Minimizing Domain Bias When Adapting Sentiment Analysis Techniques to the Legal Domain

Gathika Ratnayaka

198051D

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science in Computer Science and Engineering

Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

April 2022

DECLARATION

I, Gathika Ratnayaka, declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters thesis/Dissertation under my supervision.

Name of Supervisor: Dr. Amal Shehan Perera

Signature of the Supervisor: Date:

Name of Supervisor: Dr. Nisansa de Silva

Signature of the Supervisor: Date:

ABSTRACT

Sentiment Analysis can be considered as an integral part of Natural Language Processing with a wide variety of significant use cases related to different application domains. Analyzing sentiments of descriptions that are given in Legal Opinion Texts has the potential to be applied in several legal information extraction tasks such as predicting the judgement of a legal case, predicting the winning party of a legal case, and identifying contradictory opinions and statements. However, the lack of annotated datasets for legal sentiment analysis imposes a major challenge when developing automatic approaches for legal sentiment analysis using supervised learning. In this work, we demonstrate an effective approach to develop reliable sentiment annotators for legal domain while utilizing a minimum number of resources. In that regard, we made use of domain adaptation techniques based on transfer learning, where a dataset from a high resource source domain is adapted to the target domain (legal opinion text domain). In this work, we have come up with a novel approach based on domain specific word representations to minimize the drawbacks that can be caused due to the differences in language semantics between the source and target domains when adapting a dataset from a source domain to a target domain. This novel approach is based on the observations that were derived using several word representational and language modelling techniques that were trained using legal domain specific copora. In order to evaluate different word representational techniques in the legal domain, we have prepared a legal domain specific context based verb similarity dataset named *LeCoVe* . The experiments carried out within this research work demonstrate that our approach to develop sentiment annotators for legal domain in a low resource setting is successful with promising results and significant improvements over existing works.

Keywords: Sentiment Analysis; Deep Learning; Word Representation ; Semantic Analysis

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisors Dr. Amal Shehan Perera and Dr. Nisansa de Silva for the continuous support, motivation, and valuable insights which were tremendously helpful for the successful completion of this work. This research project would not have been possible without their valuable support.

I would also like to thank Dr. Uthayashanker Thayasivam, Dr. Charith Chithraranjan for their valuable feedback and advice related to the research.

Moreover, I would like to extend my gratitude to Mr. Gayan Kaviratne, Mr. Anajana Fernando, Mr. Ramesh Pathirana, and Ms. Thirasara Ariyaratne for the support given during this research work. I am also grateful for my father Mr. Dhammika Ratnayaka, my mother Mrs. Geethani Udugamakorale, and my sister Ms. Akshila Ratnayaka for the continuous support given to me throughout this journey.

Thank you!

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continous Bag of Words
ELMO	Embeddings from Language Model
NLP	Natural Language Processing
POS	Part-Of-Speech
RNN	Recurrent Neural Network
RNTN	Recursive Neural Tensor Network
SG	Skip Gram

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Background

Law and order are an integral part of human civilization. The legal systems have been evolved for centuries in order to match up with the emerging requirements of human civilizations. As a result, the accessibility of resources related to the legal domain is becoming more and more important. The World Wide Web enabled humans to make publishable legal resources easily accessible by digitalising them and publishing them on the internet. With the emergence of Artificial Intelligence related technologies such as machine learning and deep learning, it can be seen that there is an emerging trend to develop more sophisticated applications that can organize and extract valuable legal information in a useful manner with minimum human intervention.

A given Legal Opinion Texts may contain information which are potentially applicable in cases which have legal scenarios similar to the scenario that is considered in the Legal Opinion Text. More precisely, the related incidents, arguments, legal opinions and legal judgements are some of such information that can be used in a new similar legal scenario. As a result, legal officials make use of the information available in legal opinion texts to support their arguments related to a particular legal situation. Therefore, the development of automated systems that have the capability to support legal officials by extracting valuable information from legal documents such as legal opinion texts can be regarded as an impactful task.

This work is specifically focused on developing techniques to analyze sentiments in the descriptions that are available in Legal Opinion Texts. Sentiment analysis is a well known information extraction task that has several use cases over many domains. It can also be considered as an important but an under

explored information extraction task in the legal domain. When a legal case is considered, two major parties can be identified. One party bring up the lawsuit, and that party is commonly identified as the plaintiff. The opposition party to the plaintiff is usually called the defendant. The legal opinion texts usually contain descriptions about the ways in which parties are related to a specific incident, the actions performed by the related parties on a considered event and also about the arguments brought forward by each party when the legal case was proceeding. More importantly, legal opinion texts also contain Legal opinions or the opinions of judges related to a court case. Such opinions may have a direct impact on a party involved in a court case in a positive, neutral or negative manner. If some legal opinion has a positive impact on the p. Performing sentiment analysis on these descriptions will enable the automatic identification of the type of impact a particular precedent, statute, legal opinion, incident or an argument may have on a considered party. This can also be considered as a key step when developing systems that are capable of predicting the outcome of a court case.

In addition to the opinions that are directly related to the conduct of the parties, legal opinion texts also provide interpretations related to the previous judgements and also on statutes that are relevant to the legal case. Such opinions may elaborate on the justifications, purposes, drawbacks and loopholes that are associated with a particular statute or a precedent. Moreover, the descriptions also contain information related to the proceeding of court cases such as adjournment of the case and lack of evidence which can be considered as factors that can directly have an impact on the outcomes. For example, let's consider the Sentence 1.1 of Example 1 which was extracted from a Legal Opinion Text [1]. It can be seen that the description in Sentence 1.1 is favorable to Lee, who is the subject of that sentence. So, Sentence 1.1 has a sentiment which is positive to the subject of that sentence. If we consider Sentence 1.2, which is obtained from the same Legal Opinion Text [1], it can be seen that the description of the sentence is unfavorable to the subject of the sentence (The Government) and has a negative sentiment towards it.

Example 1

- Sentence 1.1: *Lee has demonstrated that he was prejudiced by his counsel's erroneous advice*
- Sentence 1.2: *The Government makes two errors in urging the adoption of a per se rule that a defendant with no viable defense cannot show prejudice from the denial of his right to trial.*

When all of the above mentioned factors are considered, sentiment analysis on legal opinion texts can be considered as a task that can facilitate a wide range of use cases. Despite its potential and usefulness, the attempts to perform sentiment analysis in the legal domain are limited. This study aims to address this issue by developing a sentiment annotator that can identify sentiments in a given sentence/phrase extracted from the legal opinion texts related to the United States Supreme Court. Information that can be derived from such a sentiment annotator can then be adapted to facilitate more downstream tasks such as identifying advantageous and disadvantageous arguments for a particular party, contradictory opinion detection [2], and predicting outcomes of legal cases [3] .

In order to develop a reliable sentiment annotator using supervised learning, it is required to have a large amount of labelled data to train the underlying classification model. However, creating such sophisticated datasets with manually annotated data (by domain experts) for a specialised domain like legal opinion texts is not practical due to extensive resource and time requirements [4, 5]. In a low resource setting, transfer learning can be used as a potential technique to overcome the requirement of creating a sophisticated data set, by leveraging information available in a already labelled data from another domain to perform sentiment analysis in the target domain. The sentiment annotators that are being widely used with English language are trained using data belongs to domains such as the movie review domain. Adapting these models directly into the legal domain will create drawbacks, especially due to the negative transfer; which is a phenomenon that occurs due to dissimilarities between two domains. Domain specific usage of words, domain specific meanings and sentiment polarities of

words can be considered as one major reason that causes negative transfer when adapting datasets/models from one domain to another domain [5].

In this thesis, we demonstrate novel techniques that can be effectively utilized to overcome drawbacks that occur because of negative transfer, when using a dataset from a source domain (other than the legal domain) to create information extraction tools to the legal domain. The propose methodologies are facilitated by an algorithmic approach developed to automatically identify words that can cause negative transfer when adapting a source dataset to the legal domain. Moreover, by utilizing the outcomes of the algorithmic approach, we propose two transfer learning mechanisms that enable the development of legal sentiment annotator with a minimum amount of resources and human annotations. The sentiment annotators proposed in this study are capable of performing 3 class sentiment classification where a given sentence is classified as having a positive or negative or neutral sentiment.

Our algorithmic approaches to perform sentiment analysis on legal domain make use of modern word representation and language representation techniques. Therefore, as a part of this study, we have also carried out extensive experiments to evaluate the effectiveness of various word embeddings and language representation techniques in the legal domain.

1.2 Research Objectives

Objectives of this research are as follows:

1. Developing a phrase level sentiment annotator to perform sentiment analysis on legal opinion texts

2. Coming up with a novel methodology to mitigate the effect of negative transfer when adapting sentiment analysis datasets from other domains to the legal opinion texts domain.

3. Evaluate the effectiveness of the word emebdding and language representation techniques n identifying words with similar meanings in the legal

domain.

1.3 Contributions

Within this work, the following contributions have been made:

- Developed a sentiment annotator to analyze the sentiments of legal opinions in legal opinion texts.
- Proposed a transfer learning based approach to develop a legal sentiment annotator. Within the proposed approach, there is an algorithmic approach that exploits domain specific word representation techniques to overcome negative transfer.
- Developed a verb similarity dataset that provides information related to the similarity of verbs based on the context it is being used and made it publicly available to the research community.
- Evaluated the performances of different word representational models considering the task of identifying verbs with similar meanings in the legal domain.

1.4 Publications

- **Gathika Ratnayaka**, Nisansa de Silva, Amal Shehan Perera, and Ramesh Pathirana, "Effective Approach to Develop a Sentiment Annotator For Legal Domain in a Low Resource Setting".
  - Conference : 34th Pacific Asia Conference on Language, Information and Computation ***(Published)***.
  - CORE rank of the conference: B
- **Gathika Ratnayaka**, Nisansa de Silva, Amal Shehan Perera, Gayan Kavirathne, Thirasara Ariyarathna, and Anjana Wijesinghe, "Context Sensitive Verb Similarity Dataset for Legal Information Extraction".

- Journal : Data by Multidisciplinary Digital Publishing Institute ***(Published)***.

