FINAL YEAR PROJECT

GROUP 16 - FINAL REPORT

ONTOLOGY-BASED INFORMATION EXTRACTION FOR AUTOMATIC GENERATION OF LEGAL ARGUMENTS

Supervisors

Dr. Amal Shehan Perera

Mr. Nisansa de Silva

Students

Viraj Gamage (140173T)

Gathika Ratnayaka (140528M)

Thejan Rupasinghe (140536K)

Menuka Warushavithana (140650E)

August 18, 2021

Contents

List of Figures

List of Tables

Abstract

Legal professionals dedicate a major part of their time to read previous court cases to gather legal precedents, understand the evolution of law etc. To the best of our knowledge, none of the existing computer applications related to legal domain provides information within court case transcripts in an intuitive manner. This study attempts to address these issues by coming up with information extraction mechanisms that will ultimately facilitate the task of representing information in court case transcripts in a well-structured, intuitive manner. We have carried out multiple information extraction tasks in the legal domain that utilize concepts and methodologies which span across machine learning, natural language processing, semantic analysis, sentiment analysis, and linguistics. Our end goal is to create a system that is to be assistance to legal officials in their practice, specifically, a system that can automatically identify legal arguments, facts, and citations for a given legal case. First, we developed a system combining a machine learning model and a rule-based component to determine relationships among sentences in legal cases. Identifying relationships among sentences can be considered as a fundamental and important task. It will enable a computer system to identify the information flow within a court case transcript and also to determine the facts, evidence which is related to a particular legal argument. On the process of developing a system to automatically identify relationships among sentences, we experienced drawbacks in the existing sentiment annotators when used for analyzing texts in the legal domain. One of the reasons for the issues was that none of the sentiment annotators were trained using texts from the legal domain. Having an accurate sentiment analyzer is a crucial aspect when it comes to Information Extraction. Therefore, we came up with a novel and fast approach to build a sentiment annotator for the legal domain using transfer learning. Extracting arguments from court case transcripts can be considered as another information task with significant importance. Given a sentence from a court case transcript, a system should have the capability to determine whether the sentence is an argument or not. After analyzing various methodologies, it was decided to use linguistic and rule-based approaches to detect argumentative sentences. Classifying an argument as to whether it is in favor of the plaintiff or the defendant can be considered as a

crucial problem to be solved when it comes to information extraction from court case transcripts. In this study, we have demonstrated some approaches which can guide the process of solving this problem.

1 INTRODUCTION

1.1 Problem

When we consider the legal domain, a large number of past court case descriptions are available and the literature is growing every day. Specifically, Case Law (the law as established by the outcome of former cases) can be helpful in finding arguments that could be potentially helpful when handling a legal scenario. Legal officials (lawyers, paralegals) are required to read previous court cases and statutes to find arguments and evidence before they represent a petitioner or a defendant in a trial. As there are lots of cases that can be found related to a given scenario, lawyers have to dedicate a considerable part of their time reading and analyzing past cases and other legal documents to find necessary arguments, evidence, and related facts [1]. A system which can provide relevant arguments, evidence, and facts with their inter-relationships from case law, for a given legal scenario in natural language text, will be extremely useful for lawyers and other legal officials.

There are a few systems implemented to assist lawyers and other legal professionals in finding previous court cases. FindLaw [2], WestLaw [3] and BAILII [4] are capable of retrieving a large amount of legal cases. These search engines can retrieve relevant legal cases relevant to what the user requires. However, the searching mechanism is based only on querying using keywords. The searching process is solely based on lexical analysis but lacks the use of semantic analysis. The output provided is simply a legal case. Lawyers will not be automatically provided with an insight on the internals of legal cases.

There are a few commercial products implemented in the legal domain with natural language processing techniques and machine learning. “Ross Intelligence” [5] is one of them. It is claimed that “Ross Intelligence” has the ability to provide answers to questions raised in natural language by lawyers. In the meantime, it is capable of retrieving applicable statutes and case law-related statements for substantive legal issues raised by a client. Another example for such system is “CaseIQ” [6]. When a legal text or a legal document (or a brief) is inserted into the system, it can provide all the

statutory conditions relevant to that legal case. It also uses natural language processing to complete the task. “AI and La”, a system developed by LawGeek is capable of reviewing legal contracts.

None of the aforementioned systems provide insights on the information flow within a court case and the circumstances which lead to a particular argument or a decision. Therefore, the information related to Case Law that can obtained from these systems is limited. The proposed solution is Natural Language Processing (NLP) based legal information extraction system for the United States legal system, which will ultimately guide lawyers and paralegals to get relevant legal information regarding a court case of interest.

Therefore, a system which will clearly show the interdependencies between arguments, facts, and evidence in a court case transcript can be considered as a major requirement. It can also be considered as a major step of developing a system which will automatically suggest legal arguments for lawyers. We propose a tree like structure which contains the information in a court case transcript in an intuitive and well-structured manner. Fig. 1 shows a sample argument tree that would be generated by the system. In order to develop such a sophisticated system, systematic information extraction mechanisms relevant to court case transcripts are required. In this study, our primary intention is to develop such information extraction mechanisms, which will ultimately facilitate the task of developing a system which will comprehensively assist legal officials.

A part of a tree developed for a single legal case using the methodologies at the end of the project is shown in Fig. 2.

1.2 Motivation

Case Law can be described as a part of common law, consisting of judgments given by higher (appellate) courts in interpreting the statutes (or the provisions of a constitution) applicable in cases brought before them [7]. In order to find useful information related to Case Law for a given legal scenario, lawyers and other legal officials have to spend a significant amount of effort and time.

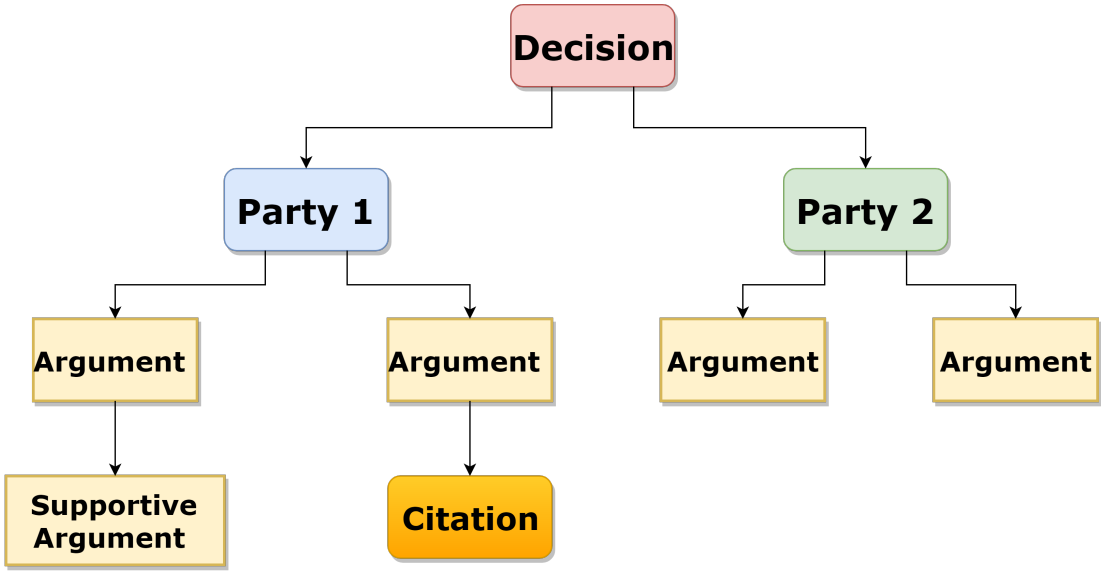

Fig. 1. Sample Argument Tree

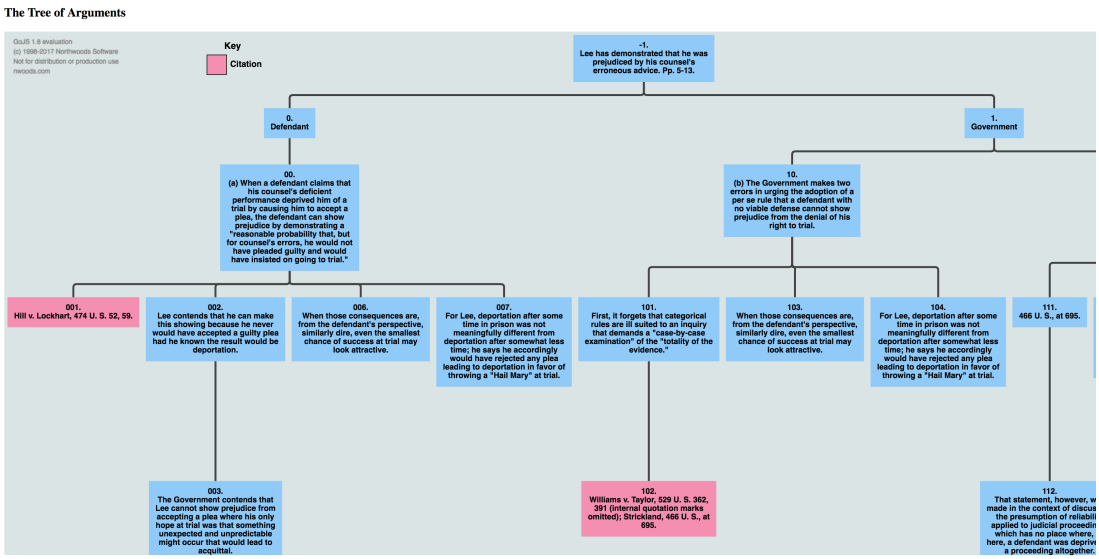

Fig. 2. Part of one of the system generated trees

The main motivation for the project is the absence of a system or a research methodology that can represent legal information in an intuitive and well-structured manner. While there are quite a few systems developed with the intention helping the legal officials be more effective in their profession, they do not provide a well-structured representation of legal information.

Our system and methodologies are to be of assistance to lawyers and other legal officials. They will help reduce the time and effort a lawyer has to put in to find related court cases and related arguments.

1.3 Research Objectives

The major research objective of this study is coming up with a methodology to identify the relationships among sentences in court case transcripts. Extracting sentences which provide legal arguments can be considered as another research objective. These research objectives are aligned with the ultimate objective of developing a methodology to extract information in court case transcripts in order to represent them in a well-structured manner.

2 LITERATURE REVIEW

Reading and understanding natural language text is a hard task for computers. The human understanding process of a given text is based on domain knowledge and past experiences. As the available information for a domain is increasing rapidly, the problem of natural language understanding in computers becomes even more challenging.

The process of information extraction is introduced to find solutions to the above problem by retrieving certain types of information from natural language text [8]. An ontology is a formal and explicit specification of a conceptualization of a given domain [9]. In Ontology-Based Information Extraction (OBIE), an ontology is used to make the process of information extraction more effective and efficient [10]. Open Information Extraction systems extract binary relational triples from plain text [11]. Open Information Extraction can also be used to support Ontology-Based Information Extraction when there are fewer identified relationships between the entities in the ontology [10].

Generally, ontologies can differ from domain to domain. In other words, there can be different ontologies which are specific for different domains such as the medical domain, legal domain, and sports domain. In their seminal paper Wimalasuriya and Dou states that OBIE is described as "a system that processes unstructured or semi-structured natural language text through a mechanism guided by ontologies to extract certain types of information and presents the output using ontologies" [8]. The set of relationships present in an ontology plays a key role when that ontology is used for OBIE. These relationships can be used to build extraction rules of the OBIE system. One of the main problems to be faced is the lack of an ontology in the legal domain which is capable of inferring rules to extract arguments from court cases in a generalized manner.

In order to overcome the drawbacks of OBIE, Open Information Extraction (Open IE) can be used. Open Information Extraction systems do not require pre-specified vocabulary to extract relational triples from a text. We have analyzed various Open IE tools such as Reverb [12], OLLIE [13] and Stanford OpenIE [11]. It can be seen that Stanford OpenIE suits best for this study than other tools as it outperforms other tools

due to various reasons such as the recursive relationship extraction [14].

Though Open Information Extraction is helpful in Information Extraction in Legal Domain, there are some problems to be overcome when using it. Such a problem is that, in court cases, words such as *defendant*, *petitioner*, and *government* are used to describe parties. But, Named Entity Recognition on these words (*defendant*, *petitioner*, *government*) are not taken as *Person* entities, so it is difficult to recognize them as different parties in a court case. (This is because the definition of a person in law expands beyond the definition of a person in linguistics. In fact, a person defined in linguistics is considered as a *natural person* in law, where *natural person* is only a proper subset of *person* given that another separately defined subset, *legal person* exists [15].)

Semantic Similarity Measures also play an important role when it comes to Natural Language Processing. In semantic similarity measures, the similarity of two entities is measured by the likeness of the semantic content of the two entities. In their paper, Wu and Palmer [16] have proposed a method using WordNet [17] which gives the similarity between two words on a scale of 0 to 1. An approach to measure the semantic similarity between two-word pairs using corpus statistics and lexical taxonomy is proposed in another study [18]. In the study [19], the above proposed methodologies are evaluated using WS4J [20].

Methodologies and comparisons available in studies related to semantic similarities [16–19] are evaluated in the study by de Silva et al [10]. This provides a detailed clarification on drawbacks occur when methodologies described in previous studies [16, 18] are used to measure the oppositeness between two sentences. In order to overcome the issues when measuring oppositeness between two entities, the study [10] has proposed a novel solution with a set of equations to calculate oppositeness. As this project intends to develop a measure to find the relationship between sentences, the oppositeness measure proposed in the study by de Silva et al [10] will be useful. Again, that study [10] has focused on finding inconsistencies in literature related to the medical domain. The effectiveness of the proposed oppositeness measure [10] needs to be evaluated for the legal domain.

Irrespective of the domain, understanding how information is related to each other in machine-readable text has always been a challenge when it comes to Information Extraction. Analyzing of relationships between sentences can be considered as an effective approach to understanding the way how two textual units are connected with each other. Though there are researches based on semantic similarity between two sentences [21], semantic similarity alone does not provide any information on the relationship between the sentences. Analyzing discourse relationships between sentences can be considered as a more sophisticated approach when it comes to determining the type of relationship existing between two sentences. Several theories related to discourse structures have been proposed in recent years. Cross-document Structure Theory (CST) [22] , Penn Discourse TreeBank (PDTB) [23], and Rhetorical Structure Theory (RST) [24] can be considered as prominent discourse structures. The effectiveness of analyzing discourse relations between sentences has become evident due to its successful applications in different domains such as text summarizing [25, 26], question answering systems[27] and natural language generation [28].

The way in which the relationship types are defined is different from one discourse structure to another as they have been intended for different purposes. However, none of these discourse structures are specifically intended to be used in the legal domain. Though that is the case, many of the linguistic patterns in the English language are the same, regardless of the domain in which the language is being used. There are studies [25, 29] which demonstrate how original discourse structures like CST can be redefined in order to facilitate different research purposes related to different domains. Furthermore, the study [30] discusses the potential of discourse analysis for extracting information from legal text. Therefore, we see discourse analysis as a viable mechanism to identify relationships among sentences in court case transcripts.

Discovering situations where two sentences provide different opinions on the same topic or entity is an important part when it comes to identifying relationships among sentences. There are studies [31, 32] which have proposed approaches to find contradictions or contrastive viewpoints in text. Both of these studies emphasize that before determining the contradictions or contrastive viewpoints, it is important to identify

whether both textual units are discussing on the same topic or the same event. The study by Paul et al [32] provides steps to cluster documents on viewpoints using dependency features in a text. The methodology which is provided in that study [32] cannot be used as it is when determining contradictions in court case transcripts. The reason is finding contradictions in court case transcripts needed to be done in sentence (including argumentative sentences) level while considering the respective parties in the court case. But the dependency features in a text which are considered in the study [32] can be useful to get insight on the properties of sentences that should be considered when determining contrastive viewpoints or contradictions. In the study by Marneffe et al [31], the RTE dataset [33] has been annotated for contradiction. The annotated dataset has been used to determine contradictions. But, to the best of our knowledge, there is no available dataset which has been annotated for contradiction relationships which can be seen among sentence pairs in United State Court Case transcripts. Therefore, it is needed to come up with a novel approach to determine contrastive viewpoints in court cases transcripts.

