as the underlying representation, and thus it is comparable to the Word-BiLSTM model used in this paper. Note that the Word-BiLSTM is not trained with any domain alignment objective, but it receives the weakly labeled data as input along with the source domain data.

Target Domain	Approach	PRAUC	AUC	P	R	F1
SE	ACAN	0.619	0.699	0.469	**0.936**	0.625
	Proposed	**0.653**	**0.729**	**0.541**	0.740	0.625
GI	ACAN	0.185	0.651	0.127	**0.933**	0.224
	Proposed	**0.225**	**0.660**	**0.213**	0.442	**0.288**
HA	ACAN	**0.220**	**0.548**	0.206	**0.905**	**0.336**
	Proposed	0.217	0.533	**0.209**	0.555	0.304

Table 4.6: Comparison of the proposed approach with model-driven domain adaptation approach, ACAN (Qu et al., 2019)

Table 4.6 shows the performance comparison of our approach and ACAN. For the SE and GI datasets, our proposed approach performs better than ACAN across a variety of evaluation metrics, primarily driven by higher precision. However, ACAN performs better on the HA dataset. The HA data set is the most dissimilar to the source data, as it includes a code-mixed Hindi and English examples, where Hindi words are transliterated using the English alphabet. The better performance of model-driven adaptation suggests that model-based approaches may be suitable when the source and target domains are very different. We only use ACAN as a

reference point as to the best of our knowledge, there is no work on UDA for hate speech detection. It is possible that using both in combination would improve the results further.

4.4 Discussion and Summary

The main challenge in hate speech detection is not the bias, but the data imbalance that arises from having a limited set of examples of hate speech because hate speech is generated by few users. Even if a large number of examples are sampled from a source such as Twitter, a domain gap exists because of the many linguistic variants, targets of the hate speech, and topics that are vectors of hate. We created a domain-specific hate speech data generator by turning a large collection of weakly supervised negative sentiment sentences into domain adapted hate speech. We demonstrated that the approach improves results over training on data from a different domain, even when bias has been reduced in the original data.

Although WA was shown to be biased by Arango et al. (2019), training with only DBW yields worse performance compared to WA. We didn't experiment with this further by checking if bias exists in the hate speech examples from DBW as well, as it is not our research direction, but Table 4.2 reflects that DBW has a greater class imbalance compared to WA. Over-sampling the hate speech class in both cases did not resolve the problem.

Training with WA augmented with hate speech examples from DBW results in fewer true positives, compared to training with the unbiased WA data. This suggests that the high precision and low recall is the result of over-fitting to the hate speech of a few users. The overall performance is still close to the unbiased WA dataset, indicating that adding more data from the hate speech class reduces the bias.

The F1 value in the hate speech class reported by Arango et al. (2019) trained on the WA data is low compared to our implementation of the same model, indicated in

Table 4.3. We looked at the source code obtained from the authors and found that our implementation differed in three ways: we created a validation set, implemented an early stopping strategy, and did not consider the test data vocabulary while constructing the word embedding table. However, we observed a little change in F1 in the hate speech class when training with unbiased WA. Even though we obtained different results, the gain in terms of F1 with unbiasing is still evident.

A limitation of the data generation approach is that it captures sentences that follow a specific template, requiring two slots for imputing offensive content, rather than just one. The assumption is that to be hate speech (rather than just offensive content) there must be an offensive descriptor, directed at a subject in the sentence. In real life, there are myriad ways to express hate, which may not be reflected in this particular template. The template approach will be most effective when the negative sentiment sentences are topically related to the domain of hate speech. It will do poorly when the hate speech contains implicit mentions of target groups, or implicit hate.

The template generation process is noisy. For example, a one-slot negative example (not hate speech) from the actual data is "I wish i got to ... it with you i miss you and how was the premiere". A positive example (hate speech, with two slots) is "fml so ... for seniority bc of technological ineptness i now have to register for ...". This does not matter for the purpose of hate speech detection, because the only purpose of the domain-adapted data is to capture topically similar negative sentiment context, which can be made domain-specific with hate tokens. Further, we select the most topically related context sentences and discard the rest.

Deep learning is especially suited to hate speech detection because there are very few features that can be crafted that are not dependent on a specific hateful vocabulary, whereas hate speech itself is often considerably more subtle, using no specifically hateful term. Still, there may be benefit to adding features of the community or social

network structure, on the basis that people engaged in hate speech form a community and often coordinate to conduct a campaign of hate.

CHAPTER 5

ABUSIVE LANGUAGE DETECTION WITH LIMITED TARGET LANGUAGE DATA

In this chapter, we propose a novel cross-lingual transfer learning approach for abusive language detection. Even though we evaluate the effectiveness of this approach in detecting abusive language, this approach can be used to flag content that is unacceptable in an online platform given the training instances of acceptable and unacceptable content.

Online abusive language is a superset of hate speech and it might range from hate speech and cyberbullying with extreme deleterious effects, over slightly less damaging derogatory language and personal insults, to profanity and teasing, which might even be considered acceptable in some communities and on some social platforms (Nakov et al., 2021). In order to flag abusive content in a language where labeled data is scarce – e.g., in the scale of hundreds – we augment an English content flagging dataset to improve prediction in that language. Similar to the previous chapters, our target-task is data-scarce. But, in contrast to the QBE and zero-shot setting we explore the limited-data setting. This is because a limited-data setting is more practical in the context of abusive language detection in online platforms because a platform generally owns hundreds of training instances for a language. Our goal is to help such an online platform to transfer knowledge from a large-scale English content flagging dataset to their task using a model.

To battle data scarcity we do not devise any augmentation technique in this chapter. To be precise, we neither modify the English labeled dataset nor use it

to generate any new instances. Rather, we propose a novel approach to transfer knowledge from the source English dataset to improve performance on the target language-specific dataset. Thus this chapter contributes to the transfer phase of our Augment-Transfer framework.

Our cross-lingual transfer learning technique is based on the assumption of an evolving English content flagging dataset. This means that once our training is completed with task-specific limited data and English labeled data, the model can still take advantage of the English dataset if it is being continually updated. This is possible because our framework is based on the classical nearest-neighbor framework, and in this chapter we consider retrieving neighbors from an English content-flagging dataset. Unlike a classical kNN framework, our neighborhood framework uses a sentence embedding model such as LaBSE (Feng et al., 2020) for neighbor retrieval, and it fine-tunes a pre-trained language model to compute the representations of the neighbors. Our framework aggregates the computed representations to capture neighborhood information to make a decision about flagging a textual content. We refer to our neighborhood framework as $k\text{NN}^+$.

In a classical kNN setup, decision based on the the retrieved neighborhood is captured using majority voting, whereas our $k\text{NN}^+$ model learns the voting strategy. Once our $k\text{NN}^+$ model is learned with a snapshot of target-task data and English data, we can keep augmenting English data. This means that we can separate out the augmentation and inference part. $k\text{NN}^+$ can seamlessly integrate *augment* and *transfer* under a single framework, which is eventually expected from the Augment-Transfer framework. The augmentation phase will grow the neighborhood database and the classification of a textual content will be performed by retrieval from the neighbor database. Even though both in this chapter and the previous chapter we discuss abusive language and hate speech, this framework has the potential to be applied to any text classification task.

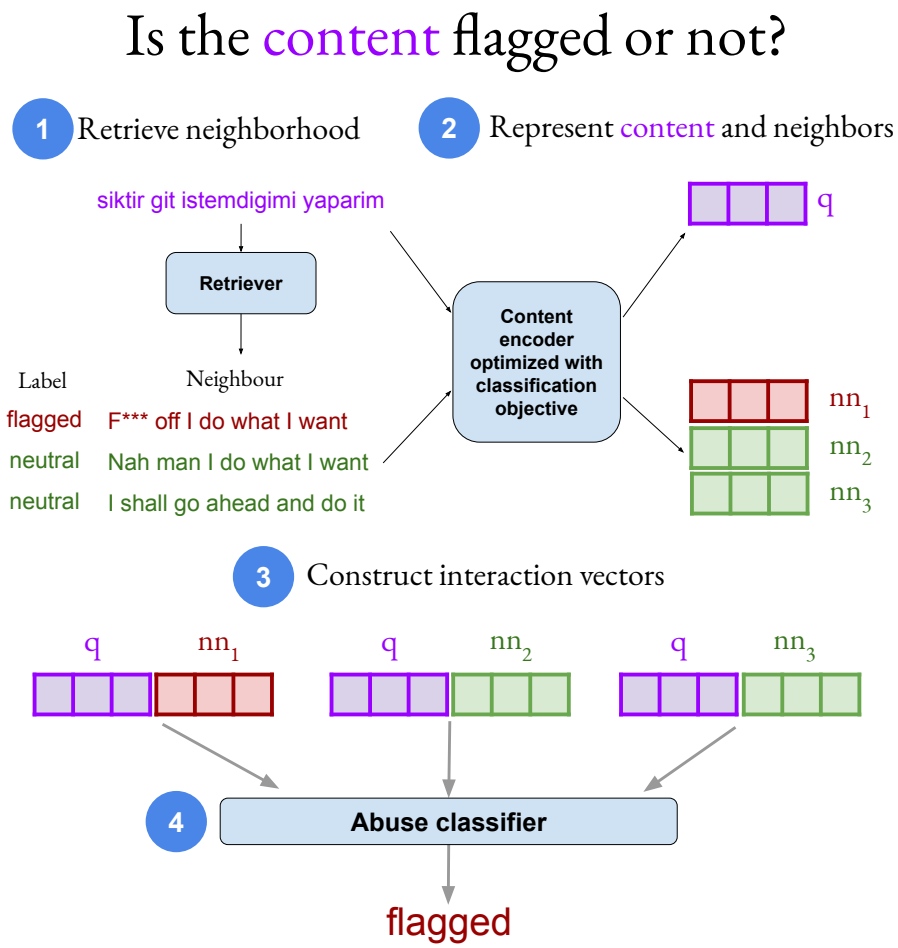


Figure 5.1: Conceptual diagram of our neighborhood framework. The query is processed using run-time compute, while the neighbor vector is pre-computed.