- Rank: CiteScore - Q2 (Information Systems and Management)

Chapter 2

LITERATURE SURVEY

2.1 Sentiment Analysis

Early methodologies of sentiment analysis [6] have made use of sentiment lexicons such as Sentiwordnet [7], ANEW[8], and AFINN[9] to determine the sentiment of a textual unit. Different domains have been considered when developing such sentiment lexicons [9]. As a result, it can be observed that the sentiment polarity of a word and the strength of the sentiment associated with that particular word change from one sentiment lexicon to another. With the recent development of machine learning and deep learning, it can be observed that techniques based on machine learning and deep learning are widely applied for sentiment analysis. The algorithms/models used in such techniques are developed to automatically capture the sentiment of a word while learning how the compositions of different words affect the overall sentiment of a considered text. The Recursive Neural Tensor Network (RNTN) proposed by Socher et al. [10] is a seminal work in this direction. The RNTN model has shown promising results for sentiment classification in the movie review domain. However, more recent approaches that make use of pretrained language models (eg: BERT[11]) have surpassed the approaches that are based on recursive neural network architectures, becoming the state of the art for sentiment classification [12]. From this point onwards, the RNTN model proposed in [10] will be denoted as $RNTN_m$.

2.2 Sentiment Analysis in the Legal Domain

Even though the studies related to applying sentiment analysis to the legal domain are limited, the ways in which sentiment analysis can be used towards facilitating legal processes is being discussed in the law-tech community [13, 3]. Gamage et al. have proposed a methodology [4] to perform phrase level sentiment

analysis in US legal opinion texts. However, certain limitations that occur when applying their methodology can be identified. The sentiment annotator proposed by Gamage et al [4] focuses only on two sentiment classes, i.e. negative sentiment and non-negative sentiment, which can also be considered as a binary classification task. Identifying words that have different sentiments in the legal domain when compared to that of the movie domain can be considered as one of the key step of the method proposed in [4]. However, the study [4] uses a manual approach to identify words with domain-specific sentiments and the identification of such words had been performed manually by human annotators. However, such a manual setting is not ideal in a low resource context, as manually going through a set of words with a significant size is tedious and sometimes infeasible. In this work, our intention is to come up with a methodology that uses a limited amount of human annotations to develop a reliable sentiment annotator for the legal domain while not compensating the accuracy . The study by Sharma et al. [5] proposes an automatic approach that is based on word representations to minimize negative transfer. The key insight is to identify transferable words that can be used for cross domain sentiment classification. However, the approach proposed in the study [5] aims only at binary sentiment classification, i.e positive and negative sentiment classes.

2.3 Word Vector Representations and Language Modelling Systems

In order to provide the capability to computers to understand human language or to extract useful information from natural language text, the textual information should be converted into a machine readable format. Therefore, one of the main requirements in Natural Language Processing is to convert a word or a text into a numerical representation. There are several word representation techniques that have been developed while taking the semantic, syntactic and contextual properties of words into the account. Such techniques have proven useful when it comes to identifying similarities between words. Word2vec [14] and Glove [15] can be considered as examples for Neural Word Embedding approaches that cre-

ate distributional similarity based representations for words. However, one key drawback in most of these approaches is that they provide only one representation for a word. However, the same word can have many meanings/senses based on the context and also based on the considered domain. Sense2vec [16], while being a distributional similarity based word embedding technique similar to Word2Vec and Glove, attempts to provide multiple representations for a word based on the Part of Speech tagging of the considered word. However, Sense2Vec and other approaches that are based on distributional similarity to create word representations do not consider the context associated with a word when providing an embedding/representation for a particular word. Consequently, these word representation approaches would not be able to capture how the meaning of a word changes in relation to context and the domain. This drawback is addressed in Language Modelling techniques such as BERT [11],ELMO [17], and XLNet [18] in which the sequential context associated with a word is considered. The models that are pre-trained based on such language modelling techniques can be used to obtain context based representations for words. Moreover, such models have become an integral part of most of the state of the art techniques related to many Natural Language Processing tasks.

2.4 Domain Adaptation

Transfer learning attempts to adapt models that are trained on one task (source task) to another task (target task). Existing literature demonstrates that drawbacks are common when adapting models trained using data from prior tasks (sources) to a low resource task (target) [4]. If we consider information extraction tasks, a model trained to perform a particular information extraction task for a considered domain may not work well for the same information extraction task in another domain. For example, it has been shown that the sentiment analysis models that are trained using movie reviews (movie review domain) creates drawbacks when they are adapted to the Legal domain [4]. Such drawbacks are mainly due to the dissimilarities between the sources and target that ultimately hinders

the performance of the adapted model for the target domain. This phenomenon is known as Negative Transfer. Domain-specific behaviors of words (domain-specific terminology) and domain-specific semantics such as relationships between concepts/entities are considered to be major reasons that cause negative transfer when it comes to text classification tasks. However, it is still the case that transfer learning overall has positive effects. The current state of the art in most text understanding tasks uses pre-trained language models such as BERT [11] which allow general transfer of word knowledge. Then, this knowledge is transferred to perform specific tasks [12].

Active Learning can be considered as another domain adaptation strategy, which aims to significantly minimize the resources needed to perform data annotations by automatically querying data instances that are most informative for a learning model. For example, if we consider a domain adaptation task, the objective of active learning will be to find data instances that will best train a considered model the domain specific behaviors of the target domain. Those selected instances will then be annotated by domain experts, but the number of data instances that are needed to be annotated will be significantly reduced. When it comes to active learning, there are various querying strategies that are developed in order to identify the most important data instances to be annotated [19]. Another technique that can be used in low resource tasks is Data Augmentation [20]. In data augmentation, the objective is to increase the amount of training data by adding synthetic data that are created using the existing data.

2.5 Evaluation Resources on Verb Similarity

It is needed to evaluate the applicability of different word representation/language modelling techniques in the legal domain. Io that regard, we evaluated how different word representation and language modelling techniques perform when identifying verbs with similar meanings in the legal domain. The similarity measures that can be derived from these techniques can be used to determine how close the words are in the embedding spaces created by these word representation

techniques. The study [2] describes an approach that can be used to classify verb pairs as verbs with similar meanings or not, by using a threshold based on the similarity value of the two given word. However, suitable data sets (evaluation resources) are imperative to identify such a threshold based on semantic similarity to classify a given verb pair as similar or dissimilar. Though the resources and datasets that provide information related to semantic similarity between words in the legal vocabulary are limited, the importance of developing such publicly available resources is discussed in recent literature related to computational legal reasoning [21]. In relation to this research direction, a study by Sugathadasa et al. [22] describes how word embedding techniques such as Word2Vec and traditional lexicon based semantic similarity methods can be combined to develop a more reliable legal domain-specific semantic similarity measurement. Their approach has been utilized in several legal information tasks such as ontology population [23, 24], deriving representative vectors [25] and to retrieve similar documents [26].

Though there are evaluation resources such as SimLex-999 [27] and *SimVerb-3500* [28], in which the similarity between verbs are annotated, those resources have not considered the impact the surrounding context can have on a considered word or a verb. Moreover, the contextual information related to the verbs is not available. It can create issues in interpreting the sense of a verb. The lack of contextual information will also limit the evaluation of models that are pretrained using language modelling techniques such as BERT. A dataset that has been developed while considering the context (based on the sentences) when annotating the similarity of two words is provided in the study [29]. However, the dataset[29] consists of only 399 verb-verb pairs. All these datasets [27, 28, 29] are focused on providing a rating for word pairs based on their similarity, but not on classifying them as similar or dissimilar. Also as these datasets were not prepared focusing the legal domain, the use of these resources to analyze the behavior of word representation techniques in the legal domain might create drawbacks. In order to overcome these issues and limitations, in this work we have introduced *LeCoVe* , which is a context based verb similarity dataset prepared considering the legal

domain.

Chapter 3

Overall Methodology

3.1 Introduction

The aim of this Chapter is to describe the overall flow of the approach proposed in this thesis to develop sentiment annotators for the legal domain. While explaining the flow of the overall methodology, this chapter also explains how the works that is described in Chapter 4 facilitates the approach described in Chapter 5 to achieve the ultimate objective of developing legal domain specific sentiment annotators for the legal domain.

3.2 Overall Flow

As shown in Figure 3.1, the data that is needed to develop and evaluate the legal context sensitive verb similarity dataset (which is described in Chapter 4) as well to fine-tune and evaluate legal domain specific sentiment annotators were extracted from the Legal Opinion Text corpus that is available in the SigmaLaw dataset [22]. The main objective of this study is to develop sentiment annotators for the Legal domain with minimal use of resources using transfer learning. In that regard, we make use of the already available models and datasets related to sentiment analysis in the movie review domain as the source models and source datasets respectively. In the process, one of the key steps is to identify words that have a Legal domain specific sense or meaning. The notion of legal domain specific meaning can be elaborated in the following manner. If a sense or meaning of a considered word in the Legal domain is different from that of the source domain (Movie Review domain), such a word will be known as a word with a legal domain specific sense (domain specific word). Otherwise, the word will be known as a domain generic word. We have come up with an approach to distinguish domain specific words from domain generic words using domain specific word