Extracting argumentative sentences from court case transcripts is another significant tasks when it comes to information extraction in the legal documents. Various researches have been carried out on automatic extraction of arguments from legal texts. The study [34] by Wyner, et al. brings out an extensive background research on the literature of argumentation and argument extraction with an analysis of various argument corpora. *Araucaria* [35, 36] is a database of arguments from various sources and a tool for diagramming and representing arguments. In another study [35], Reed and Rowe, introducing Araucaria tool, point out that arguments can be graphically represented in a tree, where premises are being branched off of conclusions. Arguments in AraucariaDB are manually annotated and marked up in an XML-based format, AML (*Argument Markup Language*). The study by Wyner et al [34] also presents out how legal arguments can be extracted, using a Context-Free Grammar. It describes legal argument construction patterns, to identify premises and conclusions, which they came up with, by analyzing legal cases from ECHR (European Court of Human Rights). Studies [37, 38] on legal argument automatic detection is also done on ECHR cases.

Moens, et al. describe argument detection as a sentence classification problem between arguments and non-arguments [37]. There a classifier is trained on a set of manually annotated arguments, considering sentences in isolation. They have evaluated different feature sets involving lexical, syntactic, semantic and discourse properties of the texts. In the study [38] Mochales and Moens, points out that arguments are always formed by premises and conclusions. So they have determined argument extraction as a sentence classification problem among premises, conclusions, and non-arguments. Furthermore, they have improved the feature set used in [37] by including features that refer to content in previous sentences.

All these researches, done on argument extraction, have used ECHR cases as their corpus. To the best of our knowledge, there has been no research carried out about argument extraction from US court case transcripts. Argument patterns identified in the study [34] are very rigid and they have specifically been identified for ECHR cases. The reporting structures in US Court Case transcripts are significantly different from ECHR case reports. Therefore, the rules that are described in the study [34] are not directly applicable for extracting argumentative sentences from US Court Cases. Also, to the best of our knowledge, there is no existing annotated corpus which contains argumentative sentences extracted from US court cases. The consequence is machine learning approaches described in previous studies on argument identification in ECHR cases [37, 38] cannot be used. Therefore, it is needed to come up with a novel way to identify arguments and non-arguments in US court case transcripts, with the guidance of a legal expert.

Non-argumentative sentences in court case transcripts provide facts, evidence and background information on the legal scenario which is being discussed in the court case. Therefore, developing a mechanism to discover relationships among legal facts and arguments is another key objective of this research. This is because when generating the legal arguments tree for a particular court case transcript, several relationship patterns between the legal scenario, extracted legal arguments, extracted legal facts and conditions have to be considered. One fact or an argument may elaborate, provide evidence or may even contradict another argument. There are studies which have been

done to identify discourse relationships between sentences [25, 39]. There are large scale corpora such as Penn Discourse Treebank [23], Cross-document Structure Theory bank [22] which are annotated with the information related to discourse semantics.

Sentiment analysis is another key area when it comes to Natural Language Processing. In the lexical resource called SENTIWORDNET [40, 41], sentiment classification of WORDNET [42] synsets is done automatically with respect to three classes; "negative", "positive" and "objective". If a certain synset does not have opinionated content, it belongs to the "objective" class. Otherwise, it is called to be "subjective" which is further classified into other two classes "negative" and "positive" depending on the sentiment it carries. There have been many approaches taken to classify sentiment of phrases and sentences using SENTIWORDNET. [43] proposes a methodology to perform opinion mining on movie reviews using support vector machine and some of the features were calculated using WORDNET. The paper [43] states that the accuracy of sentiment classification using the proposed approach is 69.35%. Further, it says that inaccuracies in SENTIWORDNET feature calculations are caused by the SENTIWORDNET's reliance on glosses.

Rather than relying on sentiment calculations for glosses given by SENTIWORDNET, the study by Socher et al [44] provides a method to identify the sentiment of a phrase or a sentence in a supervised manner using Recursive Neural Tensor Network which is a deep learning model. This learning model has the capability to identify the sentiment considering the context of that word [44]. The dataset consists of movie reviews. Each sentence in the data set is broken into phrases and each phrase is annotated by human judges. It is mentioned in the paper [44] that the provided methodology shows 80.7% of accuracy in phrase level. But still, that trained model cannot be used with the same accuracy to classify sentiment in the legal domain because the sentiment carries by a certain word depends on the context it is referred. This point is further elaborated in the methodology section. As the paper [44] states, the model can be trained over any domain by following the provided methodology. The major difficulty lies with preparing a manually annotated dataset over the legal domain. The data-set which is used in [44], contains 215,154 manually annotated phrases (from 11,855

sentences) over 5355 unique words. But, the vocabulary of the court case corpus used in our process is above 17000, thus demonstrating that the linguistic complexity of domain is larger compared to movie reviews.

We can observe that the Recursive Neural Tensor Network (RNTN) model by Socher et al [44] shows better accuracy in sentiment classification in the legal domain compared to other existing models, despite the model is trained using movie reviews. Our aim is also to build a sentiment classifier which works with a higher accuracy in the legal domain. But, in order to prepare manually labeled data set for training purpose is a costly process in terms of time and human effort. Therefore, the method called "transfer learning" is used to adapt the RNTN model [44] to the legal domain. When a learning model is trained using data from a certain domain and tested with respect to a different domain, it is called "Transfer Learning" [45]. Since the task is the same in both source [44] and the target model for the legal domain, the task belongs to the subcategory called "Domain Adaptation" as mentioned in [45, 46]. Image classification is a field where transfer learning is vastly used [45, 47]. In the previous studies [45, 46], it is mentioned that domain adaptation can be used for sentiment classification tasks.

3 METHODOLOGY

There are several platforms where we can find more than 10,000 court case transcripts related to United States. However, it is not possible to use Google scholar because the Google servers block the IP address after a few calls (of using a web crawler) due to security issues. The site we are using is FindLaw [2], where we can obtain more than 40000 legal cases. Those legal cases are properly categorized into several groups so that we can search the cases depending on their relevant category (for example environment, criminal procedure, consumer law, etc).

The ultimate solution to address the identified problem is to represent the legal arguments in a well-structured manner as shown in Figure 1. Building a complete system to achieve this ultimate goal requires a lot of research. This can be identified as a combination of different sub-tasks. In this study, our main concern is to research on identifying relationships among sentences in a court case transcript.

As the starting point, it is required to introduce a methodology to identify the information flow of a legal case using discourse relations. For that task, the "*Sentence Relationship Identifier*" component is implemented. After referring to legal cases, five relationship types are discovered; *elaboration*, *no-relation*, *shift-in-view*, *redundancy* and *citation*. When a sentence pair is provided as an input to the component, it has the capability to identify the relationship between the two sentences. The implementation details, research background and performance evaluations are explained in Section 3.1. The component has been able to identify elaboration, no-relation and citation relationships with a very high accuracy. But when it comes to identifying shift-in-view relation, the accuracy has been significantly low. That has been the main drawback of the system.

To improve accuracy in shift-in-view relation, we have decided to incorporate sentiment analysis to our study. In shift-in-view relation, both sentences should talk about a same person or entity. But those should express different opinions. It is not the best way to use existing sentiment annotators like *Stanford CoreNLP Sentiment Annotator* [44] and *Sentiwordnet*[40, 41], because those annotators are not trained for legal domain. Therefore, we had to develop the *Sentiment Annotator* component for

the legal domain. When a phrase or sentence is provided as input to the component, it provides output with its sentiment according to the legal domain. The research corresponding to this component is elaborated in Section 3.2.

To address the drawback in the *Sentence Relationship Identifier*, we have implemented another component for the identification of "Shift-in-View" relationship. We will refer to that component as "Shift-in-View Identifier". This component is a sub component of the *Sentence Relationship Identifier*. When a sentence pair is fed to Sentence Relationship Identifier, first it checks for the relationship in between. If the output is *elaboration*, then the sentence pair will be given as input to the "Shift-in-View Identifier". Then that component will do a binary classification to identify if there is a *shift-in-view* between the pair or not. The research on *shift-in-view* identification includes three separate methodologies including the sentiment based approach. All the details related to this study has been stated in Section 3.3. we could improve the accuracy in identifying shift-in-view relationship significantly through the newly introduced component.

After achieving reasonably better results for identifying relationships between the sentences, we have moved to identify argumentative sentences from legal cases. It is required to identify relations between the legal arguments rather than considering the whole text in the court case. To accomplish that task, we have implemented the "*Argument Extractor*" component. Prior to the implementation, we had a discussion with a legal expert to identify the requirements that should be satisfied in order to identify a sentence as an argument. When a sentence is provided as input, the component returns whether the argument can be considered as a legal argument or not. Further details about the research related to this study is elaborated in Section 3.4.

As a summary, we can say that the relationships between sentences play the most vital part of this system because it allows a system to automatically identify the connection between arguments, supporting facts while providing a good understanding about the flow of information within a court case. Discourse relations, Linguistic rules, and patterns can be used together with Natural Language Processing and Machine Learning techniques to identify relationships between sentences.

Section 3.1 describes the study we carried out to identify relationships among sentences in court case transcripts. After analyzing the outcomes of that research, we have conducted a separate research on identifying situations where two sentences are providing different opinions on the same topic. The overall summary of that study is presented in Section 3.3 with results. In order to facilitate that research, we have developed a Sentiment Annotator for the legal domain. The fast approach we followed in developing the sentiment annotator is described in section 3.2 while demonstrating the improvements we got in relation to the existing systems. Section 3.4 provides an overall idea on the methodologies that were used in extracting argumentative sentences in court case transcripts. Section 3.5 describes potential methodologies that can be used for party identification in court case transcripts. A brief description about the manually annotated dataset is provided in Section 3.6. Sections 3.7, 3.8, 3.9, 3.10 provide details on individual contributions of Viraj Gamage, Gathika Ratnayaka, Thejan Rupasinghe and Menuka Warushavithana throughout the whole project.

3.1 Identifying Relationships Among Sentences in Court Case Transcripts Using Discourse Relations

3.1.1 Introduction to the study

In the process of extracting information from legal court cases, it is important to identify how arguments and facts are related to one another. The objective of this study is to automatically determine the relationships between sentences which can be found in documents related to previous court cases of United States Supreme Court. Transcripts of U.S. court cases were obtained from FindLaw[1] following a method similar to numerous other artificial intelligence applications in the legal domain [48–52].

When a sentence in a court case is considered, it may provide details on arguments or facts related to a particular legal situation. Some sentences may elaborate on the details provided in the previous sentence. It is also possible that the following sentence may not have any relationship with the details in the previous sentence and may provide

[1] https://caselaw.findlaw.com/

details about a completely new topic. Another type of relationship is observed when a sentence provides contradictory details to the details provided in the previous sentence. Determining these relationships among sentences is vital to identifying the information flow within a court case. To that end, it is important to consider the way in which clauses, phrases, and text are related to each other. It can be argued that identifying relationships between sentences would make the process of Information Extraction from court cases more systematic given that it will provide a better picture of the information flow of a particular court case. To achieve this objective, we used discourse relations based approach to determine the relationships between sentences in legal documents.

Several theories related to discourse structures have been proposed in recent years. Cross-document Structure Theory (CST) [22] , Penn Discourse Tree Bank (PDTB) [23], Rhetorical Structure Theory (RST) [24, 53] and Discourse Graph Bank [54] can be considered as prominent discourse structures. The main difference that can be observed between each of these discourse structures is they have defined the relation types in a different manner. This is mainly due to the fact that different discourse structures are intended for different purposes. In this study, we have based the discourse structure on the discourse structure proposed by CST.

A sentence in a court case transcript can contain different types of details such as descriptions of a scenario, legal arguments, legal facts or legal conditions. The main objective of identifying relationships between sentences is to determine which sentences are connected together within a single flow. If there is a weak or no relation between two sentences, it would probably infer that those two sentences provide details on different topics. Consider the sentence pair taken from *Lee v. United States* [55] shown in Example 1.

Example 1

- Sentence 1.1: *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.*
- Sentence 1.2: *First, it forgets that categorical rules are ill suited to an inquiry that demands a "case-by-case examination" of the "totality of the evidence".*

It can be seen that sentence 1.2 elaborates further on the details provided by sen-

tence 1.1 to give a more comprehensive idea on the topic which is discussed in sentence 1.1. These two sentences are connected to each other within a same flow of information. This can be considered as *Elaboration* relationship, which is a relation type described in CST. Now, Consider the sentence pair shown in Example 2 which was also taken from *Lee v. United States* [55].

Example 2

- Sentence 2.1: *Courts should not upset a plea solely because of post hoc assertions from a defendant about how he would have pleaded but for his attorney's deficiencies.*
- Sentence 2.2: *Rather, they should look to contemporaneous evidence to substantiate a defendant's expressed preferences.*

In Example 2, it can be seen that the two sentences have the *Follow Up* relationship as defined in CST. But still, these two sentences are connected together within the same information flow in a court case. There are also situations where we can see sentences are showing characteristics which are common to multiple discourse relations. Therefore, several discourse relations can be grouped together based on their properties to make the process of determining relationships between sentences in court case transcripts more systematic.

The two sentences for Example 3 were also taken from *Lee v. United States* [55]:

Example 3

- Sentence 3.1: *The question is whether Lee can show he was prejudiced by that erroneous advice.*
- Sentence 3.2: *A claim of ineffective assistance of counsel will often involve a claim of attorney error "during the course of a legal proceeding"–for example, that counsel failed to raise an objection at trial or to present an argument on appeal.*

The sentence 3.2 follows sentence 3.1. A significant connection between these two sentences cannot be observed. It can also be seen that sentence 3.2 starts a new flow by deviating from the topic discussed in sentence 3.1. These observations which were provided by analyzing court cases emphasize the importance of identifying relationships between sentences.

In this study, we defined the relationship types that are important to be considered when it comes to information extraction from court cases. Next, for each of the relationship type we defined, we identified the relevant CST relations [22]. Finally, we developed a system to predict the relationship between given two sentences of a court case transcript by combining a machine learning model and a rule-based component.

3.1.2 Literature Review - Identifying Relationships Among Sentences in Court Case Transcripts Using Discourse Relations

Understanding how information is related to each other in machine-readable texts has always been a challenge when it comes to Natural Language Processing. Determining the way in which two textual units are connected to each other is helpful in different applications such as text classification, text summarization, understanding the context, evaluating answers provided for a question. Analyzing of discourse relationships or rhetorical relationships between sentences can be considered as an effective approach to understanding the way how two textual units are connected with each other.

Discourse relations have been applied in different application domains related to NLP. [56] describes CST[22] based text summarization approach which involves mechanisms such as identifying and removing redundancy in a text by analyzing discourse relations among sentences. [57] compares and evaluates different methods of text summarizations which are based on RST [53]. In another study [58], text summarization has been carried out by ranking sentences based on the number of discourse relations existing between sentences.[26, 59, 60] are some other studies where discourse analysis has been used for text summarization. These studies related to text summarization suggest that discourse relationships are useful when it comes identifying information that discusses on same topic or entity and also to capture information redundancy. Analysis of discourse relations has also been used for question answering systems [27, 61] and for natural language generation [28].

In the study [62], discourse relations existing between sentences are used to generate clusters of similar sentences from document sets. This study shows that a pair of sentences can show properties of multiple relation types which are defined in CST [22].

In order to facilitate text clustering process, discourse relations have been redefined in this study by categorizing overlapping or closely related CST relations together. In [29], the discourse relationships which are defined in [62] have been used for text summarization based on text clustering. The studies [29, 62] emphasize how discourse relationships can be defined according to the purpose and objective of the study in order to enhance the effectiveness.

When it comes to the legal domain, [30] discusses the potential of discourse analysis for extracting information from legal texts. [63] describes a classifier which determines the rhetorical status of a sentence from a corpus of legal judgments. In this study, rhetorical annotation scheme is defined for legal judgments. The study [64] provides details on summarization of legal texts using rhetorical annotation schemes. The studies [63, 64] focus mainly on the rhetorical status in a sentence, but not on the relationships between sentences. An approach which can be used to detect the arguments in legal text using lexical, syntactic, semantic and discourse properties of the text is described in [37].