We perform extensive evaluation of our framework using a large labeled English dataset (Jigsaw, 2018) as the neighbor repository and small language-specific datasets (Jigsaw Multilingual, 2020) to train our neighborhood model. Our evaluation results on eight languages from two different datasets for abusive language detection show sizable improvements of up to 9.5 F1 points absolute (for Italian) over strong baselines. On average, we achieve 3.6 absolute F1 points of improvement for the three languages in the Jigsaw Multilingual dataset and 2.14 points for the WUL dataset. The work described in this chapter is drawn from an accepted publication in the Transactions of the Association for Computational Linguistics (Sarwar et al., 2021).

5.1 Problem Setting

Our goal is to learn a content flagging model from source and target datasets in different languages with different label spaces – see Figure 5.1 for an illustration of our framework. In this framework, given a Turkish content for classification, we consider it as a query and retrieve a neighborhood of English contents along with their labels. Then we compute the interaction vector of the representation of the query and each of the neighbors. Finally, we aggregate the interaction vectors to reach a decision about the query. The process of reaching a decision from the interaction vectors is learned.

Formally, we assume access to a source dataset for content flagging, $D^s = \left\{(x_i^s, \mathbf{y}_i^s)\right\}_{i=1}^{n_s}$, where x_i^s is a textual content and $\mathbf{y}_i^s \in \mathcal{Y}$. Further, a target dataset is given, $D^t = \left\{(x_j^t, y_j^t)\right\}_{j=1}^{n_t}$, where $y_j^t \in \{flagged, neutral\}$. D^s is resource-rich, i.e., $n_s \gg n_t$, and label-rich, i.e., $|\mathcal{Y}| > 2$. The label space, $\mathcal{Y} = \{hate, insult, \ldots, neutral\}$, of D^s contains fine-grained labels for different levels of abusiveness along with the *neutral* label. We convert the label space of D^s as, $\mathcal{Y}' = \{flagged \mid x \in \mathcal{Y}, x \neq neutral\}$, to align with the label space of D^t.

5.2 Why a neighborhood Framework?

A vanilla kNN predicts a content label by aggregating the labels of k similar training instances. To this end, it uses the content as a query to retrieve neighbors from the training instances. We hypothesize that this retrieval step can be performed in a cross-lingual transfer learning scenario. In our setting, the queries are target dataset instances, and we index the source dataset for retrieval. Note that the target instances could also be considered as neighbors for retrieval, but we exclude them, as the target dataset is small.

For a vanilla kNN model, the queries and the documents are represented using lexical features, and thus the model suffers from the curse of dimensionality (Radovanović

et al., 2009). Moreover, the prediction pipeline becomes inefficient if the source dataset is considerably larger than the target dataset, as is our case here (Lu et al., 2012). Finally, for a vanilla kNN, there is no straight-forward way to map between different languages for cross-lingual transfer.

We address these problems by using a Transformer-based multilingual representation space (Feng et al., 2020) that computes the similarity between two sentences expressed in different languages. We assume that efficiency issues are less critical here for two main reasons: (*i*) retrieval using dense vector sentence embeddings has become significantly faster with recent advances (Johnson et al., 2019), and (*ii*) the number of labeled source data examples is not expected to go beyond millions, because obtaining annotations for multilingual abusive content detection is costly and the annotation process can be very harmful for the human annotators as well (Schmidt and Wiegand, 2017; Waseem, 2016; Malmasi and Zampieri, 2018; Mathur et al., 2018).

Even though multilingual language models can make the vanilla kNN model a viable solution for our problem, it is hard to make predictions with that model. Once a neighborhood is retrieved, a vanilla kNN uses a majority voting scheme for prediction, as the example in Figure 1.1 shows. Given a flagged Turkish query, our framework retrieves two *neutral* and one *flagged* English neighbors. Here, the majority voting prediction based on the neighborhood is incorrect. The problem is this: *A non-parametric vanilla kNN cannot make a correct prediction with an incorrectly retrieved neighborhood*. Thus, we propose a learned voting strategy to alleviate this problem.

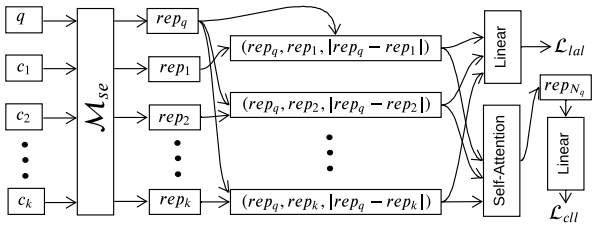
5.3 Architecture of kNN$^+$

We describe our kNN$^+$ framework (shown in Figure 5.2), including the training and the inference procedures. The framework includes neighborhood retrieval, interaction feature computation and aggregation, and a multi-task learning objective function for optimization, which we describe in detail below.

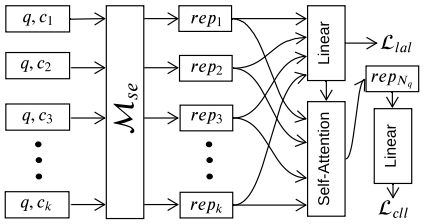
5.3.0.0.1 neighborhood Retrieval We construct a retrieval index R from the given source dataset, $D^s = \left\{(x_i^s, \mathbf{y}_i^s)\right\}_{i=1}^{n_s}$. For each given example $x_i^s \in D^s$, we compute its dense vector representation, $\mathbf{x}_i^s = \mathcal{M}_{retriever}(x_i^s)$. Here, $\mathcal{M}_{retriever}$ is a multilingual sentence embedding model that we use for retrieval. There are several multilingual sentence embedding models that we could use as $\mathcal{M}_{retriever}$ (Artetxe and Schwenk, 2019; Reimers and Gurevych, 2020; Chidambaram et al., 2019; Feng et al., 2020). In this work, we use LaBSE (Feng et al., 2020), a strong multilingual sentence matching model, which has been trained with parallel sentence pairs from 109 languages. The model is trained on 17 billion monolingual sentences and 6 billion bilingual sentence pairs and it has achieved state-of-the-art performance for a parallel text retrieval task proposed by Zweigenbaum et al. (2017). We use $\mathbf{x}_i^s$ as a key, and we assign $(x_i^s, \mathbf{y_i}^s)$ as its corresponding value. Our retrieval index R stores all the key-value pairs computed from the source dataset.

Assume we have a training data point, $(x_j^t, y_j^t) \in D^t$, from the target dataset. We consider the content x_j^t as our query q, i.e., $q = x_j^t$. We compute a vector representation of the query, $\mathbf{q} = \mathcal{M}_{retriever}(q)$. We use $\mathbf{q}$ to score each key, $\mathbf{x}_i^s$ of R using cosine similarity, i.e., $cos(\mathbf{q}, \mathbf{x}_i^s)$.

We sort the items in R in descending order of the scores of the keys, and we take the values of the top-k items to construct $N_q = \{(c_1, l_1), (c_2, l_2), \ldots, (c_k, l_k)\}$, the neighborhood of q. Thus, each neighbor is a tuple of a content and its label from the source dataset. We convert fine-grained neighbor labels to binary labels (*flagged, neutral*) as described in Section 5.1, to align the label space with the target dataset. Nevertheless, the original fine-grained labels of the neighbors can be used to get an explanation at inference time as this is one of the core features of kNN-based models. However, our focus is on combining these models with Transformer-based ones. We leave the investigation of the explainability characteristics of $k\text{NN}^+$ for future work.



(a) Bi-Encoder kNN (BE kNN$^+$) Variant.



(b) Cross-Encoder kNN (CE kNN$^+$) Variant.