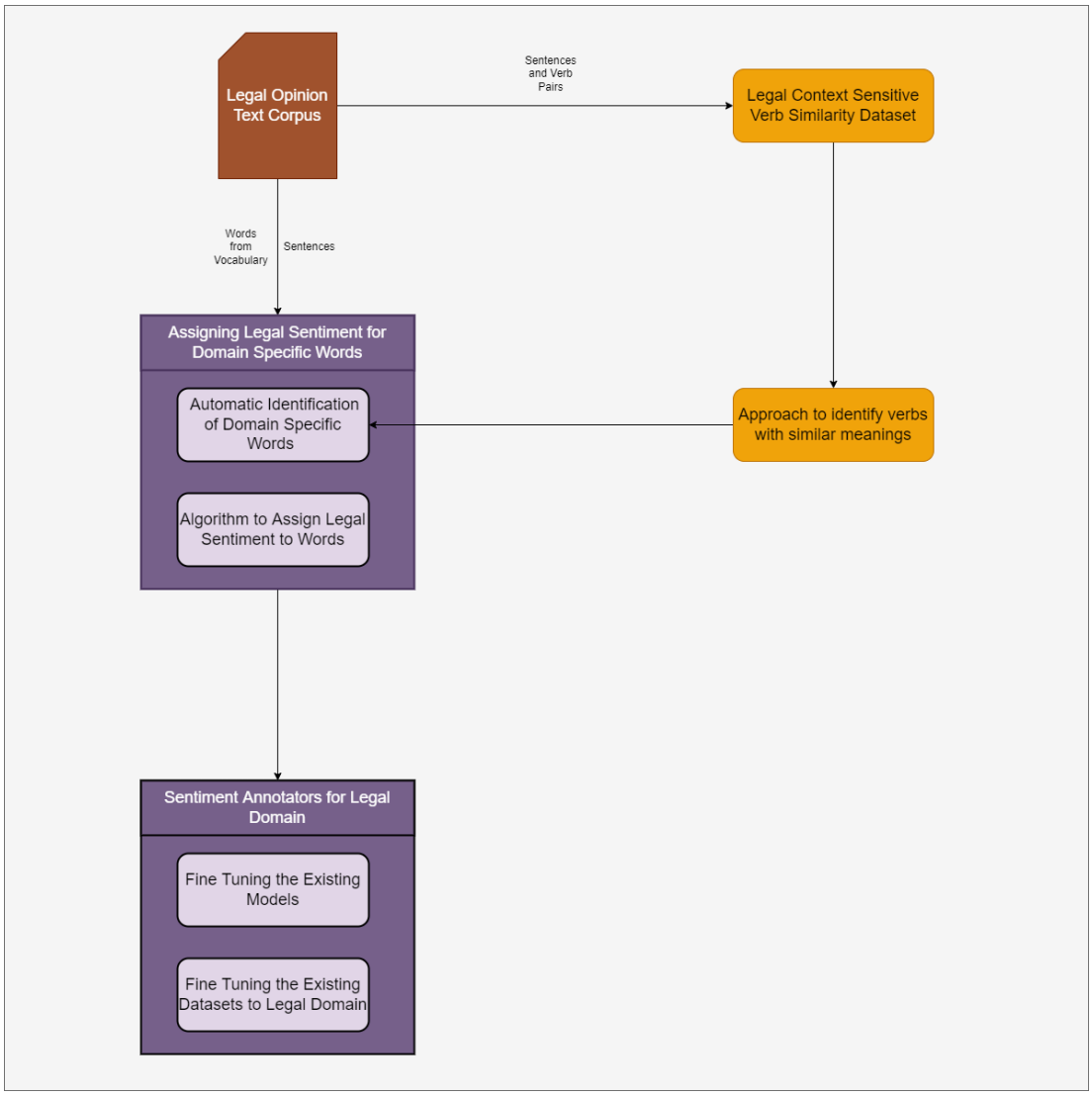

Figure 3.1: Overall Flow of the Project

representation models. The approach is described in a detailed manner in Chapter 5 and it can be briefly described as follows. For a given word w, we take the most similar word ($l(w)$) for w from a legal domain specific Word2Vec model. Similarly, the most similar word ($m(w)$) for w is taken from a movie review domain specific Word2Vec model. Then, the similarity value between $l(w)$ and $m(w)$ is derived from a legal domain specific word embedding model. Then, if this similarity value between $l(w)$ and $m(w)$ is greater than or equal to a particular threshold, w is considered as domain generic. Otherwise, the word w will be considered as domain specific. To determine this threshold value which will be used to distinguish domain specific words from domain generic words, we made use of the

observations that were derived during our attempt to automatically identify verbs with similar meanings using the legal context sensitive verb similarity dataset *LeCoVe* .

After identifying domain generic words and domain specific words, the legal domain specific sentiment of each word from the selected vocabulary is decided through an automated algorithmic approach that is developed within this study. The approach is described in a detailed manner in Chapter 5. After identifying the legal sentiments of words, we propose two mechanisms to develop legal sentiment annotators in a low resource setting. The first approach is a mechanism to adapt an existing model from the source domain (movie review domain) by changing the embeddings of words that have a different sentiment in the legal domain when compared with the sentiment in the movie review domain. The other approach is to modify the existing datasets related to sentiment analysis in movie review domain. Then, the modified dataset is used to train sentiment annotators for the legal domain. The methodologies related to these two approaches are described in a detailed manner in Chapter 5. Additionally, Chapter 5 also explains the experimental settings that were used to evaluate the proposed two approaches with the corresponding empirical results obtained after the experiments.

Chapter 4

Evaluating Word Representation Techniques Using Verb Similarity

4.1 Task Definition

In our proposed approach to minimize negative transfer when developing a sentiment annotator for the legal opinion domain (target domain) using datasets from another domain (source domain), identifying words that have different senses (meanings) across the two domains was an integral step. In that regard, we decided to make use of domain specific word embeddings and to evaluate the effectiveness of various word embedding techniques we have created a context sensitive verb similarity dataset for the legal domain.

To evaluate the effectiveness of different word embedding methods in identifying words with similar meanings, we focused on the task of identifying verbs with similar meanings in the legal domain. We choose verbs specifically because,

- Verbs are very important to understand meanings of sentences as they have a significant impact on the meaning due to their semantic and syntactic properties [30, 31, 28].
- The argument structure of verbs is pivotal for many legal domain related natural language processing tasks such as argument extraction [32], sentiment analysis[4] , discourse analysis and role labelling.
- Verbs are instrumental to understand the semantics of an event and how different parties are connected to a particular event [28](In legal opinion texts, much emphasis is given to the events/incidents related to the particular court case and involved parties).

4.2 Motivation

As described in Section 2.4, most of the existing evaluation resources including *SimVerb-3500* are focused on rating semantic similarity between two words, rather than explicitly rating whether two words in a word pair are having a similar meaning or not. Additionally, in most of the current evaluation resources, the context has not been considered when rating the similarity between verbs. However, the sense of a word may change based on the context. For example, consider the sentences given in Example 2.

Example 2

- Sentence 2.1: *Michael moved to United Kingdom.*
- Sentence 2.2: *Michael returned to Thailand.*
- Sentence 2.3: *Michael returned the balance to the customer.*

If we consider the verb *moved* in Sentence 2.1 with the verb *returned* in Sentence 2.2, the senses of both words are related to the mobility. But, the verb *returned* in Sentence 2.3 has a sense of giving back. This example demonstrate the impact of context on a meaning of a word.

Moreover, language modelling techniques such as XLNet [18], BERT [11] and ElMO [17] have surpassed traditional word embedding approaches (Word2Vec [14], Sense2Vec [16]) and have become the state of the art in several natural language processing tasks. However, in order to reap the maximum benefit from these language modelling techniques, it is important to take the context into the consideration when evaluating the similarity between two textual units.

Another important factor that determine the meaning of a word is the domain that is related to a document or a text. The verb *plea* suggests a behaviour of requesting in day to day context while in the legal domain, the same word often suggests a behavior of stating guilt or innocence. Therefore, it is important to prepare domain specific datasets to the legal domain in order to carry out comprehensive evaluations on behavior of words.

The context based verb similarity dataset *LeCoVe* was developed using legal

opinion texts related to United States criminal cases in order to overcome the above mentioned limitations in existing work (The dataset is publicly available at https://osf.io/bce9f/).

4.3 Dataset Preparation

The criminal court cases for the dataset were obtained by randomly picking criminal court cases from the publicly available legal opinion text corpus of the SigmaLaw dataset. Next, the sentences were extracted from the legal opinion text documents. Then, the sentences were split using Stanford CoreNLP [33]. The verb pairs were obtained from the sentence pairs (one verb from one sentence). When creating the sentence pairs, the sentences that are adjacent or only one sentence apart from each other in a legal opinion text were chosen. Such an approach was followed because it can be problematic to understand the context when the sentences are far away from each other. Given a sentence pair, the sentence that appears first in a legal opinion text is known as the *target sentence*. The other sentence in the same sentence pair is known as the *source sentence*.

Stanford CoreNLP PoS Tagger [34] was used to extract verbs from the sentences in a given sentence pair. Two lists were used to separately maintain the verbs from the source sentence and the target sentence. Verbs that are lemmatized into *be* or *have* were removed from the lists. Then the Wu-Palmer similarity scored [35] of each possible verb pair that can be formed by taking one verb from the target list and one verb from the source list was considered. Wu-Palmer similarity score between given two verbs is greater than 0.75, such a verb pair was added to the dataset. Otherwise, the verb pair was not included to the dataset. This step was taken as a measure of maintaining a proper balance between verb pairs with similar meanings and dissimilar meanings [2]. When a verb pair was chosen to be added to the dataset, the sentences that were used to extract the verb pair were also included to the same dataset. More precisely, these dataset contains information about *target sentence*, *source sentence*, *target verb*, *source verb*, *the lemmatized form of the target verb* and *the lemmatized form of the source*

verb.

4.4 Annotation of Verb Pairs

As the first step of the annotation process, all the human annotators were provided with a proper understanding of the two classes (Similar,Dissimilar) to which the verbs will be classified. A sets of examples that contains pre-identified data points related to each class were used to provide this understanding for the annotators. Next, the understanding of the annotators were further tested by discussing the thought process related to the annotation of randomly selected examples. Then, the human annotators were instructed to annotate each verb pair based on their similarity. More precisely, each verb pair was annotated either as a verb pair with similar meaning or as verb pair with dissimilar meaning. When providing the annotation, the annotators were instructed to interpret the meanings of the verbs while taking the context into the consideration using the corresponding sentences. The annotators were instructed to mark 1 for similar verb pairs and mark 0 for dissimilar verb pairs. Annotators were also instructed to give a score from 1 to 10 per each annotation, based on how confident they are on their annotation for the considered verb pair. The key statistics that have been identified after the annotation process is shown in Table 4.1 .

Table 4.1: Frequency Statistics of *LeCoVe*

Feature	Number of Verb Pairs
Two verbs with similar meaning (agreed by 3 human annotators)	170
Two verbs with similar meaning (agreed by atleast 2 human annotators)	285
Two verbs with similar meaning (agreed by atleast 1 human annotator)	463
Verb Pair with Same Lemmatized Form, but different meaning (considering majority agreement)	6
Verb Pair with Same Lemmatized Form and similar meaning	144
Number of unique verb pairs (lemmatized form)	714

4.5 Experiments and Evaluations

The annotation of verb pairs was performed by four human annotators. However, a given verb pair is annotated only by three human annotators. As a result, the annotators who annotate one pair may be different from the annotators of another pair. Therefore, the inter-rater reliability of the annotation process was measured using Fleiss' kappa [36]. A kappa value of 0.57 was observed. As interpreted in the study[37]), the kappa value of 0.57 falls into the range of the moderate agreement level.

As the next step, models created using different Word Representation Techniques were evaluated using The annotated dataset (*LeCoVe*). The evaluation was performed in order to get a proper understanding of the ability of these models to identify whether a given two verbs have a similar meaning in the legal domain or not. Such evaluations can also be used to get an idea of the effectiveness of the considered models in the legal domain.

4.5.1 Evaluation Resources

This section provides a detailed description of the models which have been evaluated using *LeCoVe* .

Word2Vec Models

We considered three word2vec models available in the SigmaLaw dataset[22] (SigmaLaw dataset can be found at https://osf.io/qvg8s/). These three models have been trained using a corpus of legal opinion text.

- Word2Vec (LR) - Trained using raw legal opinion text corpus.
- Word2Vec (LL) - Trained using lemmatized legal opinion text corpus.
- Word2Vec (LLR) - Trained using lemmatized legal opinion text corpus and then enhanced for lexical similarity.

The word2vec model which has been trained using Google news corpus by Google and is publicly available is also considered for the evaluations. From this point onwards, Google news word2vec model will be denoted by Word2Vec (G) .