In contrast to other studies, this study is intended to identify relationships among sentences in court case transcripts by analyzing discourse relationships between sentences. Identifying relationships among sentences will be useful in the understanding of information flow within a court case.

3.1.3 Defining Discourse Relationships in Court Cases

Five major relationship types were defined by examining the nature of relationships that can be observed between sentences in court case transcripts.

- Elaboration - One sentence adds more details to the information provided in the preceding sentence or one sentence develops further on the topic discussed in the previous sentence.

- Redundancy - Two sentences provide the same information without any difference or additional information.

- Citation - A sentence provides references relevant to the details provided in the previous sentence.
- Shift-in-View - Two sentences are providing conflicting information or different opinions on the same topic or entity.
- No Relation - No relationship can be observed between the two sentences. One sentence discusses a topic which is different from the topic discussed in another sentence.

After defining these relationships, we adopted the rhetorical relations provided by CST [22] to align with our definitions as shown in the table below.

Table 3.1: Adopting CST Relationships

Definition	**CST Relationships**
Elaboration	Paraphrase,Modality,Subsumption,Elaboration, Indirect Speech, Follow-up, Overlap, Fulfillment, Description, Historical Background, Reader Profile,Attribution
Redundancy	Identity
Citation	Citation
Shift-in-View	Change of Perspective,Contradiction
No Relation	-

It is very difficult to observe the same sentence appearing more than once within nearby sentences in court case transcripts. However, we have included it as a relationship type in order to identify redundant information in a case where the two sentences in a sentence pair are the same.

3.1.4 Expanding the Dataset

A Machine Learning model was developed in order to determine the relationship between two sentences in court cases. We used the publicly available dataset of CST bank [65] to learn the Model. The dataset obtained from CST bank contains sentence pairs which are annotated according to the CST relation types. Since we have a labeled

dataset [65], we performed supervised learning to develop the machine learning model. Support Vector Machine (SVM) was used indeveloping the machine learning model. As the training dataset in CST Bank contains only 2396 instances, the dataset is insufficient to train a deep learning model. Additionally, the CST bank dataset does not made up of sequential sentences, thus eliminating the possibility to use models such as LSTM. SVMs are prefered over Naive Bayesian models when it comes to text classification, because it is more than likely that features extracted from text to be non independent, thus violating the naive assumption of Naive Bayesian models. Also, SVMs have been used in previous studies where discourse relations have been used to identify relationships between sentences [29, 62] and promising results were shown in each of these studies.

Table 3.2 provides details on the number of sentence pairs in the data set for each relationship type.

Table 3.2: Number of sentence pairs for each relationship type

CST Relationship	**Number of Sentence Pairs**
Identity	99
Equivalent	101
Subsumption	590
Contradiction	48
Historical Background	245
Modality	17
Attribution	134
Summary	11
Follow-up	159
Indirect Speech	4
Elaboration	305
Fulfillment	10
Description	244
Overlap (Partial Equivalence)	429

By examining the CST relationship types available in the dataset Table3.2, it can be observed that a relationship type which suggests that there is no relationship between sentences cannot be found. But "No Relation" is a fundamental relation type that can be observed between two sentences in court case transcripts. Therefore, we expanded the

data set by manually annotating 50 pairs of sentences where a relationship between two sentences cannot be found. This new class was named "No Relation". The 50 sentence pairs which were annotated were obtained from previous court case transcripts.

A sentence pair is made up of a source sentence and a target sentence. The source sentence is compared with the target sentence when determining the relationship that is present in the sentence pair. For example, if the source sentence contains all the information in target sentence with some additional information, the sentence pair is said to have the subsumption relationship. Similarly, if the source sentence elaborates the target sentence, the sentence pair is said to have the elaboration relationship.

3.1.5 Developing a SVM Model to Determine the Relationship Between Sentences

In order to train the SVM model with annotated data, features based on the properties that can be observed in a pair of sentences were defined. Before calculating the features related to words, we removed *stop words* in sentences to eliminate the effect of less significant words. Also, co-referencing was performed on a given pair of sentences using Stanford CoreNLP CorefAnnotator (“coref") [66] in order to make the feature calculation more effective.

All the features were calculated and normalized such that their values fall within $[0, 1]$ range. We have defined 9 feature categories based on the properties that can be observed in a pair of sentences. Following 5 features were adopted from previous studies related to discourse analysis [25, 29].

1. Cosine Similarities

   Following cosine similarity values are calculated for a given sentence pair,

   - Word Similarity
   - Noun Similarity
   - Verb Similarity
   - Adjective Similarity

2. Word Overlap Ratios

3. Grammatical Relationship Overlap Ratios

4. Longest Common Substring Ratio

5. Number of Entities

In addition to the above mentioned features, following features have been introduced to the system. More details on these features and the rationale behind introducing them are described in detailed manner under the Section 3.8.1.4

1. Semantic Similarity between Sentences

2. Transition Words and Phrases

3. Length Difference Ratio

4. Attribution (This feature checks whether a sentence describes a detail in another sentence in a more descriptive manner)

3.1.6 Determining Explicit Citation Relationships in Court Case Transcripts

In legal court case documents, several standard ways are used to point out whence a particular fact or condition was obtained. After observing different ways of providing citations in court case transcripts, a rule-based mechanism to detect such citations was developed. If this rule-based system detects that there is citation relationship, the pair of sentences will be assigned with the citation relationship.

3.1.7 Experiments

In order to determine the effectiveness of our system, it is important to carry out evaluations using legal court case transcripts, as it is the domain this system is intended to be used. Court case transcripts related to United States Supreme Court were obtained from Findlaw[2]. Then the transcripts were preprocessed in order to remove unnecessary data and text. Court case title, section titles are some examples of details which were

removed in the preprocessing process. Those details are irrelevant when it comes to determining relationships between sentences.

The results obtained using the system for the sentence pairs extracted from the court case transcripts were then stored in a database. From those sentence pairs, 200 sentence pairs were selected to be annotated by human judges. Then the selected 200 pairs of sentences to be annotated were grouped together as clusters of five sentence pairs. Each cluster was annotated by two human judges who were trained to identify the relationships between sentence pairs as defined in this study.

As we did not have annotated data, we needed to employ a way to get pairs of sentences annotated to measure the accuracy of our model. Therefore, we created an annotation application (web-based) to get this task completed with the help of other people including our batch mates. The details of this application are described later in the chapters on individual contributions.

3.1.8 Results from the study

As expected, the redundancy relationships between sentences could not be observed within the sentence pairs which were annotated using human judges. From the 200 sentence pairs that were observed, our system did not predict redundancy relationship for any sentence pair. Similarly, human judges did not annotate the "redundancy" relationship for any sentence pair.

The confusion matrix which was generated according to the results obtained is given in Table 3.3. The details provided in the matrix are based only on the sentence pairs that were agreed by two human judges to have a similar relationship. The reasoning behind this approach is to eliminate sentence pairs where there are ambiguities of the relationship type between them.

The same approach was used to obtain the results which are presented in Table 3.4. In contrast, Table 3.5 contains results obtained by considering sentence pairs where at least one of the two judges who annotated the pair agrees upon a particular relationship type.

Evaluation results from Table 3.4, Table 3.5 the system works well when identifying

Table 3.3: Confusion Matrix

Actual \ Predicted	Elaboration	No Relation	Citation	Shift In View	Number of Sentence Pairs
Elaboration	93.9	6.1	0.0	0.0	101
No Relation	11.9	88.1	0.0	0.0	44
Citation	0.0	4.8	95.2	0.0	20
Shift-in-View	100.0	0.0	0.0	0.0	0
Number of Sentence Pairs	99	42	21	3	165

Table 3.4: Results Comparison of Pairs Where Both Judges Agree

Discourse Class	Precision	Recall	F-Measure
Elaboration	0.921	0.939	0.930
No Relation	0.841	0.881	0.861
Citation	1.000	0.952	0.975
Shift-in-View	-	0	-

"Elaboration", "No Relation" and "Citation" relationship types where F-measure values are above 75% in all cases. "Shift-in-View" relationship type was not assigned by the system to any of the 200 sentence pairs which were considered in the evaluation.

Human vs Human correlation and Human vs System correlation when it comes to identifying these relationship types were also analyzed. First, we calculated these correlations without considering the relationship type using the following approach. For a given sentence pair P, m(P) is the value assigned to the pair. n is the number of sentence pairs.

1. Human vs Human Correlation (Cor(H,H))

When both human judges are agreeing on a single relationship type for the pair P, we assign m(P) = 1. Otherwise, we assign m(P)=0.

$$Cor(H,H) = \frac{\sum_{P=1}^{n} m(P)}{n} \tag{1}$$

2. Human vs System Correlation (Cor(H,S))

Table 3.5: Results Comparison of Pairs Where at least One Judge Agrees

Discourse Class	Precision	Recall	F-Measure
Elaboration	0.930	0.902	0.916
No Relation	0.846	0.677	0.752
Citation	1.000	0.910	0.953
Shift-in-View	-	0	-

When both human judges are agreeing with the relationship type predicted by the system for the sentence pair P, we assign m(P) = 1. If only one human judge is agreeing with the relationship type predicted by the system for P, we assign m(P) = 0.5. If both human judges disagree with the relationship type predicted by the system for P, we assign m(P) = 0.0.

$$Cor(H,S) = \frac{\sum_{P=1}^{n} m(P)}{n} \tag{2}$$

The following results could be observed after calculating the correlations,

The correlation between a human judge and another human judge = 0.805

The correlation between a human judge and the system = 0.813

When analyzing these two correlations, it can be seen that our system performs with a capability which is near to the human capability.

The results obtained by calculating Human vs. Human and Human vs. System correlations in relation to each relationship type are given in Table 3.6. The approach which is described below was used to calculate these two correlations for each relationship type.

Consider relationship type R, Let,

S = the set containing all the sentence pairs which are predicted by the system as having the relationship type R

U = the set containing all the sentence pairs which were annotated by at least one human judge as having the relationship type R

V = the set containing all the sentence pairs which were annotated by two human judges as having the relationship type R.

Corr(H,H) represents Human vs Human correlation and Corr(H,S) represents Human vs System correlation. For a given set A, n(A) indicates number of elements in set A.

$$Corr(H,H) = \frac{n(V)}{n(U)} \tag{3}$$

$$Corr(H,S) = \frac{n(S \wedge U)}{n(S \vee U)} \tag{4}$$

The results obtained using this approach is provided in Table 3.6.

Table 3.6: Correlations by Type

Discourse Class	Human-Human	Human-System	$\frac{\text{Human-System}}{\text{Human-Human}}$
Elaboration	0.75	0.843	1.124
No Relation	0.646	0.603	0.933
Citation	1.0	0.955	0.955
Shift-in-View	0.188	0.0	0.0

The results which are in Table 3.6 suggest that the system performs with a capability which is near to the human capability when it comes to identifying relationships such as Elaboration, No Relation, and Citation in court case transcripts. Enhancing system's ability to identify "Shift-in-View" relationship is one of the major future challenges. At the same time Human vs Human correlation when it comes to identifying "Shift-in-View" relationship type is 0.188. This indicates that humans are also having ambiguities when identifying "Shift-in-View" relationships between sentences in court case transcripts.

Either "Elaboration" or "Shift-in-View" Relationship occurs when the two sentences are discussing the same topic or entity."Shift-in-View" relationship occurs over "Elaboration" when two sentences are providing different views or conflicting facts on the same topic or entity. The "No Relation" relationship can be observed between two sentences when two sentences are no longer discussing the same topic or entity. In other words, the "No Relation" relationship suggests that there is a shift in the information flow in court cases. As shown in Table 3.3, the sentence pairs with "Shift-in-View" relationship

are always predicted as having "Elaboration" relationship by the system. By observing these results, it can be seen that in most of the cases the system is able to identify whether the sentences are discussing the same topic or not. From this point on-wards we refer the system developed in this study as **Sentence Relationship Identifier**.

3.2 A Fast Approach to Build an Automatic Sentiment Annotator for Legal Domain using Transfer Learning

3.2.1 Introduction to the study

We designed a novel way of finding sentiments of text in court case transcripts using transfer learning. Accurate Sentiment analysis of legal text is a key part of completing the rest of our project. There were several reasons behind deciding to implement a new sentiment analyzer. The inability of the existing sentiment analyzers to infer legal context from a text, which is understandable as Law is a rather complex and esoteric field, it is challenging for a human to fully comprehend the emotion behind a given statement, let alone for a machine.

As [40] suggests, sentiment classification is a recent methodology aligns with information retrieval and computational linguistics which is focused on the opinion towards something which is represented by a certain text. To identify the requirement of sentiment classification, consider the following example which is extracted from a legal case.

- Sentence: The District Court concluded that Lee's counsel had performed deficiently.

In the above example, the "had performed deficiently" phrase induces a negative sentiment towards Lee's counsel. The sentiment of "concluded that" denotes that court agrees with the inner sentence. Complete sentence denotes that court's opinion towards Lee's counsel is negative. Consider the following,

- Sentence: the Government conceded that Lee's counsel had performed deficiently.

This sentence contains the same inner sentence, but in legal domain the phrase called "conceded that" indicates a situation where government initially disagreed but eventually

did agree. Therefore, that phrase induces a negative sentiment on the inner sentence which is negative towards the Lee's counsel. Therefore, it is fair to assume that the government and Lee's counsel were on the same side in this situation.

The sentiment analysis system which was developed specifically for this domain can be used to identify major parties in a court case, to evaluate the bias of an argument or a fact towards a party, and in the study described in Section 6.3.

3.2.2 Literature Review - A Fast Approach to Build an Automatic Sentiment Annotator for Legal Domain using Transfer Learning

Owing to the difficulties in handling legal jargon, efficient and effective computing applications in the field are somewhat sparse. The study by [67] claims that there is a significant vacuum in computerized applications for the field of law which has resulted in an information crisis. The fact that legal vocabulary possesses words of mixed origin such as English and Latin has been raised as a reason for the difficulty of creating computing applications for the legal domain [51].

However, recently, there have been attempts to involve and build legal ontologies [50, 52] as well as attempts to calculate similarity measures in legal domain text [49] and build information retrieval systems thereof [51]. Given the popularity of knowledge embedding, a number of studies have also attempted to embed legal jargon in vector spaces [49, 68]. A more recent study by [69] uses discourse relations in an attempt to identify relationships among sentences in court case transcripts.

Social media is one of the most used domains for research in sentiment analysis due to the availability of plentiful data. Social media platforms usually contain opinions expressed by people on various topics including politics, sports, entertainment, and others. For instance, [70] states a research conducted in analyzing language in Twitter posts of millions of users, along with a method to automatically collect a corpus with positive and negative sentiments, where the authors have performed statistical linguistic analysis on the collected corpus and built a sentiment classification system for micro-blogging. They have used a Naive Bayes classifier that uses N-grams and part-of-speech tags as features to train the model. This method is not suitable for analyzing legal text

because of the inherent objectivity that needs to be preserved in law.

Sentiment classification is also known as *opinion mining* [40]. As such, the study on *Opinion Mining* in legal blogs [71] is closest implementation for this study that we have found in the literature. The *Lingpipe* toolkit, of which the sentiment annotation is based on a character-based language model, is used for the sentiment classification in the approach by [71]. Further, the data set used for evaluation is based on movie reviews, customer reviews, and MPQA corpus [72].

SentiWordNet [40, 41] classifies synsets in WordNet [42] to three classes; *negative*, *positive*, and *objective*. Synsets that do not contain opinionated content are assigned to the *objective* class while the Synsets that do contain opinionated content are assigned to the *subjective* which is then further classified into the two classes *negative* and *positive* depending on the sentiment it carries.

There have been numerous studies that were built upon SentiWordNet [40, 41] which attempts to classify sentiments of phrases and sentences. One such study by [43] proposes a methodology to perform opinion mining on movie reviews using support vector machine where some of the features were calculated using WordNet. This achieves an accuracy of 69.35% and claims that the inaccuracies in SentiWordNet feature calculations are caused by the SentiWordNet's reliance on glosses. [73] evaluates the SentiWordNet for identifying opposing opinion networks in forum discussion. The average SentiWordNet opinion score of words is considered to identify whether a user's expressed comment for a given post has either *for* or *against* relationship. The achieved accuracy using the SentiWordNet opinion score of words is 56%.