Figure 5.2: Two variants based on two encoding schemes used in our proposed kNN$^+$

5.3.1 kNN$^+$ Framework

5.3.1.1 Interaction Feature Modeling

As discussed in Section 5.2, the neighborhood retrieval process might lead to prediction errors. Thus, we propose a learned voting strategy to mitigate this. Our proposed strategy depends on how q relates to its neighborhood N_q. To model this relationship, we compute the interaction features between q and the content of its j-th neighbor, $c_j \in N_q$. We obtain a set of k interaction features from k neighbors, and we optimize them using query and neighbor labels.

Similarly to Reimers and Gurevych (2019b), we apply two encoding schemes to compute the interaction features: a **Cross-Encoder (CE)** and **Bi-Encoder (BE)**. Under our kNN$^+$ framework, we refer to the schemes as CE kNN$^+$ for CE, and BE kNN$^+$ for BE. The BE kNN$^+$ is computationally inexpensive, while the CE kNN$^+$ is more effective. We provide a justification for this as we describe the schemes in the following paragraphs.

For the CE kNN$^+$ implementation (see Figure 5.2b), we first form a set of query–neighbor pairs $S_{ce} = \{(q, c_1), (q, c_2), \ldots, (q, c_k)\}$ by concatenating q with the content of each of its neighbors. Then, we obtain the output representation, $rep_j = \mathcal{M}_{feature}(q, c_j)$ of each $(q, c_j) \in S_{ce}$, from a pre-trained multilingual language model $\mathcal{M}_{feature}$. In this way, we create a set of interaction features, $I_{ce} = \{rep_1, rep_2, \ldots, rep_j\}$

from q and its neighborhood. Throughout this paper, the [CLS] token representation of $\mathcal{M}_{feature}$ is taken as its final output. We use varieties of implementations of $\mathcal{M}_{feature}$ in the experimentation. Figure 5.2b shows how the interaction features are computed and optimized with a CE kNN$^+$.

Note that the feature interaction model $\mathcal{M}_{feature}$ is different from the neighborhood retrieval one $\mathcal{M}_{retriever}$. We optimize interaction features from $\mathcal{M}_{feature}$, and we leave retrieval model optimization for future work.

For the BE kNN$^+$ scheme (see Figure 5.2a), we obtain the output representations of q and each of the neighbors individually from $\mathcal{M}_{feature}$. Given the representation of the query, $rep_q = \mathcal{M}_{feature}(q)$, and the representation of its j^{th} neighbor, $rep_j = \mathcal{M}_{feature}(c_j)$, we model their interaction features by concatenating them along with their vector difference. The interaction features obtained for the j-th neighbor are $(rep_q, rep_j, |rep_q - rep_j|)$, and we construct a set of interaction features I_{be} from all the neighbors of q. We use the vector difference $|rep_q - rep_j|$ along with the content vectors rep_q and rep_j following the work of Reimers and Gurevych (2019b). They trained a sentence embedding model using a Siamese neural network architecture with Natural Language Inference (NLI) data. They tried the following approaches to obtain features between the representations u and v of two sentences: $(u, v), (|u - v|), (u*v), (|u-v|, u*v), (u, v, u*v), (u, v, |u-v|), (u, v, |u-v|), (u*v)$. Their empirical analysis showed that $(u, v, |u - v|)$ works the best for NLI data, and thus we apply this in our framework. We plan to explore other options in future work.

Both the cross-encoder and the bi-encoder architectures were shown to be effective in a wide variety of tasks including Semantic Textual Similarity and Natural Language Inference. Reimers and Gurevych (2019b) showed that a bi-encoder is much more efficient than a cross-encoder, and that bi-encoder representations can be stored as sentence vectors. Thus, once $\mathcal{M}_{feature}$ is trained, the vector representations $\mathcal{M}_{feature}(x_i^s)$ of each $x_i^s \in D^s$ can be saved along with the textual contents and label.

Then, at inference time, only the representation of the query needs to be computed, which reduces the computation time from $k \times \mathcal{M}_{feature}$ to a constant time. Moreover, the model can easily adapt to new neighbors without the need for retraining. However, from an effectiveness perspective, the cross-encoder is usually a better option as it encodes the query and its neighbor jointly, thus enabling multi-head attention-based interactions among the tokens of the query and of the neighbor.

5.3.1.1.1 Choice of $\mathcal{M}_{feature}$ We explore two $\mathcal{M}_{feature}$ models for both the CE and the BE schemes: a pre-trained XLM-R model, which we will refer to as $\mathcal{M}_{feature}^{\text{XLM-R}}$, as well as an XLM-R model augmented with *paraphrase* knowledge, which we will refer to as $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ (Reimers and Gurevych, 2020). Sentence representations from XLM-R are not aligned across languages (Ethayarajh, 2019) and $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ overcomes this problem. In particular, $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ is trained to learn sentence semantics with parallel data from 50 languages. Moreover, the training process includes knowledge distillation from a Sentence BERT model Reimers and Gurevych (2019b) trained on 50 million English paraphrases. As such, we expect $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ to outperform $\mathcal{M}_{feature}^{\text{XLM-R}}$, as it more accurately captures the semantics of the query and its neighbor sentences. Note that there is work on producing better alignments of multilingual vector spaces Zhao et al. (2020), which would allow us to consider a variety of pre-trained sentence representation models, but exploring this is outside the scope of this paper.

5.3.1.1.2 Interaction Features optimization Given a query q and its j-th neighbor, we obtain features $rep_j \in I_{ce}$ and $(rep_q, rep_j, |rep_q - rep_j|) \in I_{be}$ from $\mathcal{M}_{feature}$ for the CE $k\text{NN}^+$ and BE $k\text{NN}^+$ schemes, respectively. For both schemes, we optimize the interaction features to indicate whether a query and its neighbor have the same or different labels. We do this to later aggregate interaction features from all the neighbors of a query to model the overall agreement of the query with the

retrieved neighborhood. Our hypothesis is that understanding individual neighbor-level agreement and aggregating it will allow us also to understand the neighborhood.

We apply a fully connected layer with two outputs over the interaction features to optimize them. The outputs indicate the label agreement between q and its j-th neighbor, $(c_j, l_j) \in N_q$. There is a label agreement if both q and the j-th neighbor are flagged or are both neutral, i.e., $y_j^t = l_j$. We learn the label agreement using a binary cross-entropy loss $\mathcal{L}_{lal}$, which is computed using the output of a softmax layer for each example in a batch of training data. We refer to $\mathcal{L}_{lal}$ as label-agreement loss. In our implementation, a batch of data comprises a query and its k neighbors. We provide more details about the training procedure in Section 5.4.4.

Note that as our model predicts label agreement, it also indirectly predicts the label of the query and of the neighbor. In this way, it learns representations that separate flagged from the non-flagged examples.

5.3.1.1.3 Interaction Features Aggregation The main reasons to use interaction features for label agreement is to predict whether q should be flagged or not. In a vanilla kNN setup, there is no mechanism to back-propagate classification errors, as the only parameter to tune there is the hyper-parameter k. In our model, we propose to optimize the interaction features – using a self-attention module – to minimise the classification error with a fixed neighborhood size k. To this end, we propose to aggregate the k interaction features: I_{ce} for CE kNN$^+$ and I_{be} for BE kNN$^+$. The aggregated representation captures global information, i.e., the agreement between the query and its neighborhood, whereas the interaction features capture them locally.

We use structured self-attention (Lin et al., 2017) to capture the neighborhood information. At first, we construct an interaction features matrix, $H \in \mathbb{R}^{k \times h}$ from the set of k neighbors (I_{ce} or I_{be}), where h is the dimensionality of the interaction feature space. Then, we compute structured self-attention as follows:

$$\vec{a} = \text{softmax}\left(W_2 \tanh\left(W_1 \mathbf{H}^T\right)\right) \tag{5.1}$$

$$\mathbf{rep_i} = \vec{a}\mathbf{H} \tag{5.2}$$

Here, $W_1 \in \mathbb{R}^{h_r \times h}$ is a matrix that encodes interactions between the representations and projects the interaction features into a lower-dimensional space, $h_r < h$, thus making the representation matrix $h_r \times k$ dimensional. We multiply another matrix $W_2 \in \mathbb{R}^{1 \times h_r}$ by the resulting representation, and we apply softmax to obtain a probability distribution over the k neighbors. Then, we use this probability distribution to produce an attention vector that linearly combines the interaction features to generate the neighborhood representation rep_{N_q}, which we eventually use for classification.

5.3.1.1.4 Classification Loss optimization The aggregated interaction features, rep_{N_q}, are used as an input to a softmax layer with two outputs (*flagged* or *neutral*), which we optimize using a binary cross-entropy loss, $\mathcal{L}_{cll}$. We refer to $\mathcal{L}_{cll}$ as classification loss.

optimizing this loss means that the classification decision for a query is made by computing its agreement or disagreement with the neighborhood as a whole. Our approach is a multi-task learning one, and the final loss is computed as follows:

$$\mathcal{L} = (1 - \lambda) \times \mathcal{L}_{lal} + \lambda \times \mathcal{L}_{cll} \tag{5.3}$$

As both the classification and the label-agreement tasks aid each other, we adopt a multi-task learning approach. We balance the two losses using the hyper-parameter λ. The classification loss forces the model to predict a label for the query. As the model learns to predict a label for a query, it becomes easier for it to reduce the label agreement loss $\mathcal{L}_{lal}$. Moreover, as the model learns to predict label agreement,

it learns to compute interaction features, which represent agreement or disagreement. This, in turn, helps to optimize $\mathcal{L}_{cll}$.