Sense2Vec Models

Word2Vec provides only one representation for a given word. However, Sense2Vec provides multiple vector representations for a single word. In other words, the noun form and the verb form of a word will be provided with the same representation by Word2Vec. But, a Sense2Vec model will provide two different representations for the noun form and the verb form of a word. In order to train the Sense2Vec models, each word in the legal opinion text corpus available at SigmaLaw [22] was lemmatized. Then, the POS tags related to each of the lemmatized words were appended behind the each considered word.Spacy[1] was used to obtain the POS tag of the words. Using the modified corpus, three Sense2Vec models were trained[2]. Table 4.2 illustrates the key parameters that were used in the training of the Sense2Vec models.

Table 4.2: Sense2Vec Parameter Configurations

Parameter	SG-2	CBOW-10	SG-10
Model	Skip-gram	CBOW	Skip-gram
Size (Dimensionality)	128	128	128
Min. Count	5	10	10
Context Window Size	5	10	10
Training Algoriithm	Negative Sampling	Hierarchical Softmax	Negative Sampling
Number of Iterations	2	10	10

Moreover, the publicly available Reddit Vectors 1.1.0 Sense2Vec model was considered in our experiments. From this point onwards, Reddit Vectors 1.1.0 model is also denoted as Sense2Vec(R).

[1]https://spacy.io/

[2]The Sense2Vec and BERT models developed in this study are available at https://osf.io/s8dj6/

BERT

BERT [11] is a popular language modeling technique that is being used for many NLP tasks. Unlike the word embeddings provided by Word2Vec/Sense2Vec, the representation that BERT provides for a word can vary based on the surrounding context of the word. The publicly available pre-trained BERT model ('bert-base-uncased') was used in our experiments. Moreover, we made use of the implementation mechanisms provided by Transformers [3] library.

The pretrained BERT model ('bert-base-uncased') which is publicly available was trained using a very large Wikipedia corpus and a book corpus. In order to post train the BERT model, a corpus was created using the Criminal Court Cases available at the SigmaLaw dataset. Following the instructions for BERT training as provided in BERT implementation repository by Google Research [4], the legal corpus was modified to suit the post training of the BERT model. From this point, the BERT implementation by Google Research will be denoted as BERT(G). Only the sentences with more than 4 tokens were considered for the training corpus. This step was followed to overcome the issues that can be occurred when splitting the sentences. The text data set prepared for post training of BERT consists of 90851 sentences. Then, the prepared text dataset was used to post train the 'bert-base-uncased' model. In the training phase, BERT is designed to learn two tasks. The first task is masked language modelling. In masked language modelling, the task is to predict the tokens which are masked. The second task is *next sentence prediction*. In the post training phase of the 'bert-base-uncased' model using the legal text data, the performances of the post trained model after one training iteration and 500 training iterations were observed. The observations are included in the Table 4.3. The observations demonstrate that the accuracy of the BERT model for the legal text data is low after the first iteration. However, there is a significant improvement in accuracy when the model is further trained for 500 training iterations. From this point onwards, the BERT model which has been post trained using legal text data will be denoted by BERT(L).

[3]https://github.com/huggingface/transformers
[4]https://github.com/google-research/bert

Table 4.3: Post training of BERT using criminal court case corpus

No. of Training Steps	Masked LM Accuracy	Masked LM Loss	Next Sentence Accuracy	Next Sentence Loss
1	0.55	2.71	0.60	2.47
500	0.70	1.42	0.95	0.14

4.5.2 Evaluation of the distributional word representation models

The steps related to the evaluations of models that are based on Word2Vec or Sense2Vec are shown below.

- The cosine similarity of the two vectors (vector representations/ embeddings of the two verbs in the considered verb pair) is calculated.
- Given a verb pair, let **U** be the embedding of the *Source Verb* and **V** be the embedding of the *Target Verb*. The cosine similarity between **U** and **V** will be in the range of -1 to 1.
- Linearly scale the cosine similarity between **U** and **V** to be in the range of 0 to 1. Let *sv* be the value obtained after scaling. Then, $sv = \frac{U^T V+1}{2}$). The *sv* value was considered as the similarity score between the two verbs corresponding to **U** and **V** .
- After obtaining the similarity score between two verbs using a word embedding model, it is checked whether the similarity score is greater than or equal to a predefined threshold value. If the similarity score is greater than or equal to the considered threshold value, it is considered that the verb pair is classified (by the considered model) as having two verbs with a similar meaning. Otherwise, the verb pair is considered to be classified as having two verbs with dissimilar meanings.
- When evaluating Word2Vec models or Sense2Vec models that were trained using a lemmatized legal opinion text corpus, the lemmatized forms of the verbs were considered.
- Then the classifications obtained using each model were compared with the ground truth, which is the human annotations. In *LeCoVe* , each

Table 4.4: Recall (R) and F-Score (F) received for different thresholds of considered Word2Vec/Sense2Vec models

ModelThreshold	0.60		0.65		0.70		0.75		0.80		0.85		0.90	
	R	F	R	F	R	F	R	F	R	F	R	F	R	F
Word2Vec(G)	0.85	0.62	0.75	0.70	0.64	**0.71**	0.52	0.66	0.45	0.60	0.33	0.49	0.29	0.45
Word2Vec(LR)	0.80	0.65	0.74	0.70	0.68	**0.72**	0.60	0.70	0.53	0.66	0.41	0.57	0.33	0.49
Word2Vec(LL)	0.80	0.51	0.72	0.57	0.62	0.62	0.55	0.66	0.52	**0.67**	0.51	0.67	0.51	0.66
Word2Vec(LLR)	0.79	0.65	0.74	0.70	0.66	0.72	0.62	**0.73**	0.60	0.72	0.55	0.70	0.54	0.69
Sense2Vec(R)	1.00	0.46	1.00	0.47	0.98	0.50	0.86	0.54	0.71	0.62	0.58	0.64	0.52	**0.67**
Sense2Vec(SG-2)	0.92	0.55	0.83	0.58	0.78	0.64	0.70	0.67	0.65	0.71	0.62	**0.72**	0.56	0.70
Sense2Vec(CBOW-10)	0.94	0.53	0.89	0.63	0.81	0.69	0.68	0.69	0.64	0.72	0.61	0.72	0.59	**0.72**
Sense2Vec(SG-10)	0.82	0.62	0.75	0.66	0.71	0.69	0.65	**0.74**	0.61	0.73	0.56	0.70	0.55	0.69

pair of verbs was annotated by three human annotators. The class (Similar/Dissimilar) of a verb pair is determined based on the majority agreement, i.e. the class agreed by at least two human annotators.

- Each of the considered models was evaluated while varying the value of the pre-defined threshold that is used to classify a verb pair as similar or dissimilar.

- The evaluations were carried out in relation to the *Similar* class because the intention of this experiment was to evaluate the capability of the considered models to identify the verb pairs with similar meanings.

Table 4.4 provides the results obtained from the evaluations. Precision and Recall values were calculated as follows. Let C be the number of verb pairs classified by the system as having verbs with similar meaning and let D be the number of verb pairs classified as having verbs with similar meaning according to human annotations. Then,

$$Precision = \frac{C \wedge D}{C} \tag{4.1}$$

$$Recall = \frac{C \wedge D}{D} \tag{4.2}$$

4.5.3 Deriving Embeddings for Words using BERT

The representation that can be obtained by language modelling techniques like BERT are also called as Contextual Word Embeddings as the provided representations consider the context around the considered word. As a result, unlike the traditional word embedding models like Word2Vec/Sense2Vec (statice vector representation), the representation provided by BERT for the same word can change based on the context (dynamic vector representation). Also, unlike when using Word2Vec/Sense2Vec, when we try to obtain the representation of a verb using BERT, it is needed to provide the sentence (context) where the verb resides to obtain the embedding of that verb.

For our experiments with BERT, we use the 'bert-based uncased' model. It consists of 12 hidden layers. In a single layer, each token in a sentence is represented by 768 hidden units. Promising results have been shown in previous studies when contextual word embedding is obtained by averaging the representations provided for the considered word by the last 4 hidden layers. Therefore, the same methodology was followed in this work to obtain the contextual word embeddings. However, due to the tokenization mechanism of BERT, contextual embeddings for some words could not be obtained. *Substantiate* can be considered as one of those verbs where the tokenization causes issues to obtain the embedding.*Substantiate* is tokenized into sub-tokens *sub*, *##stan*,*##tia*,*##te*. To address this issue, we first identify each sub token of a word and get the corresponding vector representation (embedding) for each subtoken. Then, we take the mean of the subtoken embeddings as the contextual vector representation for the word. However, there can be situations where a subtoken of a word can be the lemma of that particular word. In such cases, the contextual embedding of the subtoken is directly taken as the contextual embedding of the verb. The results obtained following this subtoken embedding based approach are shown under BERT(G) Improved and BERT(L) Improved models in Table 5.2. Following the above mentioned approach, the contextual embeddings for each verb in the verb pairs of *LeCoVe* were obtained from both BERT models, BERT(G) and

Table 4.5: Recall (R) and F-Score(F) received for different thresholds of BERT based approaches

ModelThreshold	0.50		0.525		0.55		0.60		0.65		0.70		0.75	
	R	F	R	F	R	F	R	F	R	F	R	F	R	F
BERT(G)	0.80	0.65	0.77	0.69	0.72	0.68	0.65	**0.71**	0.57	0.68	0.45	0.60	0.34	0.50
BERT(G) Improved	0.85	0.66	0.81	0.70	0.75	0.69	0.67	**0.72**	0.59	0.69	0.46	0.61	0.38	0.54
BERT(L)	0.72	**0.71**	0.69	**0.71**	0.65	0.70	0.58	0.69	0.50	0.64	0.50	0.64	0.38	0.54
BERT(L) Improved	0.75	0.72	0.72	**0.73**	0.66	0.71	0.60	0.70	0.51	0.65	0.39	0.54	0.27	0.43

BERT(L). Then, both the models were evaluated considering the cosine similarity of the verb embeddings following a similar approach as described in 4.5.2. However, unlike the approach described in 4.5.2, we considered cosine similarity values as it is (without performing linear scaling on the cosine similarity values). The results obtained from the experiments are shown in Table 5.2.

4.5.4 Evaluating models based on most similar words

As another way of evaluating Word2Vec/Sense2Vec models on their usefulness in identifying verbs with similar meanings, the topmost words predicted as most similar for given verb can be considered. When a verb pair is considered it consists of two verbs; source verb (which is taken from the source sentence) and target verb (which is taken from the target sentence). When evaluating the considered moded using this approach, following aspects related to source verb and target verb is considered in this approach.