The method proposed by [44] provides an algorithm to identify the sentiment of a phrase or a sentence in a supervised manner using a deep learning model of the type Recursive Neural Tensor Network (RNN). It is claimed that this learning model has the capability to identify the sentiment considering the context of that word. A dataset which consists of movie reviews where each sentence in the data set was broken into phrases and each phrase is annotated by human judges were created for this study. The authors claim a testing accuracy of 80.7% in phrase level for a test set drawn from the same dataset. Further, the authors claim that the proposed model can be trained

over any domain by following the provided methodology. While, theoretically, it is possible, following this for legal domain in a practical implementation which covers a corpus which is both significant and sufficient is difficult. This claim is substantiated by referring the dataset of the original research [44] which utilized 215,154 manually annotated phrases (from 11,855 sentences) with over 5355 unique words. In comparison to this, the legal corpus used in our study has a vocabulary exceeding 17000 words. The difficulties are not merely of scale given that the linguistic complexity of legal jargon exceeds that of the average text corpus [48, 49, 51, 52].

It is observed that the Recursive Neural Tensor Network (RNTN) model by [44] shows better accuracy in sentiment classification compared to other models. However, the trained model being biased towards the movie reviews which it was trained on is a difficulty that needs to be overcome. For this purpose, several studies [45, 46] claim the process of *domain adaptation* to be a suitable solution. *Domain adaptation* is a sub-category of *Transfer Learning* [45]. While the generic process of transfer learning is defined as the process of "learning model is trained using data from a certain domain and tested with respect to a different domain" [45], the specific case of *domain adaptation* occurs when the Since the task is same in both source and target model. Given that both this study and the original study by [44] works on sentiment classifying, the transfer learning done in this study falls under the definition of domain adaptation [45]. Even though transfer learning is not very common in the NLP field, it is extensively used in other fields such as image classification [45, 47].

The aim of this study is also to build a sentiment classifier specific to the legal domain. But to prepare a manually labeled data set for training purpose is a costly process in terms of time and human effort. Therefore, a *Transfer Learning* approach is used to adapt the RNTN model [44] to the legal domain. When a learning model is trained using data from a certain domain and tested with respect to a different domain, it is called *Transfer Learning* approach [45]. Since the task is same in both source [44] and target model for legal domain, the task belongs to the subcategory called *Domain Adaptation* as mentioned in [45, 46]. Image classification [45, 47] is a field where transfer learning is vastly used.

Table 3.7: Sentiment Mapping

	Human annotation	**Stanford CoreNLP output**
Class 1	Negative	Very negative, negative
Class 2	Non-negative	Neutral, Positive, very positive

3.2.3 Methodology

First, it is required to pick the vocabulary from a corpus comprised of legal case transcripts. Then we input a set of words extracted from that corpus to the Recursive Neural Tensor Network (RNTN) model for sentiment annotation mentioned in [44]. We will refer to this model as the *source model*. After that, three human annotators check for words with deviated sentiment based on the classified classes. Using that identified set, we perform a transfer learning method to identify the sentiment of a given phrase in the legal domain.

In a domain like movie reviews, the sentiment of a statement can be either positive, negative, or neutral depending on whether the reviewer's opinion. But in a legal case, there is a plaintiff and a defendant. The plaintiff brings arguments against the defendant and the defendant representative tries to prove that the plaintiff's arguments are not valid. Therefore, the sentiment of a legal argument is defined to be categorized into two classes; *negative* and *non-negative*. In this approach, we use a model trained for movie reviews domain as the source model, in which the sentiment is classified into five classes. Hence, we have come up with a mapping as mentioned in Table 3.7.

3.2.3.1 Selecting the Vocabulary

Depending on the size of the corpus, availability of human annotators and the time, it might not be feasible to analyze and modify the sentiment of every word in the corpus. Therefore, it is required to select the vocabulary such that the end-model can correctly classify the sentiment of most of the phrases from the legal domain. In a nutshell, term frequencies for each word are taken, and based on that the vocabulary will be selected such that 95% of the total words are covered in the corpus.

First of all, the stop-words will be removed from the text. Stop-words are defined

as words which are occurring frequently but meaningless for information retrieval process‘[74]. A classical stop-word list known as the Van stop-list [75] is used for this task. Then the term frequency of each term is computed and ordered them in descending order of frequency. Then we define the cutoff term frequency such that total word list represents 95% of the total word elements in the corpus.

3.2.3.2 Assigning Sentiments for the Selected Vocabulary

The selected vocabulary is provided to the sentiment annotator RNTN model [44] as input. From the model, sentiment is classified into one of the five classes, namely, **very negative**, **negative**, **neutral**, **positive** and **very positive**. The RNTN model [44] was trained using movie reviews. However, our end goal is to identify sentiment in legal case transcripts. In the legal domain, the basic requirement of finding sentiment is to identify whether a given statement is against the plaintiff's claim or not. Therefore, we define two classes for sentiment: negative and non-negative.

Three human judges analyze the selected vocabulary and classify each word into the two classes depending on its sentiment separately and independently. If at least two judges agree, the given word's sentiment is assigned as the class those two judges agreed. For the same word, the output from the sentiment annotator [44] belongs to one of the five classes mentioned in the above subsection. In this approach, we map the output from RNTN model [44] to the two classes we defined as follows.

For a given word, if the two assigned sentiment values by the RNTN [44] model and human judges do not agree to the above mapping, we define that the RNTN model's output has deviated from its actual sentiment.

For example:

Sentence: *Sam is charged with a crime.*

RNTN model's output: positive

Human judges' annotation: negative

The word *charged* has several meanings depending on the context. As the RNTN model [44] was trained using movie reviews, the sentiment of the word *charged* is identified as positive. Although the sentiment of the term *crime* is recognized as negative, the sentiment of the whole sentence is output as positive. But in the legal

domain, *charged* refers to a formal accusation. Therefore, the sentiment for the above sentence should have been negative. From the selected vocabulary, all the words with deviated sentiment are identified and listed separately for the further processing.

3.2.3.3 The RNTN Model

From the previous subsection, we came across a situation where the sentiment values from the RNTN model mentioned in [44] does not match with the actual sentiment value because of the domain. And there are words like *insufficient*, which were not recognized by the model because that term was not included in the training data-set. One approach to solve that is to annotate the phrases extracted from legal case transcripts manually as [44] suggests. That will require a considerable amount of human effort and time. Instead of that, we can change the model such that the desired output can be obtained using the same trained RNTN model [44] without explicitly training using phrases in the legal domain. Hence, this method is called a transfer learning method.

In order to change the model, first it is required to understand the internals of the original RNTN [44] model. When a phrase is provided as input, first it generates a binary tree corresponding to the input in which each leaf node represents a single word. Each leaf node is represented as a vector with d-dimensions. The parent nodes are also d-dimensional vectors which are computed in the bottom-up fashion according to some function g. The function g is composed of a neural tensor layer. Through the training process, the neural tensor layer and the word vectors are adjusted to support the relevant sentiment value. The neural tensor layer corresponds to identify the sentiment according to the structure of words representing the phrase.

For Example:

phrase: *not guilty.*

sentiment: non-negative

Both words in the above phrase, have negative sentiment if we consider each of them individually. But the composition of those words has the structure of negating a negative sentiment term or phrase. Hence the phrase has a non-negative sentiment. If the input was a phrase like “very bad”, the neural tensor layer has the ability to identify that the term “very” increases the negativity in the sentiment. The hidden process is

same as in the above example.

3.2.3.4 Adjusting Word Vector Values in RNTN Model

The requirement of the system is to identify the sentiment of a given phrase. The proposed approach is not to modify the neural tensor layer completely. We simply substitute the word vector values of individual words which are having deviated sentiments between *Socher Model* and human annotation (See sections 3.2.3.2). The vectors for the words which were not in the vocabulary of the training set which was used to train the RNTN model should be instantiated. The vectors of the words which are not deviated (according to the definition provided in the preceding subsection 3.2.3.3) will remain the same.

As the words with deviated sentiments (provided by the *Socher Model*) in the vocabulary are already known, we initialize the vectors corresponding to the sentiment annotation for those words. Since the model is not trained explicitly, the vector initialization is done by substituting the vectors of words in which sentiment is not deviated comparing the *Socher Model* output and its actual sentiment. After the substitution is completed, we consider the part-of-speech tag. For that purpose, the part-of-speech tagger is used. The substitution of vectors is carried out as shown in Table 3.8.

The number of words which have deviated sentiments is a considerably lower amount compared to the selected vocabulary. The rest of the words' vectors representing sentiments are not changed in the modification process. The neural tensor layer also remains unchanged from the trained *Socher Model* using movie reviews [44]. When the vectors for words with deviated sentiments are initialized according to the part-of-speech tag as shown in Table 3.8, it is possible to make a fair assumption that when deciding the sentiment with the proposed implementation, it does not harm the structure corresponding to the linguistic features of English. Consider the sentence "*evidence is insufficient.*" as an example.

The term "*insufficient*" is not in the vocabulary of the *Socher Model* due to the limited vocabulary in training data set. Therefore, the *Socher Model* provides the sentiment of that word as neutral which indicates as a word with a deviated sentiment. Following the Table 3.8, the sentiment related vector is instantiated by substituting

Table 3.8: Substituted Word Vectors for words which should be deviated

POS Tag	Substituted word vector sentiment	
	non-negative	negative
JJ	wrong	natural
JJR	worse	natural
JJS	worst	natural
NN	failure	thing
NNS	politics	things
RB	insufficiently	naturally
RBR	insufficiently	naturally
RBS	insufficiently	naturally
VB	hate	do
VBZ	hates	does
VBP	hate	do
VBD	hated	did
VBN	bored	given
VBZ	ignoring	doing

the vector of **wrong** as the part-of-speech tag of **insufficient** is **JJ** [76]. Therefore the modified version of the RNTN model has the capability of identifying the sentiment of the above sentence as negative. The figure 3 shows how the sentiment is induced through the newly instantiated word vector.

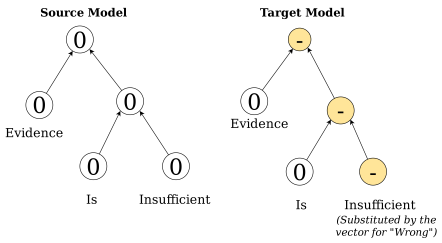

Fig. 3. Sentiment Prediction for a phrase with words not in source's vocabulary but in target's vocabulary

And there are scenarios where the term is in the vocabulary of the *Socher Model* but has a different sentiment compared to the legal domain. Consider the sentence "*Sam is charged with a crime*" which was mentioned in section 3.2.3.2.

In section 3.2.3.2, we have identified that the term *charged* denotes a different sentiment in legal domain compared to movie reviews. The source RNTN model outputs a positive sentiment for that given sentence as the term *charged* is identified

as having a positive sentiment according to movie reviews domain. And that term is the cause for having such an output from the source model. The figure 5 indicates how the change we introduced in the target model (in section 3.2.3.2) induce the correct sentiment up to the root level of the phrase. Therefore, the target model identifies the sentiment correctly for the given phrase.

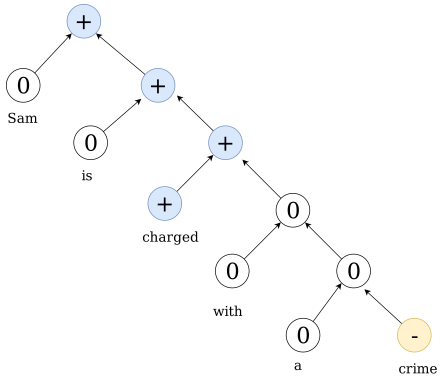

Fig. 4. Sentiment Prediction for a phrase with words having deviated sentiment in two domains - source model

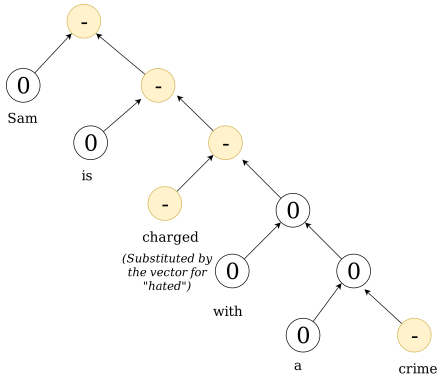

Fig. 5. Sentiment Prediction for a phrase with words having deviated sentiment in two domains - target model

To improve the recall in identifying phrases with negative sentiment, we have added another rule to the classification criteria. The source RNTN model (*Socher Model*) provides the score for each of the five classes such that all those five scores sum up to 1. If the negative sentiment class has the highest score, the sentiment label of the phrase will be *negative*. Otherwise, the phrase again can be classified as having a *negative* sentiment if the score for negative sentiment class is above 0.4. If those two conditions are not met, the phrase will be classified as having a *non-negative* sentiment. Section

3.2.5 provides observations and results regarding the improved criteria.

3.2.4 Further Proceedings

So far, the substitution happens according to the Tables 3.8. In order to preserve the context, we have considered the POS tag [77] and the sentiment of the unigrams in the corpus so far. But the model could be further improved by considering the cosine similarity [17] between two words during the substitution process. The improved approach for the substitution is as follows.

During the manual annotation process of unigrams, we have identified the unigrams and the sentiment of each unigram in the legal document corpus. Further we have listed down the unigrams in which the sentiment is different between two domains. Additionally, we collect the unigrams from the source model's corpus and the provided sentiment output for each unigram by the source model, which is either *negative* or *non-negative*. We refer to this list as "source model unigram-sentiment list" for the rest of this section.

During the automatic sentiment annotation process, we need to identify whether there are unigrams with deviated sentiment between two domains. Since the source model (which is trained using movie reviews) induces a wrong sentiment based on that word, it is required to substitute that vector as in the previous occasion. Once we identified such unigram with a deviated sentiment, we need to find the unigram with the maximum cosine similarity score and the same expected sentiment assignment from the "source model unigram-sentiment list". As similarity measure, we use the Wu & Palmer similarity measure [16].

3.2.5 Experiments

The proposed approach is based on transfer learning. Therefore, we needed to create a golden standard for identifying sentiments of phrases and sentences in the legal domain in order to evaluate the model. The phrases and sentences for the test data set are randomly picked from legal case transcripts based on the United States Supreme Court. During the selection process, we have selected an equal amount of phrases for both

classes according to the *Socher Model*. Each of these phrases and sentences is annotated by three human annotators. Since the classification process is binary, we pick the sentiment class for each test subject based on the maximum number of votes. In the end, we prepare the test data set containing nearly 1500 annotations to use in the evaluation process.

In the experiment, we compare the sentiment class picked by human judges and the modified RNTN model. As the baseline model, we use the source RNTN model (*Socher Model*) to check the impact caused by the proposed transfer learning approach. The acquired results from the baseline model is shown in Table 3.9 and results from the target model is shown in Table 3.10.

According to Table 3.9 and Table 3.10, there is a 10% improvement in identifying phrases with negative sentiment. The reason is that there are a lot of unknown words which are in the legal domain but not in movie reviews corpus. In addition, we have introduced new criteria based on a threshold for the score of negative class to improve the recall. Due to that reason, the precision in identifying phrases with a negative sentiment is 0.8441. But if we compare with the precision of the baseline model (*Socher Model*) for negative sentiment class is 0.7962 which is a lower value. Since the test dataset is not skewed a lot towards one class, it is fair to consider the accuracy of the system in predicting the sentiment for any given phrase. The baseline model shows the accuracy of 70.17% while the target model shows 76.80%. The improvement in accuracy is above 6%.