Note that, at inference time, our framework requires neither the labels of the neighbors for classification, nor a heuristic-based label-aggregation scheme. The classification layer makes a prediction based on the pooled representation from the interaction features, thus removing the need for any heuristic-based voting strategy based on the labels of the neighbors. Each individual interaction feature from the query and a neighbor captures the agreement between them as we optimize the features via the L_{lal} loss. The opinion of the neighborhood is captured using an aggregation of individual interaction features – which is different from a vanilla kNN – where neighborhood opinion is captured using an individual neighbor label. As our aggregation is performed using a self-attention mechanism, we obtain a probability distribution over the interaction features that we can use to find the neighbor that influenced the neighborhood opinion the most. We also know both the original and the converted label of the neighbor (see Section 5.1 for further details about the label space conversion). The original label of the neighbor could help us understand the prediction behind the query better. For example, if the query is flagged and the original label of the most influential neighbor is *hate*, we could infer that the query is hate speech. However, we do not explore this direction in this paper, and we leave it as a future work.

5.4 Experimental Setting

5.4.1 Datasets

We conduct experiments on two different multilingual datasets covering eight languages from six families: Slavic, Turkic, Romance, Germanic, Albanian, and Finno-Ugric. We use these datasets as our target datasets, and an English dataset as the source dataset, which contains a large number of training examples with fine-

grained categorization. Both the source and target datasets are from the same domain (Wikipedia), as we do not study domain adaptation techniques in this work. We describe these three datasets in the following paragraphs. The number of examples per dataset and the corresponding label distributions are shown in Table 5.1.

5.4.1.1 Jigsaw English

This is an English dataset, containing over 159 thousand manually reviewed comments (Jigsaw, 2018). The labels (*toxic*, *severe toxic*, *obscene*, *threat*, *insult*, and *identity hate*) are mapped into binary ones: *flagged* and *neutral*. If at least one of those six labels is present, we consider it as *flagged*, otherwise as *neutral*.

As it is a resource-rich dataset, covering different aspects of abusive language, we use it as the source dataset. We use all its examples for training, as we validate our models on *target* datasets' dev sets.

5.4.1.2 Jigsaw Multilingual

Jigsaw Multilingual (2020) aims to improve toxicity detection by addressing the shortcomings of the monolingual setup. The dataset contains examples in Italian, Turkish, and Spanish. It has binary labels (toxic or not), and thus it aligns well with our experimental setup. The label distribution is fairly similar to Jigsaw English, as shown in Table 5.1. It is used for experimenting in a resource-rich environment. As this dataset does not have standard train, test and dev sets, we split the examples in each language as follows: 1,500, 500, and 500 for Italian and Spanish, and 1,800, 600, and 600 for Turkish.

5.4.1.3 WUL

Glavaš et al. (2020) aims to create a fair evaluation setup for abusive language detection in multiple languages. Although originally in English, multilinguality is achieved by translating the original comments as accurately as possible into five dif-

Dataset	Examples	Flagged %	Neutral %
Jigsaw En	159,571	10.2	89.8
Jigsaw Multi	8,000	15.0	85.0
WUL	600	50.3	49.7

Table 5.1: Dataset sizes and label distributions.

ferent languages: German (DE), Hungarian (HR), Albanian (SQ), Turkish (TR), and Russian (RU). We use this dataset partially, by using the test set originally generated from Wulczyn et al. (2017), which focuses on identifying personal attacks.

In contrast to Jigsaw Multilingual, it is used for experimenting in a low-resource environment. For each language, we have 600 examples, which are split as 400, 100, and 100 for train, test, and dev, respectively.

5.4.2 Baselines

We compare our proposed approach against three families of strong baselines. The first one considers training models only on the target dataset, the second one is source adaptation, where we use Jigsaw English as our source dataset, and the third one consists of traditional kNN classification method, but with dense vector retrieval using LaBSE.

5.4.2.1 Target Dataset Training

This family of baselines uses only the target dataset for training:

fastText is a baseline that uses the mean of the token vectors obtained from fastText (Joulin et al., 2017) word embeddings to represent a textual example. Then, a binary logistic regression classifier is trained for content flagging.

XLM-R Target is a pre-trained XLM-R model fine-tuned on the target dataset.

#	Method	Jigsaw Multilingual			WUL					
		ES	IT	TR	DE	EN	HR	RU	SQ	TR
1	FastText	55.3	47.2	64.2	74.2	72.7	58.9	74.2	65.9	72.5
2	XLM-R Target	63.5	56.4	80.6	82.1	75.7	73.2	76.7	77.3	78.8
3	XLM-R Mix-Adapt	**64.2**	58.5	76.1	83.2	**93.9**	87.3	82.1	86.2	86.0
4	XLM-R Seq-Adapt	60.5	58.3	81.2	83.9	88.0	80.0	80.0	86.3	83.5
5	LaBSE-kNN	44.7	48.5	66.0	70.8	77.1	84.1	79.1	83.1	75.6
6	Weighted LaBSE-kNN	44.8	38.3	52.1	71.7	85.4	82.4	79.5	83.7	81.0
7	CE kNN$^+$ + $\mathcal{M}_{feature}^{\text{XLM-R}}$	58.9	63.8	78.5	80.4	83.8	86.2	77.6	83.5	85.4
8	CE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}}$	59.4	**67.0**	84.4	84.8	88.0	86.3	83.8	83.0	86.5
9	CE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ $\rightarrow$ SRC	61.2	61.1	**85.0**	**89.5**	92.3	**90.6**	84.9	89.5	87.3
10	BE kNN$^+$ + $\mathcal{M}_{feature}^{\text{XLM-R}}$	52.2	60.3	75.0	81.6	80.8	77.9	78.0	79.6	79.6
11	BE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}}$	58.8	56.6	80.6	83.8	86.9	82.2	**86.9**	84.9	83.7
12	BE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ $\rightarrow$ SRC	59.1	59.5	81.6	88.7	90.7	87.6	86.3	**90.2**	**88.7**

Table 5.2: Comparison of F1 values of the baselines and our model variants. BE kNN$^+$ and CE kNN$^+$ indicate Bi-encoder and Cross-encoder schemes, respectively. SRC indicates that the model has been further pre-trained with source Jigsaw English, having data from it as both query and neighbours.

5.4.2.2 Source Adaptation

XLM-R Mix-Adapt is a baseline model, which we train by mixing source and target data. This is possible because the label inventories of our source and target dataset are the same: $\mathcal{Y} = \{flagged, neutral\}$. The mixing is done by oversampling the target data to match the number of instances of the source dataset. As the number of instances in the target dataset is limited, this is preferable to undersampling.

XLM-R Seq-Adapt is a Transformer pre-trained on source data and then fine-tuned on target data (Garg et al., 2020). Here, we fine-tune XLM-R on the Jigsaw English dataset, and then we perform a second round of fine-tuning on the target dataset.

5.4.2.3 Nearest Neighbor

We apply two nearest neighbor baselines, using majority voting for label aggregation. We varied the number of neighbors from 3 to 20, and find that using 10 neighbors works best on average.

LaBSE-kNN is a baseline where the source dataset is indexed using representations obtained from LaBSE sentence embeddings, and neighbors are retrieved using cosine similarity.

Weighted LaBSE-kNN is a baseline that uses the same retrieval step as LaBSE-kNN, but uses a weighted voting strategy: each label is scored by summing the cosine similarities for the retrieved flagged and neutral neighbors respectively; then, the label with the highest score is returned.

5.4.3 Evaluation Measures

Following prior work on abusive language detection, we use F1 measure for evaluation. The F1 measure combines precision and recall (using a harmonic mean), which are both important to consider for automatic abusive language detection systems. In particular, online platforms strive to remove all content that violates their policies, and thus, if the system were to achieve 100% recall, the contents could be further filtered by human moderators to weed out the benign content. However, if the system's precision were very low, it would mean that the moderators would have to read every piece of content on the platform.

5.4.4 Fine-Tuning and Hyper-Parameters

We train all the models for 10 epochs with XLM-R as a base transformer representation with a maximum sequence length of 256 tokens. However, we make an exception for SRC (see Section 5.4.5): we train it for a single epoch, as training a neighbourhood-based model on a large dataset is resource-intensive. For all the approaches, we use Adam with β_1 0.9, β_2 0.999, ϵ 1e-08 as the optimiser setting. For the

baseline models, we use a batch size of 64, and a learning rate of 4e-05. For kNN$^+$-based models, we create a training batch from a query and its 10-nearest neighbours. For stable updates, we accumulate gradients from 50 batches before back-propagation. We selected the values of all of the aforementioned hyper-parameters based on the validation set. For kNN$^+$-based models, the best learning rate is selected from {5e-05, 7e-05}.