- *most similar words source* : List of first k words predicted as the most similar words to the lemmatized form of the source verb.
- most similar words target : List of first k words predicted as the most similar words to the lemmatized form of the target verb.
- Condition 1 : Lemmatized form of the target verb is in *most similar words source*.
- Condition 2 : Lemmatized form of the source verb is in *most similar words target*.

If Condition 1 or Condition 2 is true, the two verbs (source verb and target verb) will be classified as having a similar meaning. Furthermore, if the lemmatized form of two verbs are exactly the same, such verb pairs were also considered as having a similar meaning. The approach described here was evaluated using Word2Vec and Sense2Vec models in relation to different k values as shown in Table 4.6. Similar to the Section

Table 4.6: Precision(P), Recall (R) and F-Measure (F) received by considering k most similar words predicted by models

	k=5			k=10			k=15			k=20		
	P	R	F	P	R	F	P	R	F	P	R	F
Word2Vec (LR)	0.83	0.60	0.69	0.78	0.62	0.69	0.77	0.67	0.71	0.73	0.68	0.70
Word2Vec (LL)	0.85	0.55	0.67	0.76	0.60	0.67	0.71	0.61	0.66	0.67	0.61	0.63
Word2Vec (LLR)	0.87	0.63	0.73	0.83	0.66	0.74	0.76	0.67	0.71	0.75	0.68	0.72
Sense2Vec (SG-2)	0.93	0.59	0.73	0.88	0.61	0.72	0.84	0.64	0.73	0.84	0.64	0.73
Sense2Vec (CBOW-10)	0.82	0.63	0.71	0.75	0.65	0.70	0.73	0.68	0.70	0.71	0.69	0.70
Sense2Vec (SG-10)	0.92	0.61	0.74	0.87	0.64	0.73	0.85	0.64	0.73	0.82	0.65	0.73

4.5.2, equation 4.1 and equation 4.2 have been used to measure precision and recall respectively, where C is the number of verb pairs classified by this model as having verbs with similar meaning and D is the number verb pairs classified as having verbs with similar meaning according to human annotations.

4.5.5 Evaluating BERT models based on most similar words

In BERT, the model is trained to correctly predict the tokens in a token sequence, which have been masked or corrupted. This training mechanism itself can be exploited to identify verbs which convey similar meaning. However, when following this approach, it is necessary to consider the sentence which is used to extract the considered verb. Let *moved* and *returned* in Example 3 be the verb pair which is needed to be classified either as two verbs with similar meaning or two verbs with dissimilar meaning. Here, Sentence 3.1 is the *Target Sentence* and Sentence 3.2 is the *Source Sentence*, making *moved* the *Target Verb* and *returned* the *Source Verb*. A BERT based approach can be used to classify the verb pair using the following procedure. First, the verb *moved* in Sentence 3.1 is replaced with token [MASK], thus corrupting the *Target Sentence*. Then, the corrupted sentence can be input into the pretrained BERT model[5], and the first k tokens that are predicted by the model as the tokens which should replace the token which is occupied with value [MASK] can be considered. If *returned* (which is the other verb in the verb pair) is in those first k predictions, it can be considered that *returned* is also having a significance similarity to *moved* in the considered context. The same procedure can be followed with the other sentence as well. Here, a suitable value for k has to be decided empirically. Our experiments considering different k values has been provided in Table 3. For each value of k, we have considered four approaches when

[5]for our experiments we have used https://github.com/huggingface/pytorch-transformers

Table 4.7: Precision (P), Recall (R) and F-Measure (F) received from different approaches based on BERT

[40mm]Approachk value	k=10			k=25			k=50		
	P	R	F	P	R	F	P	R	F
C1 AND C2	0.90	0.10	0.18	0.83	0.18	0.29	0.82	0.24	0.37
C1 OR C2	0.68	0.29	0.41	0.62	0.39	0.48	0.53	0.48	0.50
(C1 OR C3) AND (C2 OR C4)	0.88	0.10	0.18	0.80	0.18	0.30	0.79	0.27	0.40
(C1 OR C3) OR (C2 OR C4)	0.63	0.41	0.50	0.59	0.55	0.57	0.49	0.65	0.55

determining whether a verb pair is similar or not. These approaches are developed considering different conditions. First condition (C1); source verb is in the first k predictions when corrupted target sentence is input to the model. Second condition (C2); target verb is in the first k predictions when corrupted source sentence is input to the model. Third condition (C3); lemma of the source verb is in the first k predictions of corrupted target sentence. Fourth condition (C4); lemma of the target verb is in the first k predictions of corrupted source sentence. We have evaluated different approaches based on these four conditions as shown in Table 3. In the table, *C1 AND C2* suggest that if both C1 and C2 are satisfied, then the verb pair is considered to be classified by the system as having two verbs with similar meaning. Other approaches mentioned in the table can also be interpreted in the same manner. Precision and Recall for each k value were also calculated as described in equations 4.1 and 4.2.

4.5.6 Analysis of Results

When we consider the results depicted in Table 4.4 related to the Word2Vec and Sense2Vec word embedding techniques, the following observations can be derived.

- Word2Vec/Sense2Vec models that were trained using the legal corpus (domain-specific and relatively small corpus) tend to outperform the Word2Vec/Sense2Vec models that were trained using large corpora which are either domain generic or belongs to another domain. In other words, the legal opinion text corpus based Word2Vec (LLR) and Word2Vec(R) models outperform the Word2Vec(G) model that was trained using a large Google News Corpus. Similarly, Sense2Vec(SG-2), Sense2Vec(CBOW-10), and Sense2Vec(SG-10) models that were trained from the legal opinion text corpus perform better than Sense2Vec(R), the sense2vec model that has been trained using a corpus from Reddit.

- Based on the above mentioned results, it can be observed that the word embedding models that are trained using domain specific corpus tend to perform better in a considered domain than the word embedding models that are trained using a domain generic corpus.

- Word2Vec(LLR) model performs better than all other Word2Vec models, agreeing with the observations of [22].

- Sense2Vec(SG-10) model's overall performance is better than that of the other Sense2Vec models. This result shows when the dimensionality of word representation vectors and the number of iterations used for training the models are the same, Sense2Vec models that are trained using skip-gram approach tend to overperform the CBOW (Continuous Bag of Words) approach.

- The performance of Sense2Vec(SG-10) is on par with Word2Vec(LLR) model. Lexical resources have been used to enhance the reliability of the lexical similarity that can be obtained from Word2Vec(LLR) [22]. However, such enhancements were not performed in Sense2Vec(SG-10).

- As shown in Table 4.4, for different threshold values, different models show the best F-Scores. Sense2Vec(CBOW-10) model has the highest F-Scores when the threshold is above 0.8. However, Sense2Vec(SG-10) and Word2Vec(LLR) outperform other models when the threshold values are between 0.4 and 0.8. This can be considered as a good indication that it is possible to use techniques such as ensemble modelling where several models are combined together to achieve better performances.

When interpreting the results related to BERT based approaches as shown in Table 4.5, the following observations can be made.

- For all four BERT based approaches, their highest F-Scores are in the range of 0.71-0.73. These results are comparable with the results obtained using Word2Vec/Sense2Vec models.

- The results also demonstrate that the improvements introduced in this study for deriving contextual word embeddins by considering the embeddings of subtokens

have enhanced the performances of both BERT(G) model as well as BERT(L) model.

- The BERT(L) Improved approach, where the BERT model was post trained using a legal domain corpus and then further improved by considering the embeddings of subtokens when obtaining the representation for a considered word has outperformed all other BERT based approaches in identifying verbs which has similar meanings in legal opinion texts.

4.6 Discussion

In summary, this chapter described the information related to the preparation of a legal context based verb similarity dataset *LeCoVe* and the motives behind developing *LeCoVe* . *LeCoVe* has been made publicly available to the research community. As described in this chapter, we have also evaluated the performances of several word representation techniques and language modelling techniques in the legal domain using *LeCoVe* . In addition to evaluating existing Word2Vec models developed for the legal domain, we have also developed new Sense2Vec models focusing on the legal domain for the evaluations. Additionally, we also post-trained 'bert-base-uncased' BERT model using a legal opinion text corpus and using BERT models, we demonstrated that *LeCoVe* can be used to unleash the contextual word representations provided by the language modelling techniques such as BERT. In other words, as *LeCoVe* provides the context associated with each verb rather than just providing only the verb, *LeCoVe* will enable the evaluation of language representation models that consider the sequential context associated with a text.

Chapter 5

Developing a Legal Sentiment Annotator in a Low Resource Setting

5.1 Task Definition

In this chapter, it is explained how a sentiment analysis dataset from another domain (source domain) can be effectively utilized to develop a sentiment annotator for the legal domain (target domain) while minimizing the negative transfer.

5.2 Methodology

5.2.1 Detecting words that can cause negative transfer

Minimizing resource requirements to develop a reliable sentiment annotator for the legal domain can be considered as our main objective. To that regard, our intention is to utilize resources from a source domain (a domain which has adequate amount of labeled resources) to perform sentiment analysis in the legal domain (a low resource domain/target domain). As the source dataset, Stanford Sentiment Treebank (SST-5) [10] was considered. It consists of Rotten Tomato movie reviews annotated according to their sentiments. The words available in the source dataset can be assigned into three main categories as shown below.

- *Domain Generic words* - Words that behave in a similar manner in both the domain (the movie review domain and the legal domain).
- *Domain Specific words* - Words that behaves differently in the two domains. In other words, the most frequently used sense of a word in source domain may be different from that of the target domain. Thus, these words have the potential to cause negative transfer. A word belongs to this set may have different sentiment polarities across the two domains.
- *Under Represented Words* - The words that occur frequently in the target dataset

(legal domain), but occur very less frequently or not available in the source dataset.

Manual identification of *domain specific words*, *domain generic words*, and *under represented words* in a dataset by going through each word that is available in the legal opinion text corpus is not feasible because of the limited human resources. The sequence of steps that were followed in order to minimize the manual annotations is described below.

- The stop words in the considered legal opinion text corpus were removed. The Van stop list [38] was used to identify stop words.
- *Word frequency* was calculated for each word in the legal opinion text corpus. Word frequency is the frequency of occurrence of a particular word within the corpus.
- Then, the words were sorted in the descending order based on their word frequencies in order to obtain the sorted set D. Let $k = \min_j\{j \in Z^+ | \sum_{i=1}^{j}(w_i) \geq 0.95 \cdot \sum_{i=1}^{n}(w_i)\}$ given that w_i is the i^{th} element of D and n is the cardinality of D. Then, the first k words of D were then chosen as the set of words S that will be considered to identify the words with negative transfer.