Table 3.9: Confusion Matrix for Results from the Baseline Model

Actual \ Predicted	Negative	Non-negative	*Total*
Negative	60.43%	39.57%	278
Non-negative	18.29%	81.71%	235
Total	211	301	513

The observed results in Table 3.9 and Table 3.10 show that there is a 6% improvement of the sentiment with respect to the baseline model. There are a few reasons behind the results. As we randomly selected phrases from the legal case transcripts

Table 3.10: Confusion Matrix for Results from the Improved Model

Actual \ Predicted	Negative	Non-negative	*Total*
Negative	70.14%	29.86%	278
Non-negative	15.32%	84.68%	235
Total	231	282	513

corpus, only 45% of the phrases actually contained the words where we had substituted the vector regarding sentiment. Therefore, the output for 55% of the phrases from the baseline model and the target model was the same. If we compare the output provided by the baseline model and the target model, output of 9.5% of the total phrases are different to each other. Therefore the difference between the two models is based on that 9.5% of the total phrases.

3.3 Shift-of-Perspective Identification within Legal Cases

3.3.1 Introduction to the study

Transcripts describing legal cases (court cases) carry a significant importance when it comes to the legal literature. The information presented in court case transcripts are used in different capacities such as evidence, arguments, and facts by legal officials in the process of constructing new legal cases [51]. Therefore, information extraction from court case transcripts can be considered as an area of significant importance, within the topic of **automatic information extraction** in the legal domain. In order to perform systematic information extraction from court case transcripts, a system should be able to interpret the meaning of a given text. In the process of interpreting the meaning of a text, understanding the context can be considered as a major requirement, especially in the legal literature.

Identifying how textual units are related to each other within a machine-readable text is an important task when it comes to interpreting the context. Humans are good at comparing two textual units to determine the way in which those two units are connected. Granting this ability to computers is a major discussion topic in the research related to areas of Natural Language Processing and Artificial Intelligence. A sentence

can be considered as a textual unit with significant importance in a text. Therefore, analysis of relationships between sentences can be useful to get a clear picture on the information flow within a text which is made up of a considerable number of sentences.

Similarly, identifying the types of relationships existing between sentences in court case transcripts can be used to identify the information flow within a legal case. Within a court case transcript, different types of relationships between sentences can be observed such as *elaboration* and *contradiction*. Pairs of sentences can be classified into two major groups based on whether the topics which are being discussed by the two sentences in the sentence pair is the same or not. In other words, the two sentences in a sentence pair may discuss the same topic or they may discuss completely different topics. Even if the two sentences are discussing the same topic, the opinions or views presented in the two sentences on the topic may be different. Consider the following sentence pair taken from *Lee v. United States* [55].

Example 4

- Sentence 4.1: *Applying the two-part test for ineffective assistance claims from Strickland v. Washington, 466 U. S. 668, the Sixth Circuit concluded that, while the Government conceded that Lee's counsel had performed deficiently, Lee could not show that he was prejudiced by his attorney's erroneous advice.*
- Sentence 4.2: *Lee has demonstrated that he was prejudiced by his counsel's erroneous advice.*

The above two sentences discuss whether a person named Lee was able to convince that he was prejudiced by his attorney's advice or not. While the first sentence says that *Lee could not show that he was prejudiced by his attorney's advice*, the second sentence contradicts the first sentence by saying that *Lee has demonstrated that he was prejudiced by his counsel's erroneous advice*. Thus, the two sentences provide different opinions on the same topic. Contradiction is not a necessary condition in order to classify a pair of sentences as providing different opinions on the same topic. For example, consider Example 2 which consists of two adjacent sentences which are also taken from *Lee v. United States* [55].

It can be seen that both sentences in this example discuss the topic – the deportation of a person named Lee. Though the two sentences here do not provide contradictory

Example 5

- Sentence 5.1: *Although he has lived in this country for most of his life, Lee is not a United States citizen, and he feared that a criminal conviction might affect his status as a lawful permanent resident.*
- Sentence 5.2: *His attorney assured him there was nothing to worry about–the Government would not deport him if he pleaded guilty.*

information, they provide two different viewpoints regarding the same topic. It can be seen that the opinions of Lee and his attorney on the possibility of Lee being deported is different. Therefore, when discussing sentences with different opinions on the same topic, not only the sentences providing contradictory information but also the sentences providing multiple viewpoints on the same discussion topic should also be considered. In each of the above two examples, Sentence 1 comes before Sentence 2. From this point onwards, the first sentence in a sentence pair will be referred to as the *Target Sentence* and the second sentence as the *Source Sentence*.

An important observation which can be made by considering Example 2 is that the identification of the shift in the viewpoint in that particular occasion is not straightforward. This implicit nature makes the task of identifying sentences which provides different opinions on the same discussion topic even more challenging. At the same time, it can be considered a vital task due to its potential to enhance the capabilities of Information Extraction from Legal Text by facilitating automatic detection of counter-arguments, identification of the stance of a particular party in a court case and to discover multiple viewpoints to analyze or evaluate a particular legal situation.

Hence, the objective of this study is to identify sentences which have different perspectives on the same discussion topic in a given court case. For this study, United States court case transcripts obtained from FindLaw[2] were used. The next section provides details on the previous work which are related to our study. Section III describes the methodology followed in this study while the outcomes of the study are discussed in Section IV. Finally we conclude our discussion in Section V.

[2]https://caselaw.findlaw.com/

3.3.2 Literature Review - Shift-of-Perspective Identification within Court Cases

Computing applications which can be considered to be both efficient and effective are scarce due to the challenges in handling legal jargon[48, 52, 67]. This, in turn, has caused an information crisis [67]. The nature of legal documents employing a vocabulary of mixed origin ranging from Latin to English has been put forward as a reasoning for difficulties of building computing applications for the legal domain [51].

Regardless, there have been some recent attempts to circumvent these problems in the legal domain including information organization [48, 50, 52], information extraction [49] and information retrieval [51]. Further, in the information extraction domain, the study by Gamage et al [78] attempted to build a sentiment annotator for the legal domain and the study by Ratnayaka et al [79] attempted to identify relationships among sentences in court case transcripts.

Discovering situations where two sentences are providing different opinions on the same topic or entity is an important part when it comes to identifying relationships among sentences [79]. Contradiction is a sufficient but not a necessary condition in this regard. The study [31] is focused on finding contradictions in text related to the real world context. In an attempt to define contradiction, the same study[31] claims that “contradiction occurs when two sentences are extremely unlikely to be true simultaneously” and the study [32] also agrees on that definition. However, the study [31] also demonstrates that two sentences can be contradictory while being true simultaneously. These characteristics of contradiction make the process of detecting contradiction relationships more complex.

Furthermore, [31] elaborates on the fact that "for texts to be contradictory, they must involve the same event". Involving the same event is not a necessary condition to be met in order to determine that two sentences are contradictory in the legal domain. The reason is, in order to become contradictory, two textual units can elaborate not only on the same event but also on the same entity. For example, if one sentence in a sentence pair is saying that a person is a United States citizen while the other sentence is saying that very same person is **not** a United States citizen, it is obvious that the two sentences are providing contradictory information. Here, the contradictory information

is upon a person which can be considered as an entity. Therefore, it is more reasonable to consider that in order to be contradictory, texts must elaborate on the same topic.

In order to detect contradiction, different features based on the properties of text have been considered in the previous studies [31, 32]. Polarity features and Numeric Mismatches are such commonly used features. The study [31] empirically claims that the precision of detecting contradiction falls when numeric mismatches are considered.

The structures of the texts also play a vital role when it comes to contradiction detection [31, 32]. Analysis of text structure is helpful in identifying the common entity or event on which the contradiction is occurring. When the structure of a given sentence is considered, the subject-object relationship plays an important role [14, 80]. Analysis of Typed Dependency Graphs [81] is another useful approach to understand the structure of a particular text and to obtain necessary information.

Polarities of the sentences in relation to the sentiments can also play a vital role when it comes to identification of sentence pairs which provide different opinions on the same topic. It can be observed the seminal RNTN (Recursive Neural Tensor Network) model [44] which is trained on movie reviews is used in many recent studies [77, 82] which perform sentiment analysis.The trained RNTN model [44] has a bias towards the movie review text[78]. In order to overcome the problem, the study [78] has proposed a methodology to develop a sentiment annotator for the legal domain using transfer learning and has obtained 6% increase in accuracy over the original model[44] within the legal domain.

The study [10] introduces a new algorithm to calculate the oppositeness of triples that can be extracted from microRNA research paper abstracts using open information extraction. As the study proposes a mechanism to detect inconsistencies within paragraphs, we see it as one potential methodology which can be adapted to detect the *Shift-in-View* relationship between sentences. However, as the above-mentioned study [10] specifically focuses on discovering inconsistencies in the medical domain it is needed to adopt the proposed methodology to the legal domain in order to detect shift-in-perspectives in court case transcripts. From this point onward in this paper, we will refer to the study [10] as the **PubMed Study**.

In the study [25], discourse relations between sentences have been used to generate clusters of similar sentences within texts. A Support Vector Machine model is used in this study[25] to determine the relationships existing between sentences. In the process of Multi-Class classification performed using the SVM Model, [25] has defined a class named *Change of Topic* which combines the *Contradiction* and *Change of Perspective* relations as defined in Cross Document Structure Theory (CST) [22]. The study[25] has obtained lower results for *Change of Topic* than other relationship types claims that average results are due to lack of significant features which could properly detect *Contradiction* and *Change of Perspective*. CST relations and data from CST bank have also been used to train an SVM model in the study **Sentence Relationship Identifier** [79] in order to predict relationships between sentences in the legal domain. Though the study has done improvements to the features in [25] and introduced new features which suit the legal domain, the results obtained in relation to the *Contradiction* and *Change of Perspective* relationships as defined in CST [22] is very low. One possible reason is that the CST Bank[65] data set is made up of sentences from newspaper articles, where the structural and linguistic features may differ from that in the court case transcripts, especially when it comes to relationships such as *Contradiction* and *Change of Perspective*.

It can be seen that the relationship type *Shift-in-View* defined in **Sentence Relationship Identifier** aligns with the relationship type that is being discussed in this study.
.

As shown in Table 3.1, the *Shift-in-View* relationship includes both *Contradiction* and *Change of Perspective* relationships as defined in CST [22]. *Elaboration*, *Redundancy*, *Shift-in-View* or *Citation* relationships defined in the study[79] suggest that a sentence pair is discussing the same topic while *No Relation* suggests that the two sentences are discussing completely different topics. It has been stated that **Sentence Relationship Identifier** is able to detect situations where the discussion topic is changed with a considerable accuracy [79]. However, it is also stated that the proposed methodology is not able to detect situations where two sentences provide different opinions on the same topic.The results obtained in this study [79] are shown in Table

3.3.

It is clear that the machine learning model inside the **Sentence Relationship Identifier** is not able to detect *Shift-in-View* relationship. However, Table 3.3 shows that the sentences pairs having *Shift-in-View* relationships are detected as *Elaboration*. It can be considered as a positive aspect, as *Elaboration* suggest that both sentences are elaborating on the same topic, which is a necessary condition when detecting sentences providing different perspectives on the same topic or entity as described in other studies [31, 32] too.

3.3.3 Methodology

3.3.4 Identifying Sentence Pairs where Both Sentences Discuss the Same Topic

It is needed to identify whether the two sentences are discussing the same topic in detecting sentence pairs which provide different opinions on the same topic. Therefore, as the first step, we implemented the **Sentence Relationship Identifier** as it is successful in identifying whether two sentences are discussing on the same topic or not [79].

According to the study [79], *Elaboration*, *Redundancy*,*Citation* and *Shift-in-View* relationships occur when both sentences discuss the same topic. *Shift-in-View* occurs over *Elaboration* when the two sentences provide different opinions on the same topic.

We only consider sentence pairs which are detected as having *Elaboration* relationship type were considered in order to identify whether *Shift-in-View* relationship is present. Though *Redundancy* and *Citation* relationship types also suggest that two sentences are discussing the same topic, the sentence pairs detected with those relationship types are not considered. As the *Redundancy* relationship suggests that two sentences provide similar information, there is no possibility of having different perspectives. In *Citation* relationship, one sentence provides evidence or references to confirm the details presented in the other sentence. Thus, it is not probable to occur a situation where two sentences provide different perspectives on the same topic.

However, If the machine learning model described in the study [79] detect a pair of sentences as having *Shift-in-View* relationship, such a pair will be detected as a sentence pair which provides different opinions on the same topic. Confirming the observations

of the study [79] the **Sentence Relationship Identifier** did not identify any pair of sentences as having *Shift-in-View* relationship.

3.3.5 Filtering Sentences using Transition Words and Phrases

There are *Transition Words* or *Transition Phrases* which suggest that the *Source Sentence* of a sentence pair is elaborating or building up on the *Target Sentence*. In the *Source Sentence* of Example 3 (which was taken from *Lee v. United States* [55]), the transition word *"Accordingly"* implies that the *Source Sentence* is developing while agreeing with the *Target Sentence*.

Example 6

- Sentence 6.1: *Lee's claim that he would not have accepted a plea had he known it would lead to deportation is backed by substantial and uncontroverted evidence.*
- Sentence 6.2: *Accordingly we conclude Lee has demonstrated a "reasonable probability that, but for [his] counsel's errors, he would not have pleaded guilty and would have insisted on going to trial"*

Therefore, when such a *Transition Word* or *Transition Phrase* is present in the *Source Sentence*, such a sentence pair will be considered as having the *Elaboration* relationship. Therefore, such sentence pairs are not processed further for detecting the *Shift-in-View* relationship type. We have implemented this mechanism as a way to increase the precision of the *Shift-in-View* detection approaches. Given below are some Transition Words and Transition Phrases we used.

Transition Words: *thus*, *accordingly*. *therefore*

Transition Phrases: *as a result*, *in such cases*, *because of that*, *in conclusion*, *according to that*

3.3.6 Use of Coreferencing

Prior to checking for linguistic features which imply that the sentence pair is showing *Shift-in-View* relationship, co-referencing is performed on the sentence pair. For coreferencing, Stanford CoreNLP CorefAnnotator ("coref") [66] was used. The co-referencing provides a better picture when the same entities are being mentioned in the

two sentences using different names[79].

3.3.7 Analyzing Relationships between Verbs

The first linguistic approach to detect deviations in opinions expressed in sentences regarding a particular topic is based on verb comparison. Under this approach, verbs are compared using the negation relationship and using adverbial modifiers.

In this approach, subject-object pairs in the *Target Sentence* is compared with that of the *Source Sentence*. If the subject or object in one sentence is present in the other sentence, the verbs in the sentences are considered. Here, we do not consider verbs which are lemmatized into *"be", "do"* in order to focus only on effective verbs. The Stanford CoreNLP POS Tagger (“pos") [83] was used in identifying verbs in sentences. After extracting the verbs in two sentences, each verb in *Target sentence* is compared with each verb in *Source sentence* to detect verb pairs with similar meaning.

3.3.8 Determining Verbs which Convey Similar Meanings

In order to convey a similar meaning, it is not necessary that both verbs are the same. Also, when semantic similarity measures between two verbs are considered, it can be observed that there are verb pairs which have very similar meanings but different semantic similarity scores. For example, if the lemmatized forms of verbs in Example 1 are considered, it can be observed that the verb *demonstrate* in the *Target sentence* and verb *show* in the *source sentence* have similar meanings. Confirming that observation further, a Wu-Palmer similarity score of 1.0 can be obtained for that verb pair. When the lemmatized forms of verbs in two sentences in Example 2 are considered, it can be observed that the word "fear" in the *Target sentence* and "worry" in *Source sentence* are two verbs with similar meanings. However, the Wu-Palmer semantic similarity score between verbs *fear* and *worry* is 0.889. Therefore, it is needed to determine an acceptable threshold based on semantic similarity scores in order to identify verbs with similar meanings.

In order to determine this threshold, we first took 1000 verb pairs whose Wu-Palmer similarity scores are greater than 0.75. As our objective is to identify pairs of verbs with

similar meanings, it could be observed that a Wu-Palmer score of 0.75 was a reasonable lower bound as per the precision values. We annotated those 1000 pairs of verbs based on whether a given verb pair actually has two verbs with similar meanings or not. Then we gradually incremented the threshold by 0.1 from 0.75 to 0.95 and observed the precision and recall values as shown in Table 3.11.