5.4.5 Experimental Results

Table 5.2 shows the effectiveness of our model variants compared to six strong baselines (rows 1-6). The highlighted rows in the table indicate different variants of our proposed framework. The variants fall into two categories of representation learning: CE kNN$^+$ and BE kNN$^+$, indicating the cross- and the bi-encoders in our framework, respectively. For each of the encoding schemes, we instantiate three different models by using three different pre-trained representations fine-tuned in our neighborhood framework, namely: $\mathcal{M}^{\text{XLM-R}}_{feature}$, which is pre-trained XLM-RoBERTa (XLM-R); $\mathcal{M}^{\text{P-XLM-R}}_{feature}$, which is XLM-R fine-tuned with **paraphrase** data and *parallel* data (Reimers and Gurevych, 2020); and $\mathcal{M}^{\text{P-XLM-R}}_{feature} \rightarrow$ SRC, which is $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ fine-tuned with source data (here, Jigsaw English) in our neighborhood framework. To train with SRC, we use all training data in Jigsaw English, and we retrieve neighbors from Jigsaw English using LaBSE sentence embeddings.[1] Then, we use this training data to fine-tune $\mathcal{M}^{\text{P-XLM-R}}_{feature}$ with our kNN$^+$-based cross- (CE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature} \rightarrow$ SRC) and bi- (BE kNN$^+$ + $\mathcal{M}^{\text{P-XLM-R}}_{feature} \rightarrow$ SRC) encoder setups. This is analogous to sequential adaptation (Garg et al., 2020), but in a neighborhood framework.

The SRC approach addresses one of the weaknesses of our kNN framework. The training data is created from instances in the target dataset and their neighbors from the source dataset. Thus, the neighborhood model cannot use all source training data,

[1]Note that we only use LaBSE for retrieval, as it has a large coverage of languages.

as it pre-selects a subset of source data based on similarity. This is a disadvantage compared to the sequential adaptation model, which uses all source training instances for pre-training. To overcome this, we use the neighborhood approach to pre-train our models with source data.

In Table 5.2, we report the F1 scores for eight language-specific training and evaluation sets stemming from two different data sets: Jigsaw Multilingual and WUL. Jigsaw Multilingual is an imbalanced dataset with 15% abusive content and WUL is balanced (see Table 5.1). Thus, it is hard to obtain high F1 score in Jigsaw Multilingual, whereas for WUL the F1 scores are relatively higher. Our CE kNN$^+$ variants achieve superior performance to all the baselines and our BE kNN$^+$ variants as well in the majority of the cases. The performance of the best and of the second-best models for each language are highlighted by **bold-facing** and underlining, respectively. We attribute the higher scores achieved by CE kNN$^+$ variants compared to the BE kNN$^+$ on the late-stage interaction of the query and the neighbors.

The CE kNN$^+$ variants show a large performance gain compared to baseline models on the Italian and the Turkish test sets from Jigsaw Multilingual. Even though the additional SRC pre-training is not always helpful for the CE kNN$^+$ model, it is always helpful for the BE kNN$^+$ model. However, both models struggle to outperform the baseline for the Spanish test set. We analysed the training data distribution for Spanish, but we could not find any noticeable patterns. Yet, it can be observed that the XLM-R Target baseline for Spanish (2nd row, 1st column) achieves a higher F1 score compared to the Seq-Adapt baseline, which yields better performance for Italian and Turkish. We believe that the in-domain training examples are good enough to achieve a reasonable performance on Spanish.

On the WUL dataset, the BE kNN$^+$ + $\mathcal{M}_{feature}^{\text{P-XLM-R}}$ variant with SRC pre-training outperforms the CE kNN$^+$ variants and all the baselines for Russian, Turkish, and Albanian. Both the BE kNN$^+$ variants and the CE kNN$^+$ variants perform worse

compared to the XLM-R Mix-Adapt baseline for English. Seq-Adapt is a recently published effective baseline (Garg et al., 2020), but for the WUL dataset, it does not perform well compared to the Mix-Adapt baseline.

5.4.5.1 Evaluation in a Multilingual Setting

In this subsection, we go beyond our cross-lingual setting and we analyse the effectiveness of our proposed model in a multilingual setting. A multilingual setting has been explored in recent work on abusive language detection (Pamungkas and Patti, 2019; Ousidhoum et al., 2019; Basile et al., 2019; Ranasinghe and Zampieri, 2020; Corazza et al., 2020; Glavaš et al., 2020; Leite et al., 2020) and it is desirable because online platforms are not limited to specific languages. An effective multilingual model unifies the two-stage process of language detection and prediction with a language-specific classifier. Moreover, abusive language is generally code-mixed (Saumya et al., 2021), which makes language-agnostic representation spaces more desirable.

We investigate a multilingual scenario, where all target languages in our cross-lingual setting are observed both at training and at testing time. To this end, we create new training, development, and testing splits in a 5:1:2 ratio from the 8,000 available data cases in the Jigsaw Multilingual dataset. Each split contains randomly sampled data in Italian, Spanish, and Turkish.

We train and evaluate our BE kNN$^+$ and CE kNN$^+$ using the aforementioned splits; the results are shown in Table 5.3. Here, we must note that our neighborhood retrieval model is language-agnostic, and thus we can retrieve neighbors for queries in any language.

We find that in a multilingual scenario, our BE kNN$^+$ model with SRC pre-training performs better than the CE kNN$^+$ model. Both the BE and the CE approaches supersede the best baseline model Seq-Adapt. Compared to the cross-lingual setting, there is more data in a mix of languages available. We hypothesise that the success

Model	Representations	F1
Seq-Adapt	XLM-R	64.4
CE-kNN	$\mathcal{M}_{feature}^{\text{XLM-R}}$	64.2
	$\mathcal{M}_{feature}^{\text{P-XLM-R}}$	62.8
	$\mathcal{M}_{feature}^{\text{P-XLM-R}} \rightarrow$ SRC	65.1
BE-kNN	$\mathcal{M}_{feature}^{\text{XLM-R}}$	65.5
	$\mathcal{M}_{feature}^{\text{P-XLM-R}}$	63.7
	$\mathcal{M}_{feature}^{\text{P-XLM-R}} \rightarrow$ SRC	**67.6**

Table 5.3: Effectiveness of our BE kNN$^+$ and CE kNN$^+$ schemes in the multilingual setting that we create from Jigsaw Multilingual.

of the bi-encoder model over the cross-encoder one stems from the increase in data size.

5.5 Summary

We proposed kNN$^+$, a framework for cross-lingual content flagging, which significantly outperforms strong baselines with limited training data in the target language. We further show the effectiveness of our framework in a multilingual scenario, where a test data point can be in Turkish, Italian or Spanish. We also provide a qualitative analysis of the representations learned by our BE kNN$^+$ framework variant, and show that flagged and non-flagged contents stay close in that representation space. Even though our framework is interpretable by design, in future, we plan to analyse it using human-centered evaluation. We also plan to evaluate our framework on other content flagging tasks as the framework is not limited to abusive content detection.

CHAPTER 6

CONCLUSIONS AND FURTHER WORK

In this thesis, we stressed that even though data is available to solve a task, it is scarce to solve a problem such as hate speech detection. We recommended that, to be certain of the progress in solving a problem, it is essential to address data scarcity settings such as QBE, zero-shot, and limited data (section 2.3) by using datasets from multiple tasks and evaluating model performance across tasks. We provided a discussion on techniques to solve data scarcity (section 2.4), and conducted experiments based on a number of approaches to tackle data scarcity in event retrieval and abusive language detection. We selected the event retrieval task because data scarcity in event retrieval is a challenge for social scientists who want to monitor various activities of different influential political actors, but cannot use existing event-annotated datasets to train models because they do not provide comprehensive coverage of all types of actors and events.

Data scarcity is also prevalent in abusive language and hate speech detection – but in a rather counter-intuitive way. There are a number of datasets for abusive language and hate speech detection. From that perspective, it might seem that data scarcity is a less severe problem in these areas. However, these datasets vary to a large degree based on annotation guidelines, language, and domain. The definition of abuse and hate speech vary widely across cultures making it very difficult to develop a model that performs well across tasks and achieves generalizability.

We motivated different data-scarce settings and developed solutions based on data augmentation and transfer learning to tackle data scarcity in event retrieval and abu-

sive language detection. For event retrieval, we proposed a data-scarce setting, QBE for events, for the first time in literature, and showed that a sentence-embedding based transfer learning approach is effective in event matching. We show that segmenting a sentence into events using a rule-based Semantic Role Labeling (SRL) model boosts the performance of the transfer and outperforms all the strong baselines. We evaluated the performance of this approach in three QBE settings: QBE-PoliceKilling, QBE-ACE, and QBE-IndiaPoliceEvents. In all the settings, our approach outperformed all the strong baselines. Specifically, for the QBE-ACE setting we showed that given 10 examples our approach achieves a 5% absolute improvement for precision at top-10 over the RM3 baseline.