Next, the sentiment of each word in S was annotated using the Stanford Sentiment Annotator ($RNTN_m$). Based on the annotated sentiment by the Stanford Sentiment Annotator, the words were distributed into three sets P_M, N_M, O_M.

- P_M - The set of words that were annotated as Very Positive or Positive. The number of words in $P_M(|P_M|)$ is equal to 336.
- N_M - The set of words that were annotated as Very Negative or Negative. $|N_M|$= 253.
- O_M - The words that were annotated as having a Neutral sentiment. $|O_M|$ = 4992.

As $|O_M| = 4992$, it is difficult to manually identify words in O_M that have different sentiments across the two domains. A heuristic approach to identify words in O_M that have deviated sentiment across the domains was developed to overcome this challenge.

Moreover, in our algorithmic approach as described by Algorithm 1, Algorithm 2, and Algorithm 3, words with deviated sentiments are identified while automatically assigning each word with a legal sentiment. Note that Algorithm 1, Algorithm 2, and Algorithm 3 are 3 parts of the same algorithm.

Though it is feasible to manually annotate all the words in P_M and N_M, we have developed our algorithmic approach to automatically identify words that can have deviated sentiments in P_M and N_M as well (Algorithm 3). Such an automated heuristic approach is useful because it can be used to minimize the number of required manual annotations. Moreover, such approaches will be useful to the automatic generation of domain specific sentiment lexicons with minimal human intervention.

We have derived the following two key information from word embedding models in order to facilitate the method we have developed to distinguish domain specific words and domain generic words separately.

- $Cosine_{domain}(u, v)$ - Cosine similarity between the embeddings of two words u and v.
- $mostSimilar_{domain}(w)$- The most similar word for a particular word w as given by the considered word embedding model.

Domain specific word embeddings have been utilized within our approach to identify domain specific words from domain generic words. The Word2Vec model publicly available at SigmaLaw dataset [22] that has been trained using a United States legal opinion text corpus was selected as the legal domain specific word embedding model. The SST-5 dataset does not contain an adequate amount of text data to be used as a corpus to create an effective word embedding model. Therefore, we selected the IMDB movie review corpus [39] to train the movie review domain specific Word2Vec embedding model. From this point onwards, the following notations will be used:

- $Cosine_{legal}$ will be denoted by $Cosine_l$
- $Cosine_{movie-reviews}$ will be denoted by $Cosine_m$
- $mostSimilar_{legal}(w)$ will be denoted by $l(w)$
- $mostSimilar_{movie-reviews}(w)$ will be denoted by $m(w)$

First, for a given word w, we obtain $l(w)$ and $m(w)$. As Word2Vec [40] embeddings are based on distributional similarity, it can be assumed that the most similar word output by a domain specific embedding model to a particular word is related to the domain specific sense of that considered word. For example, *convicted* is obtained as $l(charged)$. It can be observed that the word convicted is associated with the sense of accusation, which is the most frequent sense of *charge* in the legal domain. However, when it comes to $m(charged)$, *sympathizing* is obtained as the output. *Sympathizing* is associated with the sense of *filled with excitement or emotion*, which is the most frequent sense of *charged* in the movie reviews. After obtaining the most similar words for a given word w, we define a value $domainSimilarity(w)$ such that $domainSimilarity(w) = Cosine_l(l(w), m(w))$. As we are considering the legal embedding model when getting the cosine similarity values, a higher *domainSimilarity(w)* value will suggest that legal sense and movie sense of the word w have a similar meaning in the legal domain while a lower *domainSimilarity(w)* will suggest that the meanings of the two senses are less similar to each other. For example, the value obtained for *domainSimilarity(Charged)* was 0.06 while it was 0.53 for *domainSimilarity(Convicted)* (*convicted* has a similar sense across the two domains).

The next step is to identify a threshold based on domainSimilarity(w) to heuristically distinguish whether a word w is domain generic or not. To that regard, we made use of *LeCoVe* . As described in Chapter 4, our approach to identify verbs with similar meaning can be briefly described as follows. First, a threshold t based on cosine similarity was defined. For a given two verbs v_i, v_j, if $Cosine_l(v_i, v_j) \geq t$, the two verbs are considered as having a similar meaning. We identified that the same approach can be used to identify whether l(w) and m(w) have the similar meaning. But it was needed to identify a suitable word representation model and a cosine similarity value as the threshold. Based on the results that were obtained after the experiments (given in Chapter 4), Word2Vec (LLR) model was selected because it supports non lemmatized tokens and also it has outperform or on par with other models (which support non lemmatized tokens) when it comes to capturing legal sense of words. The cosine similarity values of 0.2 was selected as the threshold value to identify domain generic words based on the *domainSimilarity(w)* score (Table 4.4 shows the similarity values after linearly scaling the cosine similarity values to [0,1] range. Therefore, the similarity value of 0.6 in Table 4.4 is corresponding to cosine similarity value of 0.2. From Table 4.4, it can

be seen that that precision is greater than 0.5 when the threshold similarity value is equal to 0.6 (cosine similarity value of 0.2). It was found that the threshold value drops below 0.5 when the similarity value is equal to 0.55 (cosine similarity value of 0.1)). In other words, if *domainSimilarity(w)* is greater than or equal to 0.2, the word w will be considered as domain generic and the attribute *domainGeneric(w)* will be set to true. Otherwise, the attribute *domainSpecific(w)* will be set to true. Though we have used the aforementioned approach to determine the threshold, it is a heuristic and domain specific value that can be decided based on different experimental techniques (when applying this methodology to another domain).

Even if a word behaves in a similar manner across the two domain, it still can be assigned with a wrong sentiment (neutral sentiment) due to under representation. However, it is important to identify words with sentiment polarities (positive or negative) as the descriptions with positive or negative sentiments tend to contain more specific information that will be useful in legal analysis. As a measure of identifying sentiment polarities of under represented words, we made use of AFINN [9] sentiment lexicon (denoted as set A from this point onwards), which consists of 3352 words annotated based on their sentiment polarity (positive, neutral, negative) and sentiment strength considering the domain of twitter discussions. If a frequency of a word w is less than 3 in the source dataset, *underRepresented(w)* is set to true. Assignment of AFINN sentiment for an under represented word or a domain specific word w can create a positive impact if the most frequently used sense of w in twitter discusion domain is aligned towards it's sense in the legal domain than the sense of that word (w) in the movie review domain. In order to heuristically determine this factor, we have defined an attribute name *afinnSimilarity* such that $afinnSimilarity(w) = Cosine_t(w, l(w)) - Cosine_t(w, m(w))$, where w is a given word and $Cosine_t$ is the cosine similarity obtained using a publicly available Word2Vec model [41] trained using tweets. If $Cosine_t(w, l(w)) > Cosine_t(w, m(w))$, it can be assumed that the sense of word w in twitter discussions is more closer to its sense in the legal domain than that of the movie-reviews. Thus, if *afinnSimilarity(w)* > 0 and $w \in A$, the attribute *afinnAssignable(w)* is set to true.

Both Algorithm 1, Algorithm 2, and Algorithm 3 are three parts of one major algorithmic approach denoted seperately for readability. Therefore, the functions and attributes defined in Algorithm 1 are applied globally for both Algorithm 2 and Algorithm 3 as well. The states of the attributes after executing Algorithm 1 will be

Algorithm 1 Functions

procedure $assignSentiment_o(w, sentiment)(x)$
    if sentiment == N **then** $D_{on} \cup \{w\}, O_i - \{w\}$
    else if sentiment == P **then** $D_{op} \cup \{w\}, O_i - \{w\}$
    end if
end procedure

procedure $assignSentiment_n(w, sentiment)(x)$
    if sentiment == N **then** $D_{nn} \cup \{w\}$
    else if sentiment == P **then** $D_{np} \cup \{w\}, N_i - \{w\}$
    else if sentiment == O **then** $D_{no} \cup \{w\}, N_i - \{w\}$
    end if
end procedure

procedure $assignSentiment_p(w, sentiment)(x)$
    if sentiment == N **then** $D_{pn} \cup \{w\}, P_i - \{w\}$
    else if sentiment == P **then** $D_{pp} \cup \{w\}$
    else if sentiment == O **then** $D_{po} \cup \{w\}, P_i - \{w\}$
    end if
end procedure

Algorithm 2 Assigning the Legal Sentiment

$P_i = P_m, N_i = N_m, O_i = O_m,\ D_{on} = \{\}, D_{op} = \{\}$
$D_{nn}, D_{np}, D_{no}, D_{pp}, D_{pn}, D_{po} = \{\}$
n=0,p=0
while $1 + |D_{on}| > n$ or $1 + |D_{op}| > p$ **do**
    n=$1 + |D_{on}|$, p =$1 + |D_{op}|$
    for each word w in O_i **do**
        $l = mostSimilar_l(w)$
        if underRepresented(w) and affinAssignable(w) **then** $assignSentiment_o(w, afinn(w))$
        else if domainSpecific(w) and affinAssignable(w) **then**
            $assignSentiment_o(w, afinn(w))$
        else if domainGeneric(l) and $l \in N_m \cup D_{on}$ **then**
            if notAntonym(w,l) **then** $assignSentiment_o(w, N)$
            end if
        else if domainGeneric(l) and $l \in P_m \cup D_{op}$ **then**
            if notAntonym(w,l) **then** $assignSentiment_o(w, P)$
            end if
        end if
    end for
end while