Table 3.11: Results Comparison for Different Wu-Palmer, Jiang-Conrath, and Lin Score Thresholds

Score	Wu-Palmer			Lin		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
0.75	45.65%	100.00%	62.68%	57.29%	72.37%	**63.95%**
0.80	51.39%	77.19%	61.70%	60.39%	67.54%	**63.77%**
0.85	54.59%	69.08%	60.99%	64.76%	62.06%	**63.38%**
0.86	59.34%	59.21%	59.28%	67.15%	60.96%	**63.91%**
0.90	64.49%	49.78%	56.19%	70.40%	53.73%	**60.95%**

In addition to Wu-Palmer scores, we performed the same experiment on the verb pairs using all the eight semantic similarity measures available in Wordnet[84]. It was observed that Jiang-Conrath [18] and Lin [85] are the two measures which provides reasonable accuracy in addition to Wu-Palmer semantic similarity[16]. The results from these experiments are shown in Table 3.11 and in Fig.6. It could be observed that **Lin** outperforms other two measures when F-Measures are considered. It can be seen that 0.75 is the **Lin** score which has the highest F-Measure. But, it is due to considerably high recall and undesirably low precision values. As our intention is to maintain a proper balance between precision and recall, **Lin** Score of 0.86 is selected as the threshold to detect verb pairs with similar meaning. 0.86 is the **Lin** Score with the second highest F-Measure.

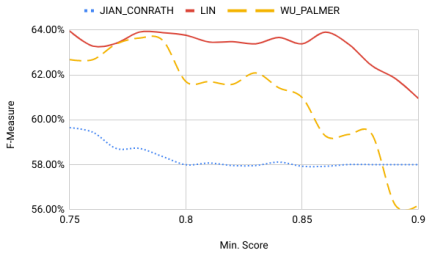

Fig. 6. Variation of F-Measures with regard to Different Similarity Measures

3.3.9 Detecting Shift-in-View Relationships by Comparing Properties Related to Identified Verbs

3.3.10 Negation on Verbs

Usage of negation relationship is a popular approach when it comes to detecting inconsistencies and contradictions in text [10, 31, 86]. In this study, we checked for the negation relationship within verbs in verb pairs identified using the method proposed in the section D. If one verb is detected as being negated while the other verb is not being negated, the sentence pair is considered as having *Shift-in-View* relationship. Stanford CoreNLP dependency parser was used to detect the negation by identifying occurrences of the "neg" tag as described in "Stanford typed dependencies manual" [87].

3.3.11 Using Adverbial Modifiers to Detect Shifts-In-View

Another approach to detecting different viewpoints on the same subjects or entities can be formulated by considering adverbial modifiers. If the adverbial modifiers related to two verbs with similar meanings give opposite or contradictory meanings, that means the viewpoints on how that task was performed is different. Therefore, the adverbial modifiers related to the verbs in verb pairs identified in the section D was considered. We classified adverbial modifiers in to three main classes shown in Table 3.12. Within each class, there exists a positive subclass and a negative subclass. In the table, we have shown the positive sub classes with unshaded rows while the negative sub classes are shown with shaded rows. After defining major classes into which adverbial modifiers can be classified, lists containing adverbs related to each class were created. Table 3.12 further contains examples of adverbs related to each type. This table does not include

all the adverbs we are maintaining in the lists.

Table 3.12: Adverbial Modifiers

Type Class	Type Name	Modifiers
Frequency	more frequent	always, often, repeatedly, regularly
	less frequent	accidentally, never, not, less, loosely, rarely, sometimes
Tone	amplifiers	so, well, really, literally , simply, for sure, completely, absolutely, heartily
	down toners	kind of, sort of, mildly, to some extent, almost, all but
Manner	positive manner	elegantly,beautifully,confidently
	negative manner	lazily, ugly,faint heartedly

If adverbial modifiers connected to both verbs in a verb pair with similar meaning belong to same Adverbial modifier type, but with opposite polarities (one positive and one negative), it can be identified that the two sentences provide different views in relation to the entities that are connected by those verbs.

3.3.12 Discovering Inconsistencies among Triples

Following the methodology presented in the PubMed study [10], a legal term dictionary was constructed to be served as a *Semantic Lexicon* for the system. 200+ court case transcripts were used to extract words for the process. Then a word list consisting 17,000+ unique words were developed by removing stop words. A TF-IDF algorithm [88] based method is used to calculate a value for each term in the dictionary.

$$TermValue = \frac{\sum_{i=1}^{casecount} \frac{f_{t,d}}{termcount}}{D.F} \tag{5}$$

Raw count ($f_{t,d}$) for each term is taken, considering each court case transcript as a document. Term frequency value for a term is calculated by dividing the raw term count by the total number of terms in the case. Term frequency value for each case is added together and the result value is divided from the document frequency ($D.F$), to calculate the value for a term in the dictionary. Then all the term values are normalized

according to the equation 18.

$$NormalizedTV = \frac{(TV - TV_{min}) * (1 - TV_{min})}{TV_{max} - TV_{min}} + TV_{min} \quad (6)$$

Here TV_{min} and TV_{max} represent the minimum and maximum values of the term values respectively. This normalized value is used to be served as the semantic weight for the system.

First, coreference resolving is done on the sentence pairs using the Stanford CoreNLP CorefAnnotator (âĂIJcorefâĂİ) [66] and the pairs with *Transition Words and Phrases* are filtered out. Then OLLIE [80], open information extraction system, is used to extract triples, in (*Subject; Relationship; Object*) format, from sentences. When comparing two sentences, for the *Shift-in-View* relationship, only triple pairs with same subject or object are considered, as the *Shift-in-View* relationship talks about different perspectives on the same topic or entity. The stop words removed relationship strings of a triple pair are then compared with each other word by word. The comparison is performed in three ways.

1. Identical words
2. Identical words with one word negated with “not"
3. Different words

In our study we consider the negation of words with similar meanings (**Lin** score above 0.86) instead of considering only the words which are exactly the same. Then, an oppositeness value is obtained for each sentence pair by comparing the triples following the algorithmic approach proposed in the PubMed Study [10]. A threshold based on the oppositeness values is introduced empirically to select sentence pairs which have the *Shift-in-View* relationship.

3.3.13 Sentiment-based Approach

Though valuable information can be obtained by analyzing the sentiment of a sentence, the sentiment of a sentence alone hardly gives any details on the topics which are being

discussed within a sentence and on the viewpoint in which the sentence is describing the topic. It is known that the two sentences which are being compared to detect shifts in views discuss on the same topic as only the sentence pairs with "Elaboration" relationship is considered. But, when the sentences in court case transcripts are considered, even if the sentiments of two sentences which elaborate on the same discussion topic is different, it can not be concluded that the two sentences are providing different opinions on the topic.

The reason is that the person entities which are described in a sentence and connected with the sentiment of the sentence have a significant impact on the topic which is being discussed. For example, consider two sentences which elaborate on the same discussion topic and having opposite sentiments. If the sentiment of the sentence with negative sentiment is connected with the *proposition* party while the sentiment of sentence with positive sentiment is connected with the *opposition* party, it might be the case where both sentences are conveying opinions which are beneficial for the *opposition* party in relation to the topic which is being discussed.

The problem becomes even more complex when the sentence is made up of several sub-sentences because each sub-sentence may have a "Subject" of its own. Therefore, when using the sentiment based approach to detect "Shift-in-View" relationship, we consider only the sentence pairs in which each sentence has only one explicit subject. If the subjects in both sentences are the same in such a sentence pair, it can be concluded that two sentences are elaborating on the same topic in relation to the same subject. Then, it is checked whether the two sentences are providing sentiments with opposite polarities. If one sentence provides negative sentiment and other provides positive sentiment while discussing the same topic in relation to the same subject, it can be concluded that the probability of two sentences giving different perspectives on the same topic is very significant.

In this approach, the sentences which are composed with subordinate clauses are first split using those clauses. When the sentence is split using a subordinating conjunction, that subordinate clause can be identified as another sentence entity. Throughout this section, we will refer the subordinate clause as *inner sentence* and the main clause

will be referred to as *outer sentence*. After the sentence is annotated using Stanford CoreNLP Constituency Parser [81] , the splitting happens by identifying associated terms with *SBAR* tag.

The proposed approach is based on analyzing the sentiment of this inner sentence to identify if there is a *Shift-in-View* relation between a sentence pair. If we consider the Example 4, The phrases *"Lee cannot convince the court that a decision to reject the plea bargain"*, and *"he can establish prejudice under Hill"* are the inner sentences. The outer sentences are *"The government argues"*, and *"Lee, on the other hand, argues"*.

Example 7

- Sentence 7.1: *The Government argues that Lee cannot "convince the court that a decision to reject the plea bargain.*
- Sentence 7.2: *Lee, on the other hand, argues that he can establish prejudice under Hill.*

If we consider the sentence pair mentioned in Example 4, both the inner sentences' subject is Lee. The phrase *"Lee cannot convince the court that a decision to reject the plea bargain"* is having a negative sentiment while the other inner sentence *"Lee can establish prejudice under Hill"* denotes a positive sentiment. Both the outer sentences are having neutral sentiment. Therefore, it can be observed that there is a Shift-in-View regarding the subject Lee.

3.3.14 Experiments and Results

First, the proposed *Shift-in-View* detection system was combined with the **Sentence Relationship Identifier**. Then, sentence pairs from court case transcripts were extracted from FindLaw. Next, the extracted sentence pairs were input into the combined system. Input sentence pairs are first processed inside **Sentence Relationship Identifier**. The sentence pairs which are identified as having *Elaboration* by **Sentence Relationship Identifier** were further processed in order to detect whether there is *Shift-in-View* relationship using the approaches mentioned under Section III.

Each sentence pair which was considered for evaluation was annotated by human judges who were trained on the relationship types which are defined in the **Sentence**

Relationship Identifier study [79]. Each sentence pair was first annotated by two human judges. If both judges are not agreeing upon a same relationship type for a particular sentence pair, such sentence pair was annotated by an additional human judge. When evaluating the results, we considered only the sentence pairs which were agreed by at least two human judges to have the same relationship type.

Each of the three major approaches which were used to detect *Shift-in-View* relationship were evaluated separately as shown in Table 3.13.

Table 3.13: Results Comparison of Approaches used to detect Shift-in-View

Approach	No. of Sentence pairs	Precision
Verb Relationships	46	0.609
Sentiment Polarities	230	0.382
Inconsistencies between triples	95	0.273

According to the Table 3.13, it can be seen that the precision which could be obtained from analyzing relationships between verbs is around 0.6. As mentioned earlier we have selected the **Lin** semantic similarity score of 0.86 as the threshold to identify verbs with similar meaning after analyzing different semantic similarity measures. The precision of identifying verbs with 0.86 **Lin** score is 0.67. Thus, it can be seen that there is a potential to improve the precision of detecting *Shift-in-View* relationships using relationships between verbs by developing a semantic similarity measure which is more accurate in identifying verbs with similar meanings for the legal domain.

Using the sentiment based model, the achieved precision is 0.38. There are few possible reasons behind this observation. The study on the sentiment annotator model[78] used in this case, states that the accuracy of the model is 76%. The study says that the errors present in its parent model [44] can be propagated to the target model[78]. The paper on the source model[44] which is based on recursive neural tensor network, shows that the accuracy is reduced down to 0.5 when the n-gram length of a phrase increases (n>10). As most of the sentences in court case transcripts are reasonably lengthier, there is a potential that the proposed sentiment based approach used for the identification of *Shift-in-View* is affected by the above mentioned error.

Only a precision of 0.27 could be observed in the approach which considers inconsistencies using triples as proposed in **PubMed Study**. The following reasons may have contributed to the poor performances of that approach. From the 2150 sentence pairs which were considered, oppositeness values were not calculated for 1570 pairs. Containing at least one sentence within a sentence pair in which the triples could not be extracted by OLLIE[80] is a major reason for not having an oppositeness value. Even if the triples are extracted from both sentences, if there is no matching between either subjects or objects of the two sentences, an oppositeness value will not be calculated for a sentence pair.

Evaluation results demonstrate that analysis of relationships between verbs in two sentences as the only approach which performs the task of detecting *Shift-in-View* relationships with a precision more than 0.5. Many studies convince the difficulty of detecting contradiction and change of perspective relationships over other relationship types that can be observed between sentences[25, 31, 79]. The study [31] also claims the difficulty of generalizing contradiction detection approaches. When considering these facts it can be considered that the results obtained via analyzing verb relationships are satisfactory. Therefore, we combined only that approach with the **Sentence Relationship Identifier** and evaluated the overall system made up by combining *Shift-in-View* detection with **Sentence Relationship Identifier** as shown in Table 3.14. The results were obtained using 200 annotated sentence pairs.

Table 3.14: Results Obtained from Sentence Pairs in which At least Two Judges Agree

Discourse Class	Precision	Recall	F-Measure
Elaboration	0.938	0.930	0.933
No Relation	0.843	0.895	0.868
Citation	1.000	0.971	0.985
Shift-in-View	0.688	0.423	0.524

According to the Table 3.14, it can be seen that there is a significance improvement, especially in relation to the *Shift-in-View* relationship type when compared with the results in the study[79] as given in Table 3.4.

3.4 Extracting Argumentative Sentences from Court Case transcripts

3.4.1 Introduction to the study

U.S. court case transcripts extracted from CaseLaw-FindLaw site [2] contain lengthy description on the way in which the case has evolved. A single case description consists of several main parts.

1. Summary of the case
2. Opinion of the Court
3. Concurring Opinions
4. Dissenting Opinions

First, a summary of the case which presents an overview of the case, main argument and the decision of the court, can be found. Then, the Opinion of the Court brings out the decision of the majority of the judges with the facts and arguments supporting the decision. If there are Concurring and Dissenting Opinions, they are presented after the Opinion of the Court. (Concurring Opinion is where one or more judges who agree with the decision of the court, but state different or additional reasons for the decision. Dissenting Opinion is where one or more judges disagree with the Opinion of the Court and bring out reasons for the disagreement.)

These descriptions contains valuable statements presented in the court in relation to the legal scenario. These statements can be classified as arguments and non-arguments. An argument can further be a premise or a conclusion. Facts, background information and court's opinions can be considered as non-arguments. Consider the statements in Example 8 taken from *Lee v. United States* [55].

Therefore identifying argumentative and non-argumentative sentences can be considered as a prominent task, in better representation of information in court case transcripts.

Example 8

- Argument: *Lee contends that he can make this showing because he never would have accepted a guilty plea had he known the result would be deportation.*
- Fact: *Petitioner Jae Lee moved to the United States from South Korea with his parents when he was 13.*
- Court's Opinion: *The District Court, however, denied relief, and the Sixth Circuit affirmed.*

3.4.2 Literature Review - Extracting Argumentative Sentences from Court Case transcripts

Detecting arguments, which consists of premises and conclusions, has always been a challenging and hard Natural Language Processing task. Various researches have been carried out about argument corpora and on automatic argument extraction, in both legal and non-legal domains.

AraucariaDB [35, 36] is a well known database of arguments from various sources. It also consists of a software tool for diagramming and representing arguments. In the study [35], Reed and Rowe, introducing *Araucaria* tool, brings out that arguments can be graphically represented in a tree. The tree can be drawn where one or more premises are branches which converge at a node with a conclusion. Arguments in *AraucariaDB* are manually annotated and marked up in an XML-based format, AML (*Argument Markup Language*). A small sample of argument trees are available in this database which also lacks the context information and criteria that the argument is based on. So it is uncertain what we can infer about the patterns which may appear in those trees.

The study "Approaches to Text Mining Arguments from Legal Cases" [34] by Wyner, et al. brings out an extensive background research on the literature of argumentation and argument extraction with an analysis of various argument corpora. This study also describes how legal arguments can be extracted, using a Context-Free Grammar. They have analyzed legal cases from European Court of Human Rights (ECHR) and identified legal argument construction patterns which can be found in premises and conclusions. These patterns have various clauses and verbs which are specifically identified for ECHR cases. So that makes theses structures very rigid and impossible to use for US court cases.

Two studies [37, 38] on automatic legal argument detection were also carried out using ECHR cases. In the study "Automatic Detection of Arguments in Legal Texts" [37] Moens, et al. present argument detection as a sentence classification problem between arguments and non-arguments where a classifier is trained on a set of manually annotated argumentative sentences. They have extracted different features involving lexical, syntactic, semantic and discourse properties of the texts, considering sentences in isolation.