To address data scarcity in hate speech detection, we proposed an unsupervised domain adaptation approach to augment labeled data for hate speech detection. Specifically, we proposed to convert a large collection of general domain negative emotion sentences into target domain specific hate speech using unlabeled data from the target domain along with a hate speech lexicon. We evaluated the effectiveness of the augmented data with three different models (character CNNs, BiLSTMs and BERT) on three different collections. We showed that our approach improves Area under the Precision/Recall curve by as much as 42% and recall by as much as 278%, with no loss (and in some cases a significant gain) in precision.

To address the data scarcity in abusive language detection we proposed a data-scarce setting based on limited abusive language data in a language and abundant abusive language data in English. We demonstrated that neighborhood methods, such as kNN are viable candidates for solving the cross-lingual abusive language detection task. We proposed a novel framework, $k\text{NN}^+$, which, unlike a classic kNN, models the relationship of a data point and each of its neighbors to represent the neighborhood, using language-agnostic transformers. Our evaluation on eight languages from two different datasets for abusive language detection showed sizeable improvements of

up to 9.5 F1 points over strong baselines. Our neighborhood framework creates an opportunity for continuous data augmentation without re-training and changing inference. This is fascinating because augmentation of novel data cases in existing models requires re-training of the whole model to take advantage of the additional knowledge made available through augmentation. We do not require that and this is a step forward towards continual transfer learning.

6.1 Future Work

The framework is intuitive, and it is a simple formalization of a machine learning ecosystem. Yet, it opens a number of avenues for further research. Based on our experiences in the application of this framework, we found that leveraging the problem-specific properties in both synthetic data generation (chapter 4) and transfer learning (chapter 3) is helpful. In chapter 4, we leverage one of the many characteristics of hate speech to generate hate speech templates, which we later fill with task-specific lexicons to generate task-specific hate speech. In chapter 3, we use a generic event segmentation algorithm, PredPatt, to create event-based segments from a sentence. When we match an event in a query sentence with an event in a corpus sentence, we score each of the segments from the corpus sentence, which helps us to obtain a precise matching score. We could apply an event-based segmentation of a document because our problem is event retrieval. The segmentation of a document into constituents that are effective for ad-hoc retrieval is a long-standing problem, and we could do it because of the specificity of our retrieval problem statement.

The performance of our models on both the problems that we address is still far from what we hope for. For the event retrieval task, we want to retrieve all the relevant events from a corpus because social scientists want to compute statistics from those events. This is an instance of a high-recall retrieval task and we did not explore the active learning avenue to evaluate total recall in this thesis. This

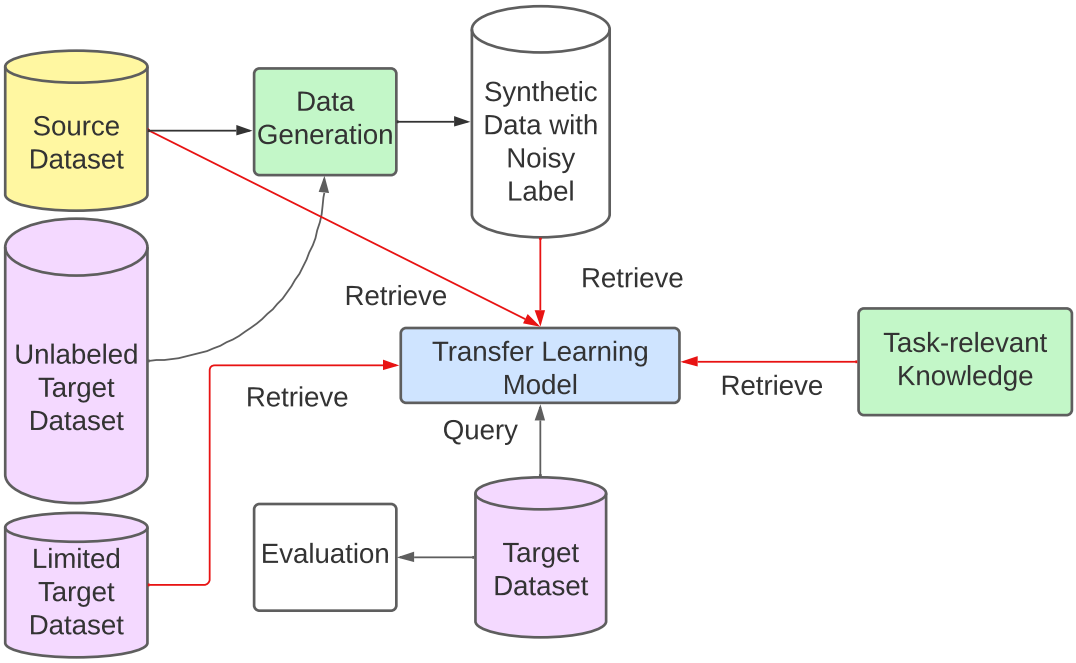


Figure 6.1: Our Augment-Transfer framework as a solution to address data scarcity.

is because we do not have a large corpus of different events for running evaluation and concluding performance on total recall. We made an attempt to do this in our IndiaPoliceEvents dataset by annotating all the documents in a corpus that is based on five events that social scientists are interested in. However, such an annotation process is prohibitively expensive and we could annotate only twenty-one thousand sentences, whereas in a typical retrieval dataset there are at least a hundred thousand documents. We recommend the development of a large-scale event detection corpus to address this. The first step we could imagine is to take an existing corpus, and an event query then increase the number of that specific event in that corpus through data generation or retrieval and getting those events annotated. In this way, we can evaluate high-recall retrieval for a few important queries.

A technical aspect of event retrieval that we want to further explore is the difference between the semantic space learned based on event similarity and sentence

similarity, and how to combine the individual strengths of these spaces to create a better semantic space for event matching. Our qualitative analysis revealed that a semantic space learned based on sentence similarity is topical. For example, if a query sentence contains "United States" the retrieved sentences will be generally focused on several aspects of the United States such as state names, president names, etc. However, if the query intent is to refer to "United States" as an agent taking part in different events, a sentence-similarity-based approach fails, because it does not capture syntactic information, which is essential to models events. However, in our experiments, a sentence embedding model worked reasonably well. We encourage further investigations to create be representation space for event matching.

Finally, our proposed neighborhood approach in chapter 5 for abusive language detection showed how to achieve cross-lingual transfer for abuse detection. This framework can be extended to other cross-lingual text classification tasks. Moreover, our framework provides the flexibility of continual adaptation of resource-rich monolingual data and offers interpretability. We did not explore these properties of our model through experimentation. In the future, we plan to investigate the seamless augmentation and transfer in our Augment-Transfer framework.

BIBLIOGRAPHY

Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., Smucker, M. D., and Wade, C. (2004). Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189.

ADL (2020). Hate and harassment report: The american experience 2020. `https://www.adl.org/online-hate-2020` visited 2020. Anti-Defamation League.

Agrawal, S. and Awekar, A. (2018a). Deep learning for detecting cyberbullying across multiple social media platforms. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*.

Agrawal, S. and Awekar, A. (2018b). Deep learning for detecting cyberbullying across multiple social media platforms. In *ECIR '18*.

Ahn, D. (2006). The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Allan, J. (2002). *Topic Detection and Tracking: Event-Based Information Organization*. Springer Publishing Company, Incorporated.

Arango, A., Pérez, J., and Poblete, B. (2019). Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19.

Artetxe, M. and Schwenk, H. (2019). Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Badjatiya, P., Gupta, M., and Varma, V. (2019). Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *WWW '19*.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *IN PROCEEDINGS OF THE COLING-ACL*, pages 86–90.

Banko, M., MacKeen, B., and Ray, L. (2020). A unified taxonomy of harmful content. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 125–137, Online.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., and Sanguinetti, M. (2019). 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.

Bassignana, E., Basile, V., and Patti, V. (2018). Hurtlex: A multilingual lexicon of words to hurt. In *Proceedings of the Fifth Italian Conference on Computational Linguistics*, volume 2253 of *CLiC-it 18*. CEUR-WS.org.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *In SIGMOD Conference*, pages 1247–1250.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *EMNLP*.

Brathwaite, R. and Park, B. (2018). Measurement and conceptual approaches to religious violence: The use of natural language processing to generate religious violence event-data. *Politics and Religion*, pages 1–42.

Bron, M., Balog, K., and de Rijke, M. (2010). Ranking related entities. In *CIKM '10*.

Burnap, P. and Williams, M. L. (2015). Cyber hate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2).

Carreras, X. and Màrquez, L. (2005). Introduction to the conll-2005 shared task: Semantic role labeling. In *CoNLL '05*.

Chen, X. and Cardie, C. (2018). Multinomial adversarial networks for multi-domain text classification. In *NAACL '18*.

Chen, Y., Liu, S., Zhang, X., Liu, K., and Zhao, J. (2017). Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.