Algorithm 3 Assigning the Legal Sentiment

n=0,p=0
while $1 + |D_{nn}| > n$ or $1 + |D_{np}| > p$ **do**
    n=$1 + |D_{nn}|$, p =$1 + |D_{np}|$
    Q = $N_i \cup D_{on} \cup D_{nn}$, R = $P_m \cup D_{op} \cup D_{np}$
    for each word w in N_i **do**
        $l = mostSimilar_l(w)$
        if domainGeneric(w) **then** $assignSentiment_n(w, N)$
        else if domainSpecific(w) and affin(w)==N **then**
            $assignSentiment_n(w, N)$
        else if domainSpecific(w) and notAntonym(w,l) **then**
            if $l \in Q$ **then** $assignSentiment_n(w, N)$
            else if domainGeneric(l) and $l \in R$ **then** $assignSentiment_n(w, P)$
            end if
        end if
    end for
end while
for each word w in N_i **do**
    $assignSentiment_n(w, O)$
end for
n=0,p=0
while $1 + |D_{pp}| > p$ or $1 + |D_{pn}| > n$ **do**
    p=$1 + |D_{pp}|$, n =$1 + |D_{pn}|$
    Q = $N_m \cup D_{on} \cup D_{pn}$, R = $P_i \cup D_{op} \cup D_{pp}$
    for each word w in P_i **do**
        $l = mostSimilar_l(w)$
        if domainGeneric(w) **then** $assignSentiment_p(w, P)$
        else if domainSpecific(w) and affin(w)==P **then**
            $assignSentiment_p(w, P)$
        else if domainSpecific(w) and notAntonym(w,l) **then**
            if $l \in R$ **then** $assignSentiment_p(w, P)$
            else if domainGeneric(l) and $l \in Q$ **then** $assignSentiment_p(w, N))$
            end if
        end if
    end for
end while
for each word w in P_i **do**
    $assignSentiment_p(w, O)$
end for$P_l = D_{op} \cup Dnp \cup Dpp$, $N_l = D_{on} \cup Dnn \cup Dpn$

transferred to the Algorithm 2 and Algorithm 3. Similarly, the states of the attributes after executing Algorithm 2 will be transferred to the Algorithm 3. In the algorithms, P, N, O denotes positive, negative, and neutral sentiments respectively. *afinn(w)* is the AFINN sentiment categorization of a given word w. When observing the algorithm, it can be observed that sentiment of $l(w)$) is also considered when determining the correct sentiments of a word. For a word in O_m, the sentiment of $l(w)$ will be assigned if $l(w)$ is *domain generic* (Algorithm 2). This step was followed as another way to identify words with sentiment polarities (positive or negative). The sentiments of domain generic words in P_m or N_m will not be changed under any condition. For a domain specific word w in P_m or N_m, if $l(w)$ has a opposite sentiment polarity to that of w, the sentiment of $l(w)$ will be assigned to w only if $l(w)$ is domain generic. All the *domain specific* words in P_m or N_m that do not satisfy any of the conditions that are required to assign a positive or negative polarity (Algorithm 3), will be assigned with a neutral sentiment. This step is taken because such *domain specific* words have a relatively higher probability to have opposite sentiment polarities in the legal domain, thus capable of transferring wrong information to the classification models [5]. Assigning neutral sentiment will reduce the impact of negative transfer that can be caused by such words (neutral sentiment is better than having the opposite sentiment polarity). Furthermore, it should be noted that an antonym of a particular word w can be given as $l(w)$ by the embedding model due to semantic drift. To tackle this challenge, WordNet [42] was used to check whether a given word w and $l(w)$ are antonyms. If they are not antonyms, notAntonyms() attribute is set true. After running the Algorithm 1, Algorithm 2, and Algorithm 3 by taking P_m, O_m, N_m as the inputs, the word sets D_{on}, D_{op} were obtained that consist of words the overall algorithm picked from O_m as having negative and positive sentiments respectively. D_{on}, D_{op} together with P_m, N_m were given to a legal expert in order to annotate the words in these sets based on their sentiments. $|D_{on}| = 220$ and $|D_{op}|$=116, thus reducing the required amount of annotations to 925 (925= $|W|$, where $W = D_{op} \cup D_{on} \cup P_m \cup N_m$). After the annotation process, three word sets N_a, O_a, P_a were obtained that contains words that are annotated as having positive, neutral and negative sentiments respectively. Then word sets D_n, D_o, D_p were created such that $D_n = \{w \in W | w \in N_a \& w \notin N_m\}$, $D_p = \{w \in W | w \in P_a \& w \notin P_m\}$, $D_o = \{w \in W | w \in O_a \& w \notin O_m\}$. P_l contains the set of words identified by the overall algorithm as having positive sentiment and N_l contains the words identified as having

negative sentiment (without human intervention).

5.2.2 Fine Tuning the Recursive Tensor Neural Network Model

As an approach to develop a sentiment classifier for legal opinion texts, $RNTN_m$ (Stanford Sentiment Annotator) [10] was fine tuned following a similar methodology as proposed by [4]. In the proposed methodology [4], there is no need to further train the $RNTN_m$ model or to modify the neural tensor layer of the model. Instead, the approach is purely based on replacing the word vectors. In this approach, if a word v in a word sequence S have a deviated sentiment s_d in the legal domain when compared with its sentiment s_m as output by the $RNTN_m$, the vector corresponding to v will be replaced by the vector of word u, where u is a word from a list of predefined words that has the sentiment s_d as output by $RNTN_m$. When choosing u from the list of predefined words, PoS tag of w in word sequence S is considered in order to preserve the syntactic properties of the language. For example, if we consider the phrase *Sam is charged for a crime*, as *charged* is a word that have a deviated sentiment, the vector corresponding to *charged* will be substituted by the vector of *hated* (*hated* is the word that matches the PoS of *charged* from the predefined word list corresponding to the negative class) [4]. When extending the approach proposed in [4] for three class sentiment classification, a predefined word list for positive class was developed by mapping a set of selected words that have positive sentiment in $RNTN_m$ to each PoS tag. The mapping can be represented as a dictionary R, where R = {JJ:beautiful, JJR:better, JJS:best, NN:masterpiece, NNS:masterpieces, RB:beautifully, RBR:beautifully, RBS:beautifully, VB:reward, VBZ:appreciates, VBP:reward, VBD:won, VBN:won, VBG:pleasing}. For the negative class and the neutral class, the PoS-word mappings provided by [4] for negative and non-negative classes were used respectively. Furthermore, instead of annotating each word in the selected vocabulary to identify words with deviated sentiments, we used word sets D_n, D_o, D_p that were derived using the approaches described in Section 4.2.1. From this point onwards, the fine tuned RNTN model developed in this study is denoted as $RNTN_l$.

5.2.3 BERT based Approach for Legal Sentiment Analysis

An approach based on $BERT_{large}$ embeddings [12] has achieved the state of the art results for sentiment classification of sentences in SST-5 dataset. In order to adapt the same approach for our task, following steps were followed. First, sentences with their sentiment labels were extracted from SST-5 training set. The SST-5 training set consists of 8544 sentences labelled for 5 class sentiment classification. As our focus is on 3 class classification, the sentiment labels in the SST training set were converted for 3 class sentiment classification by mapping very positive, positive labels as positive and very negative,negative labels as negative. Next, following a similar methodology as described in [12], *canonicalization*, *tokenization* and *special token addition* were performed as the preprocessing steps. Then, the classification model was designed following the same model architecture described in [12], that consists of a dropout regularization and a softmax classification layer on top of the pretrained BERT layer. Similarly to [12], $BERT_{large}$ uncased was used as the pretrained model and during the training phase, dropout of probability factor 0.1 was applied as a measure of preventing overfitting. Cross Entropy Loss was used as the cost funtion and stochastic gradient descent was used as the optimizer (batch size was 8). Then, the model was trained using the SST-5 training sentences. As information related to number of training epoch could not be found in [12], we experimented with 2 and 3 epcohs and calculated the accuracies with a test set of 500 legal phrases (Section 4,3). When trained for 2 epochs, the accuracy was 57% and for 3 epcohs it was reduced to 52%, possibly due to the overfitting with the source data. Therefore, 2 was choosen as the number of training epochs. This model will be denoted as $BERT_m$ in next sections.

In order to finetune the BERT based approach to the legal sentiment classification, the following steps were followed. First we selected sentences in the SST training data, that consists of words that were identified as having deviated sentiments (words in $D_o \cup D_p \cup D_n$). If the sentiment label of the sentence S that has a deviated sentiment word w is different from the sentiment label assigned to w by the legal expert, then S will be removed from the original SST training dataset as a measure of reducing negative transfer. For example, if there is a sentence S with word *charged* and if the sentiment of S is positive or neutral (sentiment of charged is negative in legal domain), then that sentence S will be removed from the training set. After removing such sentences, the

Table 5.1: Evaluating the word lists generated from Algorithm 1 and Algorithm 2

PolarityMetric	Number of Words				Percentages			
	N_m	N_l	P_m	P_l	N_m	N_l	P_m	P_l
Negative	**154**	**317**	17	20	**61%**	**80%**	5%	7%
Neutral	96	73	180	89	38%	19%	54%	41%
Positive	3	4	**139**	**181**	1%	1%	**41%**	**62%**
Total	253	394	336	290	100%	100%	100%	100%

training set was reduced to 6318 instances and this new training set will be denoted by D from this point forward. Next, for each word w in D_n or D_p, we randomly selected 2 sentences that contains w from the legal opinion text corpus. Then, the sentiments of the selected sentences were manually annotated by a legal expert. As $|D_n| = 206$ and $|D_p| = 82$, only 576 new annotations were needed ($|D_o| = 230$, but words in D_o were not considered for this approach as they are having a neutral legal sentiment). Then, these 576 sentences from legal opinion texts were combined together with sentences in D, thus creating a new training set L that consists of 6894 instances. The above mentioned steps were followed to remove the negative transfer from the source dataset and also to fine tune the dataset to the legal domain. Then, L was used to train a BERT based model using the same architecture, hyper parameters and number of training epochs that were used to train $BERT_m$. The model obtained after this training process is denoted as $BERT_l$.

5.3 Experiments and Results

5.3.1 Identifying words with deviated sentiments across the source and target domains

In order to evaluate the effectiveness of the proposed algorithmic approach when it comes to identifying legal sentiment of a word, we have compared the positive word list (P_l) and negative word list(N_l) identified by the algorithm with P_m and N_m respectively as shown in Table 5.1. The way in which P_l and N_l were obtained is described in Algorithm 2 and Algorithm 3. It can be observed that the precision of identifying words with negative sentiments is 80% in the algorithmic approach and it is a 19% improvement when compared with the $RNTN_m$ [10]. Furthermore, the number of correctly identified

Table 5.2: Precision(P), Recall (R) and F-Measure (F) obtained from the considered models

ModelMetric	Negative			Neutral			Positive			Accuracy
	P	R	F	P	R	F	P	R	F	
$RNTN_m$	0.51	0.68	0.58	0.44	0.52	0.48	0.48	0.10	0.16	0.48
$RNTN_l$ (Improved)	0.55	0.70	0.62	0.54	0.51	0.52	0.73	0.44	0.55	0.57
$BERT_m$	0.68	0.73	0.70	0.47	0.68	0.56	0.57	0.13	0.21	0.57
$BERT_l$ (Improved)	0.72	0.79	0.75	0.58	0.55	0.57	0.70	0.62	0.66	0.67

negative words have increase to 317 from 154. Though the precision of identifying words with positive sentiment is only 62%, there is an improvement of 21% when compared with the $RNTN_m$. Precision of identifying words with positive sentiment is relatively low due to the fact that most of the words that have a positive sentiment in generic language usage have a neutral sentiment in the legal domain. Sophisticated analysis in relation to the neutral class could not be performed due to the large amount of words available in O_m. When considering these results, it can be seen that the proposed algorithm has shown promising results when it comes to determining the legal domain specific sentiment of a word. Additionally, it implies that the proposed algorithmic approach is successful in identifying words that have different sentiments across the two domains. This approach can also be extended to other domains easily as domain specific word embedding models can be trained using an unlabelled corpus. Furthermore, the proposed algorithmic approach also has the potential to be used in automatic generation of domain specific sentiment lexicons.