In the "Study on Sentence Relations in the Automatic Detection of Argumentation in Legal Cases" [38] Mochales and Moens, further extending the research in argument extraction, point out that arguments are always formed by premises and conclusions. As such, they have determined argument extraction as a sentence classification problem among premises, conclusions, and non-arguments. Furthermore, they have improved the feature set used in [37] by including features that refer to the content of a window of following and preceding sentences.

To the best of our knowledge, there have been no research carried out on argument extraction from US court case transcripts. All these previous studies have used ECHR cases as their corpus which has a significantly different reporting structure than US court cases. The rules that are described in the study [34] are applicable for extracting argumentative sentences from US court cases, because of their rigid nature. After evaluating the studies [34, 37, 38], it was decided to follow a linguistic approach to identify arguments in US court cases, as we lack an annotated corpus of arguments and non-arguments to follow the machine learning approaches.

3.4.3 Methodology

A linguistic rule-based approach is followed to extract arguments. With the consultation of a legal expert, we determined structures that can be used to identify argumentative sentences in court case transcripts.

3.4.3.1 Linguistically identifying arguments using verbs

At first, words like *argue, agree, conclude, rejected, contest, contend, consider,*

testify, concede, claim, affirm were considered to identify legal arguments.

As examples:

1. A **claim** of ineffective assistance of counsel will often involve a claim of attorney error "during the course of a legal proceeding"–for example, that counsel failed to raise an objection at trial or to present an argument on appeal.

2. Lee **contends** that he can make this showing because he never would have accepted a guilty plea had he known the result would be deportation.

3. In post conviction proceedings, they **argued** that seven specific pieces of withheld evidence were both favorable to the defense and material to their guilt under Brady v. Maryland, 373 U. S. 83.

4. The D. C. Superior Court **rejected** petitioners' Brady **claims**, finding that the withheld evidence was not material. The D. C. Court of Appeals **affirmed**.

Here, sentence 1 and 4 cannot be considered as arguments. Sentence 1 represents an opinion and sentence 4 brings out the decision of the court. By observing the selected sentences we refined the word list and decided to only consider verbs to identify arguments. Lemmatized form of verbs in a sentence is extracted using Stanford POS Tagger[83] and then compared with the predefined list of verbs to check whether the sentence brings out an argument.

3.4.3.2 Citation-based argument extraction

In a legal case, there are statements with citations. And those citations links to previous cases, in which the judgments have already been finalized. And those statements come under case law category. If a statement is having a citation, it means that the statement is taken from a previous case and the same statement applies to current legal case as well. Therefore the lawyers can present the same argument in other legal cases to prove his point. Consider the following example,

- The decision whether to plead guilty also involves assessing the respective consequences of a conviction after trial and by plea. **See INS v. St. Cyr, 533 U. S. 289, 322-323.**

- But in this case counsel's "deficient performance arguably led not to a judicial proceeding of disputed reliability, but rather to the forfeiture of a proceeding itself." **Flores-Ortega, 528 U. S., at 483.**

This examples are taken from the court case *Jae Lee v. United States* [55]. The first statement links to INS v. St. Cyr, 533 U. S. which means that it is taken from the cited case. And the same statement can be presented in any other case if the statement is appropriate under the conditions of the legal case. We have taken a rule-based approach to identify these kind of arguments.

3.4.4 Experiments and Results

First, the sentences were extracted from CaseLaw-FindLaw [2]. Then, the sentences were input into the citation detection system and verb-based argument extraction system separately. After that, the detected sentences were annotated by human judges for argumentative sentences. Table 3.15 shows the results evaluation of two approaches.

Table 3.15: Results Comparison of Approaches used to detect argumentative sentences

Approach	No. of Detected Sentences	Precision
Argumentative verb based	77	64.93 %
Citation based	93	90.32 %

3.5 Party Identification

This sub-task was focused on classifying the arguments presented in a court case transcript as to whether they are in favor of the plaintiff or the defendant. For example, let's consider the case called "*Jae Lee v. United States (2017)*".

We tried the following approach to tackle this problem. First, we used a sentiment-based method for a shallow classification of the arguments into the two categories (whether the arguments were in favor of the plaintiff or the defendant). Then, we used graph structure consisting of nodes and edges to further refine the classification.

We could not follow through with this part of the research due to the following reasons.

- The problem is more of a Natural Language Understanding task than a Natural Language Processing task. It is a difficult task even for a human to do this classification as entire case needs to be read and understood before determining if a given argument is in favor of the plaintiff or the defendant.
- The unavailability of annotated data to try machine learning approaches.
- It is extremely difficult to annotate data as one has to read an entire case and understand everything before determining if a given sentence is in favor of the plaintiff or the defendant.

Therefore, the completion of this part of the research was infeasible given the time constraints of the project.

3.6 Dataset

The manually annotated dataset which was created for our evaluation purposes can be used in future studies for training and testing activities. Therefore, the dataset can be considered as one of the major research contributions of this study. Each annotation in the dataset was annotated at least by two human judges. If there was a disagreement between the two human judges regarding an annotation, that annotation was annotated by a third human judge and the majority vote was taken. The dataset includes,

1. **250 Sentence Pairs** classified on the relationship between two sentences.
2. **500 Phrases** classified on their sentiments.
3. **1000 Verb Pairs** classified on the similarity of meanings conveyed by the two verbs.
4. **170 Sentences** classified as Argumentative or Non-Argumentative Sentences.

3.7 Viraj Salaka Gamage (140173T) - Contributions

3.7.1 Contribution to the Study on Identifying Relationships Among Sentences in Court Case Transcripts Using Discourse Relations

I have implemented a rule based approach to identify the citation relation within a legal case which is one of the five discourse relations defined in our study. The reason behind moving for a rule based approach is that we could identify several patterns to detect citation relation.

If a citation is included within a sentence taken from a legal case, it means that the sentence refers to a previous case which has already been resolved. Therefore, it is required to identify rules as much as possible in order to detect such citations. In this case, the concern is about identifying the relationship between a sentence pair. Rather than considering the sentences including citations, we consider if the target sentence provides a citation to the source sentence.

These are some of the rules we have used to detect the citation relation.

- See ([A-Z][a-z]+)+v. ([A-Z][a-z]+) .*
- See [0-9]+ U. S. [0-9]+ .*
- Id., at [0-9]+.*
- App. [0-9]+.*

Proper noun followed by “v.” and another proper noun refers to a legal case name. At times without the name only the legal case number is provided. ”Id” and ”Ibid” gives the meaning that it is the same citation as the previous one. ”App. " refers to the term appellate. Other than these, rules are implemented to identify the citations referring to statute as well.

Preparing the test data set during the evaluation process has been a tough challenge. We had to annotate around 600 sentence pairs to complete that task. It was harder due to the fact that the sentence pairs are taken from legal cases. I could do a major contribution during the manual annotation process as well.

3.7.2 Contribution to the Study on Sentiment Analysis in the Legal Domain

There was a requirement of solving the identification of shift in view relation precisely from the discourse relations study. As a solution, we have decided to come up with a sentiment based approach. To solve the party identification task, we could incorporate an automatic sentiment annotator.

We researched about available sentiment annotators. But we have not been able to find a sentiment annotator specifically for legal domain. And the available sentiment annotator models (Sentiwordnet [40, 41] and CoreNLP Sentiment annotator [44, 77]) did not work well for legal domain, as there are major differences in sentiment between legal domain and some other domain like movie reviews. There has been an option to create a training data set containing phrases with its sentiment and then train a model for legal domain but it is a costly operation in terms of human effort and time. Hence we moved for a transfer learning method, where we used a model trained using movie reviews and adapted it to the legal domain. The implementation details have been included inside the methodology section.

First requirement has been to identify the unigrams with deviated sentiment. To complete that task, more than 7000 unigrams (which has been identified during the *selecting vocabulary process*) were needed to be manually annotated by three human judges. I was one of them. In addition, it has been required to understand the implementation done in Stanford CoreNLP Sentiment annotator which has been used as the source model in this task. After understanding the source model's implementation, I have developed the target model's implementation with the vector substitution and provided an interface to use the model.

And finally, it has been required to test the accuracy of the model. For that, we have annotated 513 phrases extracted from legal cases. I have also been involved in the annotation process.

3.7.3 Contribution to the Detecting Shift-in-View Relationships by Comparing Properties Related to Identified Verbs

In this task, my contribution lies with the sentiment based approach. For that, we have used the sentiment annotator mentioned above. when there is a shift-in-view relationship, there has to be two opinions about the same thing or the same person. So the approach taken here is to identify whether there is a common subject between the considered sentence pair and if the sentiment is different in the two phrases which are bound to that subject. So that is basically the approach taken to identify shift in view relation. Consider the following example.

- Sentence 1 : The Government contends that Lee cannot show prejudice from accepting a plea.
- Sentence 2 : Lee, on the other hand, argues he can establish prejudice under Hill.

In here, the government and Lee are the subjects of main clauses. And the sentiment of the main clauses are non-negative. But in the first sentence, the subordinating clause has a negative sentiment towards Lee while in the second one it shows non-negative sentiment towards Lee. Therefore we can identify that there is a shift in view relation. This approach is explained in detail in the overall methodology section.

If the considering sentence's subject is a pronoun, it is not right to consider that the sentence's subject is pronoun itself. It should have been resolved. Other than that, there may be occasions where the same entity or person referred by different names. For that task, we have used Stanford CoreNLP Coreference Resolution.

In the legal cases, the sentences are lengthy because there are two many subordinating clauses connected to a main clause. Therefore, we have used Stanford CoreNLP Constituency Parser to split the sentences by subordinating conjunctions. In Constituency Parser, subordinating conjunctions are denoted by the "*SBAR*" tag.

After having all subordinating clauses and main clauses as separate sentences, the next task was to identify the subject of each clause. To solve that problem, we have used Stanford CoreNLP Dependency Parser. The relation we have considered is "*nsubj*". But

that only captures the subject when it is active voice. To identify the subject when it is in passive voice, the "*nsubjpass*".

But the approach could not provide better results in terms of precision. The reasons for that are explained in detail in the methodology section. I have been contributed to the research paper "Fast Approach To Build a Sentiment Annotator Using Transfer Learning" as the first author.

3.7.4 Contribution to the Study on Extracting Argumentative Sentences from Court Case Transcripts

One of the major tasks in this project is to identify the arguments from a court case. My contribution lies with the citation based argument detection. As we have been advised by the legal expert, a sentence is usually have an argumentative value when there is a citation associated with that sentence. That statement logically sounds because when a sentence is associated with a citation, it links to a previous legal case. And that statement brought from a previous case because there is a specific argumentative value within it.

We have implemented a rule based system to identify the citations. This task is somewhat different from the previous task because now we are not only concerned about whether there is a citation relation between two sentences. It is required to capture when there is a citation included within a sentence. And also it needs to identify the type of citation as well. Some of the rules are explained below.

First we identify whether there is a citation relation between two consecutive sentences. In this case, we have used the rules used in the study related to identify relationships using discourse relations. If so, the preceding sentence to the citation including sentence is used. But it needs to be further categorized based on citation type. The term "*App*" denotes that it has been taken from the appellate form. So it has to be considered as a fact rather than an argument. When the citation is based on terms like "*Id., at* " and "*Ibid*" it means that the citation is the same as mentioned in the preceding citation.

In order to detect the arguments sentences containing the citations, we can use the

following basic rules,

- .* [A-Z][a-z]+ v. [A-Z][a-z]+ .*
  ex: It forgets that categorical rules are ill suited to an inquiry that we have emphasized demands a "case-by-case examination" of the "totality of the evidence." Williams v. Taylor.

- .* [0-9]+ U. S. [0-9]+.*
  ex: The Government urges that "[a] defendant has no entitlement to the luck of a lawless decisionmaker." 466 U. S., at 695.

But there are other occasions where the arguments from previous cases referred by only one party in the case. To identify those references, it is required to keep track of the previously appeared citations. Consider the following statement,

- In Hill, the Court concluded that the defendant had not made that showing, so it rejected his claim.

In here, we need to know that the term "*in Hill*" refers to the previous case called *Hill v. Lockhart*.

3.7.5 Contribution for Research Paper publications

I contributed to the research paper "Fast Approach to Build an Automatic Sentiment Annotator for Legal Domain using Transfer Learning" as the first author. In addition, I contributed to the sentiment based approach included within the research paper called "Shift of Perspective Identification in Legal Cases".

3.8 Gathika Ratnayaka (140528M) - Contributions

3.8.1 Contribution to the Study on Identification of Sentences in Court Case Transcripts

3.8.1.1 Defining Discourse Relations

After examining court case transcripts and identifying the requirements for generating an argument tree, five major relationship types that can be observed between court case transcripts were defined with the agreement of other group members. *Redundancy*, *Elaboration*, *No Relation*, *Citation* and *Shift-in-View* are the major relationship types which were defined. The definitions of the relationship types are provided under the Section 3.1.3.

The main purpose behind these definitions is to facilitate the process of representing information in court case transcripts in a structured, tree like manner as shown in Fig. 1. If we consider the tree structure shown in Fig. 1, the supportive argument/supportive fact for a particular argument will be a child node of the major argument. *Elaboration* relationship can be used to identify such child nodes. If two arguments presented by the same party discuss completely different topics, those two arguments can be presented as sibling nodes in an argument tree as shown in Fig. 1. *No Relation* relationship type would be useful in identifying such situations. There are sentences that support an argument by providing references from previous court cases. *Citation* relationship was defined to identify such sentences. There can be occurrences where the same sentence repeats twice or same information is provided again and again within a court case transcripts. When representing information in a court case transcripts in a structured manner, such redundant information should be eliminated. *Redundancy* relationship type is useful when it comes to identify redundant sentences. There are situations where two sentences in a sentence pair is discussing on the same topic but providing different opinions on the same topic. *Shift-in-View* relationship type is intended to identify such sentence pairs. This is a very important relationship type as it has the potential to facilitate the process of identifying counter arguments to a particular argument.

3.8.1.2 Adopting CST Relations

As there was no dataset where relationships between sentences in court case transcripts are annotated, CST Bank Data Set[89] was chosen to use as the dataset. It contains sentence pairs obtained from newspaper articles. Eighteen relationship types between sentences are defined in CST[22] at a very granular level. When analyzing the study by Zahri et al [25], where CST dataset is being used for text clustering, I could observe how that study has redefined CST relationships to achieve their research purpose. That study [25] has defined eight new relationship types by combining granular CST relationship types which overlap with each other. Those eight relationship types have also been used in another study[29] to facilitate text summarizing. Observing that CST relationships can be redefined in order to facilitate the intended research purpose, granular relationship types in CST were aligned with the Five Relationship types which are defined in our research (as described in Section 3.1.3).

3.8.1.3 Expanding the Dataset

As shown in Table 3.2, there is no relationship type in CST which aligns with the "No Relation" relationship defined in our study. As a result, it was needed to manually annotate such sentence pairs and add them to the dataset. I actively contributed to that process by manually annotating sentence pairs with "No Relation" relationship type. All of those sentence pairs were obtained from previous court cases.

3.8.1.4 Feature Extraction from Legal Sentence Pairs

1. Longest Common Substring

Longest Common Substring is the maximum length word sequence which is common to both sentences. When the number of characters in longest common substring is taken as $n(LCS)$ and number of characters in source sentence is taken as $n(S)$, Longest Common Substring Ratio ($LCSR$) can be calculated as,

$$LCSR = \frac{n(LCS)}{n(S)} \tag{7}$$

This value indicates the part of the target sentence which is present in the source sentence. Thus, this will be useful especially in determining discourse relations such as

Overlap, *Attribution*, and *Paraphrase*.

2. Semantic Similarity between Sentences

This feature is useful in determining the closeness between two sentences. Semantic similarity will provide the closeness between those two words. A method described in the study by Tayal et al[21] is adopted when calculating the semantic similarity between two sentences. Semantic similarity score for a pair of sentences is calculated using WordNet::Similarity [84]. When calculating the semantic similarity of two words, their synonym sets were also considered.

$$score = Average\left(\sum_{i=1}^{n} NounScore + \sum_{i=1}^{n} VerbScore\right) \tag{8}$$

3. Transition Words and Phrases

Availability of a transition word or a transition phrase at the start of a sentence indicates that there is a high probability of having a strong relationship with the previous sentence. For example, sentences beginning with transition words such as *And*, *Thus* usually elaborates the previous sentence. Phrases such as *To make that*, *In addition* at the beginning of a sentence also implies that the sentence is elaborating on the details provided in the previous sentence. Considering these linguistic properties, two boolean features were defined.