Chenoweth, E. and Lewis, O. A. (2013). Unpacking nonviolent campaigns introducing the NAVCO 2.0 dataset. *Journal of Peace Research*, 50(3):415–423.

Chidambaram, M., Yang, Y., Cer, D., Yuan, S., Sung, Y., Strope, B., and Kurzweil, R. (2019). Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model. In *Proceedings of the 4th Workshop on Representation Learning for NLP*, RepL4NLP '19, pages 250–259, Florence, Italy. Association for Computational Linguistics.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chung, Y.-L., Kuzmenko, E., Tekiroglu, S. S., and Guerini, M. (2019). CONAN - COunter NArratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829, Florence, Italy. Association for Computational Linguistics.

Cohen, D., Mitra, B., Hofmann, K., and Croft, W. B. (2018). Cross domain regularization for neural ranking models using adversarial learning. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM.

Commission, E. (2022). Shaping europe's digital future: The digital services act package. `https://ec.europa.eu/digital-single-market/en/digital-services-act-package`. Accessed: 2022-03-01.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 8440–8451.

Corazza, M., Menini, S., Cabrio, E., Tonelli, S., and Villata, S. (2020). A Multilingual Evaluation for Online Hate Speech Detection. *ACM Trans. Internet Technol.*, 20(2).

Cormack, G. V., Clarke, C. L. A., and Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR '09*.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3).

Daumé III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Daumé III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126.

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

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Daxecker, U., Amicarelli, E., and Jung, A. (2019). Electoral Contention and Violence (ECAV): A new dataset. *Journal of Peace Research*, 56(5):714–723.

De Gibert, O., Perez, N., García-Pablos, A., and Cuadros, M. (2018). Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal Stanford dependencies: A cross-linguistic typology. In *LREC '14*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186, Minneapolis, Minnesota, USA.

Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., and Bhamidipati, N. (2015). Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30.

Du, Y., Ma, T., Wu, L., Xu, F., Zhang, X., Long, B., and Ji, S. (2021). Constructing contrastive samples via summarization for text classification with limited annotations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1365–1376, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ethayarajh, K. (2019). How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China.

Fariha, A., Sarwar, S. M., and Meliou, A. (2018). Squid: Semantic similarity-aware query intent discovery. In *SIGMOD '18*.

Fehn Unsvåg, E. and Gambäck, B. (2018). The effects of user features on Twitter hate speech detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 75–85, Brussels, Belgium.

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2020). Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Ferguson, J., Lockard, C., Weld, D., and Hajishirzi, H. (2018). Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 359–364, New Orleans, Louisiana. Association for Computational Linguistics.

Foley, J., Sarwar, S. M., and Allan, J. (2018). Named entity recognition with extremely limited data. In *1st International Workshop on Learning from Limited or Noisy Data for Information Retrieval*.

Fortuna, P., Ferreira, J., Pires, L., Routar, G., and Nunes, S. (2018). Merging datasets for aggressive text identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 128–139, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., and Kourtellis, N. (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4).

Galuscáková, P., Oard, D. W., and Nair, S. (2021). Cross-language information retrieval. *ArXiv*, abs/2111.05988.

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.

Garg, S., Vu, T., and Moschitti, A. (2020). Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7780–7788.

Geng, Q., Chuai, Z., and Jin, J. (2022). Webpage retrieval based on query by example for think tank construction. *Inf. Process. Manag.*, 59:102767.

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., and Plagianakos, V. P. (2018). Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, SETN '18, New York, NY, USA.

Glavaš, G., Karan, M., and Vulić, I. (2020). XHate-999: Analyzing and detecting abusive language across domains and languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365, Barcelona, Spain (Online).

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*.

Government, U. (2022). Online harms white paper. `https://www.gov.uk/government/consultations/online-harms-white-paper/online-harms-white-paper`. Accessed: 2022-03-01.

Guo, H. (2020). Nonlinear mixup: Out-of-manifold data augmentation for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4044–4051.

Halterman, A., Keith, K., Sarwar, S. M., and O'Connor, B. (2021). Corpus-level evaluation for event qa:the indiapoliceevents corpus covering the 2002 gujarat violence. In *Findings of the Association for Computational Linguistics: ACL 2021*, Online. Association for Computational Linguistics.

Hartmann, S., Kuznetsov, I., Martin, T., and Gurevych, I. (2017). Out-of-domain FrameNet semantic role labeling. In *EACL '17*.

He, R., Lee, W. S., Ng, H. T., and Dahlmeier, D. (2018). Adaptive semi-supervised learning for cross-domain sentiment classification. In *EMNLP '18*.

Hsi, A., Yang, Y., Carbonell, J., and Xu, R. (2016). Leveraging multilingual training for limited resource event extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1201–1210, Osaka, Japan. The COLING 2016 Organizing Committee.

Hu, M., Wu, Y., Zhao, S., Guo, H., Cheng, R., and Su, Z. (2019). Domain-invariant feature distillation for cross-domain sentiment classification. In *EMNLP '19*.

Human Rights Watch (2002). "We have no orders to save you": State participation and complicity in communal violence in Gujarat. *Human Rights Watch Report*, 14(3).

Jigsaw (2018). Toxic comment classification challenge. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/. Online; accessed 28 February 2021.

Jigsaw Multilingual (2020). Jigsaw multilingual toxic comment classification. https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification/. Online; accessed 28 February 2021.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain.

Jürgens, D., Hemphill, L., and Chandrasekharan, E. (2019). A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–3666, Florence, Italy.

Karan, M. and Šnajder, J. (2018). Cross-domain detection of abusive language online. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*.

Keith, K., Handler, A., Pinkham, M., Magliozzi, C., McDuffie, J., and O'Connor, B. (2017). Identifying civilians killed by police with distantly supervised entity-event extraction. In *EMNLP '17*, pages 1547–1557.

Kukacka, J., Golkov, V., and Cremers, D. (2017). Regularization for deep learning: A taxonomy. *CoRR*, abs/1710.10686.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2020). Evaluating aggression identification in social media. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 1–5, Marseille, France. European Language Resources Association (ELRA).

Kumaran, G. and Allan, J. (2004). Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304.

Kwok, I. and Wang, Y. (2013). Locate the hate: Detecting tweets against blacks. In desJardins, M. and Littman, M. L., editors, *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*. AAAI Press.

Lai, V. D. and Nguyen, T. (2019). Extending event detection to new types with learning from keywords. In *W-NUT '19*.

Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Leite, J. A., Silva, D., Bontcheva, K., and Scarton, C. (2020). Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis. 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 914–924, Suzhou, China.

Lin, B. Y. and Lu, W. (2018). Neural adaptation layers for cross-domain named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding. In *ICLR*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692.

Longpre, S., Wang, Y., and DuBois, C. (2020). How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4401–4411, Online. Association for Computational Linguistics.

Lu, W., Shen, Y., Chen, S., and Ooi, B. C. (2012). Efficient processing of k nearest neighbor joins using mapreduce. *Proc. VLDB Endow.*, 5(10):1016–1027.

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

Majumder, P. and Patel, D. (2019). Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *FIRE '19*.

Makkonen, J., Ahonen-Myka, H., and Salmenkivi, M. (2004). Simple semantics in topic detection and tracking. *Information Retrieval*, 7:347–368.

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

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., and Patel, A. (2019). 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*, FIRE '19, page 14–17, New York, NY, USA.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Inc, P., Bethard, S. J., and Mcclosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL '14*.

Mathur, P., Shah, R., Sawhney, R., and Mahata, D. (2018). Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26.

Metzger, S., Schenkel, R., and Sydow, M. (2017). Qbees: query-by-example entity search in semantic knowledge graphs based on maximal aspects, diversity-awareness and relaxation. *Journal of Intelligent Information Systems*, pages 333–366.

Metzler, D., Cai, C., and Hovy, E. (2012). Structured event retrieval over microblog archives. In *NAACL '12*.

Min, B., Grishman, R., Wan, L., Wang, C., and Gondek, D. (2013). Distant supervision for relation extraction with an incomplete knowledge base. In *NAACL*.

Mintz, M. D., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *ACL*.

Mueller, H. and Rauh, C. (2017). Reading between the lines: Prediction of political violence using newspaper text. *American Political Science Review*, pages 1–18.

Müller, K. and Schwarz, C. (2020). Fanning the flames of hate: Social media and hate crime. *Political Economy - Development: Public Service Delivery eJournal*.

Murdock, V. (2007). Aspects of sentence retrieval. *SIGIR Forum*, 41:127.

Nakov, P., Nayak, V., Dent, K., Bhatawdekar, A., Sarwar, S. M., Hardalov, M., Dinkov, Y., Zlatkova, D., Bouchard, G., and Augenstein, I. (2021). Detecting Abusive Language on Online Platforms: A Critical Analysis.

Nguyen, T. H. and Grishman, R. (2015). Event detection and domain adaptation with convolutional neural networks. In *ACL '15*.

Nie, J. (2010). *Cross-Language Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Nockleby, J. T. (1994). Hate speech in context: The case of verbal threats. *Buffalo Law Review*, 42:653.