5.3.2 Sentiment Classification

To evaluate the effectiveness of proposed approach to develop sentiment annotators for legal domain, a test set was prepared. The test set contains 500 sentences that were obtained from legal opinion text documents. Each of the sentences in the test set was annotated according to their sentiment in the legal domain. The following step wise procedure has been used to create this test set.

- From the corpus of legal opinion texts, 500 sentences were extracted randomly. During the process of sentence extraction, it was made sure that this test set does not contain any of the sentences that were selected from the legal opinion texts to train the model $BERT_l$.

- Then, each of the 500 sentences in this test set was annotated based on the legal sentiment. The annotations were performed by a legal expert (A graduate of the Faculty of Law, University of Colombo).

- After the annotation process, there were 211 sentences that were annotated as having a negative legal sentiment. The number of sentences that were classified as having a positive legal sentiment was 121 while there were 168 sentences that were classified as having a neutral sentiment.

For this experiment, $RNTN_m$ was taken as the baseline for the Recursive Neural Tensor Network based approaches and $BERT_m$ was taken as the baseline for the BERT based approaches. The annotations by the legal expert were taken as the ground truth. The results obtained for the models developed within this study ($RNTN_l$, $BERT_l$) as well as for the baseline models are shown in Table 5.2. Precision, Recall and F Score were used as the evaluation metrics and they are denoted as P, R and F respectively in Table 5.2.

Based in on the results as shown in the Table 5.2, the following observations can be made.

- The accuracy of $RNTN_m$ model is 0.48, while the $RNTN_l$ model has achieved an accuracy of 0.57. In other words, our proposed approach to fine tune RNTN and similar models by replacing word embeddings has yielded an accuracy improvement of 9%.

- The accuracy of $BERT_m$ model is 0.57, while $BERT_l$ model has achieved an accuracy of 0.67. This accuracy improvement demonstrates the effectiveness of our approach to fine tune a source dataset for a target dataset while using a minimum number of human annotations.

- The accuracy values obtained for both $BERT_m$ and $RNTN_l$ are the same (0.57). It should be noted the $BERT_m$ is purely trained using a dataset developed for movie reviews domain and without finetuning the dataset for the legal domain. This demonstrated that BERT based approaches are more effective than RNTN based approaches, when it comes to sentiment analysis. However, it can also be observed that the recall and F-score values that were achieved by $BERT_m$ model for Positive sentiment class is significantly low. In contrast, the $RNTN_l$

model has demonstrated relatively consistent performances over all three classes (Positive, Neutral and Negative).

- $BERT_l$ model outperforms all other models that were considered in our experiments. $BERT_l$ model was trained using the modified dataset that was obtained after fine tuning the SST-5 dataset for the legal domain using the approach proposed in this work under Section 4.2.3. $BERT_l$ model has achieved an overall accuracy of 0.67, which is a 10% accuracy improvement over $BERT_m$ model and also the $RNTN_l$ model.

- The state of the art accuracy value as denoted in [12] for five class sentiment classification of SST-5 dataset is 55.5%. Utilizing our proposed approaches, we have been able to achieve 67% accuracy for three class legal sentiment classification. It can be considered as a satisfactory accuracy value due to the complex nature of language used in the legal domain. It should be noted that in order to achieve these results, only 576 sentences were newly annotated and added to the training dataset.

The above observations clearly indicate that the transfer learning based approach based on dataset finetuning as described in Section 4.2.3 is an effective mechanism to develop sentiment annotators for a low resource domain. As this mechanism is based on dataset fine tuning, it should be noted that this method can be used with any state of the art machine learning or deep learning technique for sentiment analysis.

5.4 Discussion

In this research task, the main aim was to develop a reliable sentiment annotator to analyze the sentiments of textual information in legal opinion text while minimizing the resource requirements. In other words, this chapter describes how a legal sentiment annotator can be developed in a low resource setting. To this regard, we have made use of domain adaptation techniques, where the annotated sentiment analysis datasets in the movie review domain were adapted to the legal domain. Within this work, we have demonstrated several techniques that can be used to mitigate the issues that can be caused due to negative transfer. Coming up with an algorithmic approach to automatically identify the words that have different sentiment in the target domain

when compared with the sentiment of the source domain can be considered as a key contiribution this work has made towards this direction. After identifying such words with deviated sentiments across the two domains, the algorithmic approach also consists of a mechanism to assign the target domain sentiment to the identified words. The data sets prepared within this study to train and evaluate the sentiment analysis models are made publicly available [1].

[1] https://osf.io/zwhm8/

Chapter 6

CONCLUSION AND FUTURE WORK

As demonstrated in Chapter 4, the approaches that were developed within this work to develop sentiment annotators for the legal domain in low resource settings have shown promising results with significant improvements when compared with the existing works. The results further demonstrate that the domain specific word embeddings can be effectively utilized to minimize the drawbacks that are caused due to the negative transfer when adapting a dataset from one domain to another domain. Also, the evaluations that are described in Chapter 4 demonstrate that the algorithmic approach proposed in this work to automatically identify words with deviated sentiments across the source and target domains and to assign the appropriate target domain sentiment is successful and effective. Additionally, as detailed in Chapter 3, within this work, we have also developed a contextual verb similarity dataset for the legal domain named *LeCoVe* to overcome several drawbacks that are present within the existing similar evaluation resources. Using *LeCoVe* we have evaluated the effectiveness of the word representations of several word embedding and language modelling techniques when they are applied to the legal opinion text domain. During the process, we have also created new legal domain specific Sense2Vec models and also post trained a BERT model using a corpus of legal opinion text corpus. Finally, the datasets that were developed to train and evaluate the sentiment analysis models in the legal domain, the verb similarity dataset *LeCoVe* , the legal domain specific Sense2Vec models, and the BERT model post trained using a corpus of legal opinion texts have been made publicly available to the research community.

As future work, we believe that the approaches we have developed within this study for sentence level and phrase level sentiment analysis in the legal opinion text can be extended in order to be used in party based sentiment analysis in legal opinion texts. Also, the techniques we have described within this work to minimize the negative transfer are developed in a way that they can be easily adapted to other domain adaptation tasks in the Natural Language Processing domain. Moreover, *LeCoVe* can also be used to evaluate the effectiveness of other word representation and language modelling tech-

niques such as ELMO and XLNet. This can be seen as another future work related to this research work.

References

[1] United States Supreme Court. Lee v. US. In *Supreme Court*, volume 137, page 1958. Supreme Court, 2018.

[2] Gathika Ratnayaka, Thejan Rupasinghe, Nisansa de Silva, Viraj Salaka Gamage, Menuka Warushavithana, and Amal Shehan Perera. Shift-of-perspective identification within legal cases. *arXiv preprint arXiv:1906.02430*, 2019.

[3] Yi-Hung Liu and Yen-Liang Chen. A two-phase sentiment analysis approach for judgement prediction. *Journal of Information Science*, 44(5):594–607, 2018.

[4] Viraj Gamage, Menuka Warushavithana, Nisansa de Silva, Amal Shehan Perera, Gathika Ratnayaka, and Thejan Rupasinghe. Fast approach to build an automatic sentiment annotator for legal domain using transfer learning. *arXiv preprint arXiv:1810.01912*, 2018.

[5] Raksha Sharma, Pushpak Bhattacharyya, Sandipan Dandapat, and Himanshu Sharad Bhatt. Identifying transferable information across domains for cross-domain sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 968–978, 2018.

[6] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, 61(12):2544–2558, 2010.

[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204, 2010.

[8] Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology, 1999.

[9] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*, 2011.

[10] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Manish Munikar, Sushil Shakya, and Aakash Shrestha. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5. IEEE, 2019.

[13] Jack G Conrad and Frank Schilder. Opinion mining in legal blogs. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 231–236, 2007.

[14] Tomas Mikolov et al. word2vec. *URL https://code. google. com/p/word2vec*, 2013.

[15] Jeffrey Pennington et al. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[16] Andrew Trask et al. sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*, 2015.

[17] Matthew E Peters et al. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[18] Zhilin Yang et al. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[19] Christopher Schröder and Andreas Niekler. A survey of active learning for text classification using deep neural networks. *arXiv preprint arXiv:2008.07267*, 2020.

[20] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[21] Vern R Walker et al. Semantic types for computational legal reasoning: propositional connectives and sentence roles in the veterans' claims dataset. In *ICAIL*, pages 217–226. ACM, 2017.

[22] Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. Synergistic union of word2vec and lexicon for domain specific semantic similarity. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6. IEEE, 2017.

[23] Vindula Jayawardana et al. Semi-supervised instance population of an ontology using word vector embedding. In *ICTer*, pages 1–7. IEEE, 2017.

[24] Vindula Jayawardana et al. Word vector embeddings and domain specific semantic based semi-supervised ontology instance population. *ICTer*, 10(1):1, 2017.

[25] Vindula Jayawardana et al. Deriving a representative vector for ontology classes with instance word vector embeddings. In *INTECH*, pages 79–84. IEEE, 2017.

[26] Keet Sugathadasa et al. Legal document retrieval using document vector embeddings and deep learning. In *Science and Information Conference*, pages 160–175. Springer, 2018.

[27] Felix Hill et al. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[28] Daniela Gerz et al. Simverb-3500: A large-scale evaluation set of verb similarity. *arXiv preprint arXiv:1608.00869*, 2016.

[29] Eric H Huang et al. Improving word representations via global context and multiple word prototypes. In *ACL*, pages 873–882. Association for Computational Linguistics, 2012.

[30] Ray S Jackendoff. Semantic interpretation in generative grammar. 1972.

[31] Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.

[32] Kevin D Ashley and Vern R Walker. Toward constructing evidence-based legal arguments using legal decision documents and machine learning. In *ICAIL*, pages 176–180. ACM, 2013.

[33] Christopher Manning et al. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60, 2014.

[34] Kristina Toutanova et al. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL-HLT*, pages 173–180. Association for Computational Linguistics, 2003.

[35] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *ACL*, pages 133–138. Association for Computational Linguistics, 1994.

[36] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[37] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[38] C Van Rijsbergen. Information retrieval: theory and practice. In *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, pages 1–14, 1979.

[39] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[40] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[41] Fréderic Godin. *Improving and Interpreting Neural Networks for Word-Level Prediction Tasks in Natural Language Processing*. PhD thesis, Ghent University, Belgium, 2019.

[42] Christiane Fellbaum. Wordnet. *The encyclopedia of applied linguistics*, 2012.