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., and Yeung, D.-Y. (2019). Multilingual and multi-aspect hate speech analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4675–4684, Hong Kong, China.

Pamungkas, E. W. and Patti, V. (2019). Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *ACL Student Research Workshop '19*.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359.

Parekh, B. (2012). *Is There a Case for Banning Hate Speech?*, page 37–56. Cambridge University Press.

Pavlopoulos, J., Malakasiotis, P., and Androutsopoulos, I. (2017). Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Phang, J., Févry, T., and Bowman, S. R. (2018). Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *ArXiv*, abs/1811.01088.

Phang, J., Htut, P. M., Pruksachatkun, Y., Liu, H., Vania, C., Kann, K., Calixto, I., and Bowman, S. R. (2020). English intermediate-task training improves zero-shot cross-lingual transfer too. In *AACL*.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281. ACM.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. R. (2020). Intermediate-task transfer learning with pretrained language models: When and why does it work? In *ACL*.

Pustejovsky, J. and Stubbs, A. (2012). *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc.

Qu, X., Zou, Z., Cheng, Y., Yang, Y., and Zhou, P. (2019). Adversarial category alignment network for cross-domain sentiment classification. In *NAACL '19*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*.

Radovanović, M., Nanopoulos, A., and Ivanović, M. (2009). Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 865–872, New York, NY, USA.

Raffel, C., Shazeer, N. M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Ranasinghe, T. and Zampieri, M. (2020). Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5838–5844, Online.

Rasolofo, Y. and Savoy, J. (2003). Term proximity scoring for keyword-based retrieval systems. In *European Conference on Information Retrieval*, pages 207–218. Springer.

Ratner, A. J., Bach, S. H., Ehrenberg, H. R., Fries, J. A., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, 11 3:269–282.

Reimers, N. and Gurevych, I. (2018). Event nugget detection, classification and coreference resolution using deep neural networks and gradient boosted decision trees. *Transfer*, page 554.

Reimers, N. and Gurevych, I. (2019a). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP '19*.

Reimers, N. and Gurevych, I. (2019b). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China.

Reimers, N. and Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online.

Reinsel, D., Gantz, J., and Rydning, J. (2018). The digitization of the world from edge to core. Technical report, IDC, 5 Speen Street, Framingham, MA 01701, USA.

Rizoiu, M.-A., Wang, T., Ferraro, G., and Suominen, H. (2019). Transfer learning for hate speech detection in social media.

Ruder, S. (2019). *Neural Transfer Learning for Natural Language Processing*. PhD thesis, NATIONAL UNIVERSITY OF IRELAND, GALWAY.

Rudra, K., Goyal, P., Ganguly, N., Mitra, P., and Imran, M. (2018). Identifying subevents and summarizing disaster-related information from microblogs. In *SIGIR '18*.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3859–3869, Red Hook, NY, USA.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sarwar, S., Foley, J., Yang, L., and Allan, J. (2019a). Sentence retrieval for entity list extraction with a seed, context, and topic. In *ICTIR '19*.

Sarwar, S. M. and Allan, J. (2019). SearchIE: A retrieval approach for information extraction. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '19, page 249–252.

Sarwar, S. M. and Allan, J. (2020). Query by example for cross-lingual event retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1601–1604.

Sarwar, S. M., Bonab, H., and Allan, J. (2019b). A multi-task architecture on relevance-based neural query translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Sarwar, S. M., Foley, J., and Allan, J. (2018). Term relevance feedback for contextual named entity retrieval. In *CHIIR '18*, pages 301–304.

Sarwar, S. M. and Murdock, V. (2022). Unsupervised domain adaptation for hate speech detection using a data augmentation approach. abs/2107.12866.

Sarwar, S. M., Zlatkova, D., Hardalov, M., Dinkov, Y., Augenstein, I., and Nakov, P. (2021). A neighbourhood framework for resource-lean content flagging. *ArXiv*, abs/2103.17055.

Saumya, S., Kumar, A., and Singh, J. P. (2021). Offensive language identification in Dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 36–45, Kyiv, Ukrane. Association for Computational Linguistics.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media*, pages 1–10.

Schwartz, O. (2019). In 2016, microsoft's racist chatbot revealed the dangers of online conversation. `https://spectrum.ieee.org/tech-talk/artificial-intelligence/machine-learning/in-2016-microsofts-racist-chatbot-revealed-the-dangers-of-online-conversation`. visited August 2020.

Sharifirad, S., Jafarpour, B., and Matwin, S. (2018). Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Association for Computational Linguistics.

Shnarch, E., Alzate, C., Dankin, L., Gleize, M., Hou, Y., Choshen, L., Aharonov, R., and Slonim, N. (2018). Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605.

Shorten, C., Khoshgoftaar, T. M., and Furht, B. (2021). Text data augmentation for deep learning. *Journal of Big Data*, 8.

Smucker, M. D. and Allan, J. (2006). Find-similar: Similarity browsing as a search tool. In *SIGIR '06*.

Srivastava, S., Khurana, P., and Tewari, V. (2018). Identifying aggression and toxicity in comments using capsule network. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 98–105, Santa Fe, New Mexico, USA.

Stappen, L., Brunn, F., and Schuller, B. (2020). Cross-lingual zero-and few-shot hate speech detection utilising frozen transformer language models and axel. *arXiv preprint arXiv:2004.13850*.

Strohman, T., Metzler, D., Turtle, H. R., and Croft, W. B. (2005). Indri : A language-model based search engine for complex queries (extended version).

Subramanian, K. S. (2007). *Political violence and the police in India*. SAGE Publications India.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2021). Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Tian, T., Dinarelli, M., Tellier, I., and Cardoso, P. D. (2016). Domain adaptation for named entity recognition using CRFs. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Tran, B., Karimzadehgan, M., Pasumarthi, R. K., Bendersky, M., and Metzler, D. (2019). Domain adaptation for enterprise email search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 25–34.

Tufecki, Z. (2019). Youtube's recommendation algorithm has a dark side. https://www.scientificamerican.com/article/youtubes-recommendation-algorithm-has-a-dark-side/. visited August 2020.

van Aken, B., Risch, J., Krestel, R., and Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 33–42, Brussels, Belgium.

Vandenhende, S., Georgoulis, S., Gansbeke, W. V., Proesmans, M., Dai, D., and Gool, L. V. (2021). Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, PP.

Vercoustre, A., Thom, J., and Pehcevski, J. (2008). Entity ranking in Wikipedia. In *SAC '08*.

Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., and Margetts, H. (2019). Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93, Florence, Italy.

Wadden, D., Wennberg, U., Luan, Y., and Hajishirzi, H. (2019). Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Walker, C. e. a. (2006). Ace 2005 multilingual training corpus.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020a). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. (2020b). Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3).

Warner, W. and Hirschberg, J. (2012). Detecting hate speech on the world wide web. In *Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012)*. Association for Computational Linguistics.

Waseem, Z. (2016). 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, Austin, Texas. Association for Computational Linguistics.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of NAACL SRW*.

Waseem, Z., Thorne, J., and Bingel, J. (2018). *Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection*, pages 29–55. Springer International Publishing, Cham.

Watanabe, H., Bouazizi, M., and Ohtsuki, T. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, 6:13825–13835.

Wei, J. and Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP-IJCNLP '19*.

White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R., Rawlins, K., and Van Durme, B. (2016). Universal decompositional semantics on universal dependencies. In *EMNLP '16*.

Wiegand, M., Ruppenhofer, J., and Kleinbauer, T. (2019). Detection of Abusive Language: the Problem of Biased Datasets. 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)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Wiegand, M., Siegel, M., and Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language.

Wilkinson, S. I. (2006). *Votes and violence: Electoral competition and ethnic riots in India*. Cambridge University Press.

Wu, X., Lv, S., Zang, L., Han, J., and Hu, S. (2019). Conditional bert contextual augmentation. In *ICCS*.

Wulczyn, E., Thain, N., and Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. (2019). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2251–2265.

Xue, Q., Zhang, W., and Zha, H. (2020). Improving domain-adapted sentiment classification by deep adversarial mutual learning. In *AAAI '20*.

Yang, B. and Mitchell, T. M. (2016). Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.

Yang, J. and Zhang, Y. (2018). Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Yang, S., Feng, D., Qiao, L., Kan, Z., and Li, D. (2019). Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019). 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.

Zampieri et al., M. (2020). SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *SemEval*.

Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., and Chen, E. (2019). Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5773–5780.

Zhang, S., Rudinger, R., and Durme, B. V. (2017). An evaluation of PredPatt and open IE via stage 1 semantic role labeling. In *IWCS 2017*.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc.

Zhao, W., Eger, S., Bjerva, J., and Augenstein, I. (2020). Inducing Language-Agnostic Multilingual Representations. *arXiv preprint arXiv:2008.09112*.

Zweigenbaum, P., Sharoff, S., and Rapp, R. (2017). Overview of the Second BUCC Shared Task: Spotting Parallel Sentences in Comparable Corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, Vancouver, Canada. Association for Computational Linguistics.