1. Elaboration Transition: If the first word of the source sentence is a transition word which implies elaboration such as *and*, *thus*, *therefore* or if a transition phrase is found within first six words of the source sentence, this feature will output 1. If both of above two conditions are false, the feature will return 0. Two lists containing 59 transition words and 91 transition phrases which implies elaboration are maintained. Though it is difficult to include all transition phrases in the English language which implies elaboration relationship, we can clearly say that if these phrases are present at the beginning of a sentence, the sentence is more than likely to elaborate the previous sentence.

2. Follow Up Transition: If the source sentence begins with a word like *however*, *although* or phrases like *in contrast*, *on the contrary* which implies that the source sentence is following up the target sentence, this feature will output 1. Otherwise, the feature will output 0.

4. Length Difference Ratio

This feature considers the difference of lengths between the source sentence and the target sentence. When $length(S)$ and $length(T)$ represent the number of words in source sentence and target sentence respectively, Length Difference Ratio (*LDR*) is calculated as shown below.

$$LDR = 0.5 + \frac{length(S) - length(T)}{2 * Max(length(S), length(T))} \tag{9}$$

In a relationship like Subsumption, the length of the source sentence has to be more than the length of the target sentence. In Identity relationship, both sentences are usually of the same length. These properties can be identified using this feature.

5. Attribution

This feature checks whether a sentence describes a detail in another sentence in a more descriptive manner. Within this feature, we check whether a word or phrase in one sentence is cited in the other sentence using a quotation mark to determine this property. This property can be further explained using Example 5 which is taken from *Turner v. United States* [90]:

Example 9

- Sentence 9.1 (Target): *Such evidence is 'material' . . . when there is a reasonable probability that, had the evidence been disclosed, the result of the proceeding would have been different.*
- Sentence 9.2 (Source): *A 'reasonable probability' of a different result is one in which the suppressed evidence 'undermines confidence in the outcome of the trial.*

It can be seen that source sentence define or provides more details on what is meant

by “reasonable probability” in the target sentence. Such properties can be identified using the *Attribution* feature.

3.8.1.5 Developing a SVM model

LibSVM [91], which is an open source Java library was used for developing a SVM Model. The type of the SVM model is C-SVC. RBF (Radial Basis Function) Kernel has been used with the SVM Model. Advantage of using RBF kernel over Linear Kernel is that the RBF kernel is able to handle the situations where the relations between features and class labels are nonlinear. Thus, RBF kernel is more suited for determining the relationship type of a given pair of sentences.

It was needed to choose a value for the C parameter of the C-SVC SVM model from the two possible values "0" and "1". In order to determine the most suited value for the C parameter, two SVM models with C parameter values 0 and 1 respectively were trained using CST Bank Dataset. As our study was focused on the legal domain, 30 sentence pairs from court cases which were manually annotated for relationship between sentences were initially used in determining the accuracy of the two SVM models. It was empirically decided that SVM model with C parameter value of 1 has higher accuracy than a SVM model with C parameter value 0. A mechanism to save a trained SVM model as a text file was adopted in order to facilitate the re-usability of the model.

3.8.1.6 Data Annotation

As the proposed system has to be evaluated in the legal domain, it was needed to manually annotate sentence pairs obtained from court case transcripts in order to evaluate the system. Approximately, 600 such annotations were performed. Out of these 600 annotations, more than 400 annotations were performed by our group members. I actively participated in that process.

3.8.1.7 Evaluating and Analyzing Results

As shown in Table 3.3, a confusion matrix was generated to analyze the results. Precision , recall, f-measure values were also calculated based on the number of judges

agreeing upon a single relationship type for a given pair as shown in Table 3.4 and in Table 3.5. Human-Human Correlation vs Human-System Correlation was calculated for the overall system as well as for every relationship type as shown in the Table 3.6. The above mentioned evaluation methods and matrix generation were automated in order to facilitate result analysis and re-usability.

3.8.2 Contribution to the Study - Shift of Perspective Identification in Legal Cases

3.8.2.1 Identifying sentences which discuss the same topic

As described in previous studies [31, 32], in order to have a contradiction or change of perspective within the information provided by a given pair of sentences, the two sentences should discuss the same topic. **Sentence Relationship Identifier**, the system we have developed as described in Section 3.1, has been used to identify whether the two sentence in a given pair of sentences are discussing the same topic or not. As described in Section 3.1, the empirical results have shown that **Sentence Relationship Identifier** is successful in identifying whether two sentences are discussing the same topic or not. Only the sentence pairs which are identified as having *Elaboration* by **Sentence Feature Extractor** are further processed to detect *Shift in View* relationships. As explained in Section 3.3.4, relationship types such as *Redundancy*, *Citation* were not considered even though they suggest that the two sentences are discussing the same topic.

3.8.2.2 Filtering Sentences Using Transitional Words

When a source sentence (second sentence) of a sentence pair is starting with a transition word or with a transition phrase, the sentence pair probably have the *Elaboration* relationship type. Therefore, a mechanism was developed in order to eliminate such sentence pairs when it comes to *Shift in View* identification as described under the Section 3.3.5.

3.8.2.3 Identifying Shift of Perspectives by Analyzing relationships between Verbs

Detecting *Shift in View* relationship by analyzing relationships between verbs in two sentences is one of the novel and major approach which was introduced in our study "Shift of Perspective Identification in Legal Cases" as described in Section 3.3. An overview of this methodology has been provided under the Section 3.3.7. This section is intended to provide more detailed description about the methodology.

The major focus in this approach is to compare verbs in the two sentences which are connected with same entity (subject or object). To facilitate this process, the sentences are first split into sub sentences. Stanford CoreNLP Constituency Parser [81] was used for this purpose. The reason behind splitting the sentences is that it enables the identification of verbs which are connected with the same entity with a higher accuracy. Additionally, it also facilitate the process of triple extraction. Consider the following example,

Example 10

- Sentence 10.1: *Applying the two-part test for ineffective assistance claims from Strickland v. Washington, 466 U. S. 668, the Sixth Circuit concluded that, while the Government conceded that Lee's counsel had performed deficiently, Lee could not show that he was prejudiced by his attorney's erroneous advice.*
- Sentence 10.2: *Lee has demonstrated that he was prejudiced by his counsel's erroneous advice.*

If we consider the Sentence 9.1, using Stanford Constituency Parser, that sentence can be divided for sub sentences. *"Lee could not show that he was prejudiced by his attorney's erroneous advice"* is sub-sentence in Sentence 9.1. As the comparison is done between sub-sentences of both Source Sentence and Target Sentence, the sub-sentence *"Lee could not show that he was prejudiced by his attorney's erroneous advice"* of the Target Sentence will be compared with the sub-sentence *"Lee has demonstrated that he was prejudiced by his counsel's erroneous advice"*. Then it can be identified, both the sub-sentences are having the same subject *Lee*. It can be detected that *show* and *demonstrate* are the verbs which connected with the considered subject ("Lee") and analysis can be performed on those verbs. Therefore, performing comparisons at sub-sentence level will facilitate the process of identifying the entities that are connected with a particular verb. Otherwise, ambiguities can occur as one sentence can be made

up of several sub-sentences, and each sub-sentence may have its own subject and object.

After verbs which are connected with the *same entities* are found from the Source Sentence and the Target Sentence, such a verb pair is compared in order to determine whether the two verbs are providing a similar meaning. The approach taken to determine verbs with similar meaning is described under the Section 3.3.8 and that sub task was performed by another member of our group. As described in that section, **Lin** score of 0.86 was selected as the Semantic Similarity score to determine that two verbs are conveying a similar meaning.

Then analysis was performed on identified verb pairs. First methodology was based on negation relationship as described in Section 3.3.7. Stanford CoreNLP dependency parser was used for this purpose and negations can be identified using the "neg" tag. Also, it was considered whether a verb is connected with words such as "nothing", "never" which suggest negations.

Additionally, a novel approach was introduced to determine the *Shift-in-View* relationship type using Adverbial Modifiers. First, a list of adverbial modifiers belong to English language was created. Then the adverbial modifiers were grouped into three main classes based on *Frequency*, *Tone* and *Manner*. Then for each of that class, two sub classes were defined based on the polarity as shown in Table 3.12. Then adverbial modifiers which are relevant to each of these sub-classes were maintained in separate lists as shown in the Table 3.12. When an identified verb pair is considered, it is first checked whether the two verbs belong to the same adverbial modifier type (Frequency,Time,Manner). If that is the case, it is checked whether one verb is connected to an adverbial modifier which is with a positive polarity while other verb is connected to an adverbial modifier which is with a negative polarity. If such relationship between two verbs in a verb pair is found, it can be concluded that *Shift-in-View* relationship is present between the two sentences.

3.8.2.4 Results Analyzing

Table 3.13 show the results obtained by analyzing the three major approaches used in *Shift in View* detection. Sentence pairs detected by each of the approach were manually annotated. It could be seen that the approach of analyzing verbs between two sentences

is identifying *Shift-in-View* sentence pairs with a precision of 0.609. As it was the only approach that determine *Shift-in-View* relationship with a precision more than 0.5, it was combined with the **Sentence Relationship Identifier** system. The overall results of the combined system is shown in Table 3.14.

3.8.3 Contribution for Developing a System to Generate an Argument Tree

As shown in Fig. 2, a system which can generate a tree like structure when a court case transcript is provided as the input was developed. The major limitation of this system is that it is limited only to United States criminal court cases where Government is an one party of the court case. The major reason behind this limitation is that the system does not have the ability to automatically identify major parties in a court case transcript.

A list containing major legal person entities in a criminal court cases is maintained. It includes legal person entities such as *government*,*defendant*, and *petitioner*. Then the court case transcript is split into sentences and the sentences are pre-processed. The main intention behind pre-processing is to facilitate triple extraction. Then the triples were extracted from sentences and the subjects of the triples were examined. In a given sentence, if the subject is a legal person entity which is maintained in the list, that sentence is assigned as a sentence belongs to the particular legal entity. If the subject is not a legal person entity, the legal person entity which is connected to that sentence was determined by performing co-referencing.

After the sentences are assigned to legal person entities (can be considered as major parties), the sentences in each major entity is sent separately to **Sentence Relationship Identifier** to identify the relationships between those sentences. A mechanism was developed to show the sentences which have *Elaboration*, *Redundancy*, *Citation*, *Shift-in-View* as parent child nodes. Sentences with *No Relation* relationship was shown as Sibling Nodes. Also, the sentences which provides citation are specifically identified using the rule-based approach to detect citation. Then the details of each node was converted to a JSON format to be sent into the front-end of the Argument Tree generation system.

3.8.4 Contribution to Data Annotation

In the study **Identifying Relationships among Sentences in Court Case Transcripts**, our group members performed around 400 annotations (200 sentence pairs). In the same manner, approximately 500 phrases were annotated during the study **Fast Approach to Develop a Sentiment Annotator for Legal Domain**. Also in the study of **Extracting Argumentative Sentences from Court Case transcripts**, we have annotated around 170 sentences. Additionally, we had to annotate more than 250 sentences in the study **Shift of Perspective Identification within Legal Cases**. I have actively contributed for these annotation tasks.

3.8.5 Contribution for Research Paper publications

I contributed to research paper publications as the first author of the two research papers *"Identifying Relationships among Sentences in Court Case transcripts"* and *"Shift of Perspective Identification in Legal Cases"*.

3.9 Thejan Rupasinghe (140536K) - Contributions

3.9.1 Contributions to the study - Identifying Relationships Among Sentences in Court Case Transcripts Using Discourse Relations

3.9.1.1 Replacing coreferences for the input pair of sentences

Coreference resolution is the is the process of finding all the terms referring to a same entity in a given text. The following two sentences are taken from *Lee v. United States* [55].

Example 11

- Sentence 11.1: ***Petitioner Jae Lee*** *moved to the United States from South Korea with* ***his*** *parents when* ***he*** *was 13..*
- Sentence 11.2: *In the 35 years* ***he*** *has spent in this country,* ***he*** *has never returned to South Korea, nor has* ***he*** *become a U. S. citizen, living instead as a lawful permanent resident..*

Here the “Petitioner Jae Lee" in the Sentence 10.1, is referred using the pronouns “he" and “his" in both sentences. Stanford CoreNLP CorefAnnotator (*coref*) [66] is used here to identify the *representative mentions* (in this example “Petitioner Jae Lee") in both sentences. The two sentences are concatenated together and annotated as a whole, to find the *coreference chains* inside them. Then the system is developed to replace the words in the *coreference chains* with their *representative mentions*. In this example “he" and “his" is replaced with “Petitioner Jae Lee". Then the sentences in Example 8 are changed as shown below.

Example 11 (updated)

- Sentence 11.1: *Petitioner Jae Lee moved to the United States from South Korea with* ***Petitioner Jae Lee*** *parents when* ***Petitioner Jae Lee*** *was 13.*
- Sentence 11.2: *In the 35 years* ***Petitioner Jae Lee*** *has spent in this country,* ***Petitioner Jae Lee*** *has never returned to South Korea, nor has* ***Petitioner Jae Lee*** *become a U. S. citizen, living instead as a lawful permanent resident.*

After the resolving the coreferences each sentence is annotated again through the Stanford CoreNLP pipeline [92] before calculating the features. By resolving coreferences, calculating Noun Similarity, Verb Similarity, Adjective Similarity, Subject

Overlap Ratio, Object Overlap Ratio, Subject Noun Overlap Ratio and Semantic Similarity between Sentences features were made more effective.

3.9.1.2 Feature Implementations

6. Cosine Similarities

Following cosine similarity values are calculated for a given sentence pairs,

- Word Similarity
- Noun Similarity
- Verb Similarity
- Adjective Similarity

Following equation is used to calculate the above mentioned cosine similarities.

$$CosineSimilarity = \frac{\sum_{i=1}^{n} FV_{S,i} * FV_{T,i}}{\sqrt[2]{\sum_{i=1}^{n}(FV_{S,i})^2} + \sqrt[2]{\sum_{i=1}^{n}(FV_{T,i})^2}} \tag{10}$$

Here $FV_{S,i}$ and $FV_{T,i}$ represents frequency vectors of source sentence and target sentence respectively. Standford CoreNLP POS Tagger (*pos*) [83] is used to identify nouns, verbs and adjectives in sentences.

In calculating the Noun Similarity feature, *singular* and *plural nouns*, *proper nouns, personal pronouns and possessive pronouns* are considered. Both *superlative* and *comparative adjectives* are considered when calculating the Adjective Similarity. The system ignores verbs that are lemmatized into *be*, *do*, *has* verbs when calculating Verb Similarity feature as the priority should be given to effective verbs in sentences.

7. Word Overlap Ratios

Two ratios are considered based on the word overlapping. One ratio is measured in relation to the target sentence. Another ratio is measured in relation to the source sentence. These ratios provide an indication on the equivalence of two sentences. For example, when it comes to a relationship like subsumption, source sentence usually contains all the words in the target sentence. This property will be also useful in

determining relations such as Identity, Overlap (Partial Equivalence) which are based on the equivalence of two sentences.

$$WOR(T) = \frac{Comm(T,S)}{Distinct(T)} \tag{11}$$

$$WOR(S) = \frac{Comm(T,S)}{Distinct(S)} \tag{12}$$

$WOR(T)$, $WOR(S)$ represents the word overlap ratios measured in relation to source and target sentences respectively. $Distinct(T)$, $Distinct(S)$ represents number of distinct words in source sentence and target sentence respectively. The number of distinct common words between two sentences are shown by $Comm(T,S)$.

8. Grammatical Relationship Overlap Ratios

Three ratios which represent the grammatical relationship between target and source sentences are considered.

- Subject Overlap Ratio

$$SubjOverlap = \frac{Comm(Subj(S), Subj(T))}{Subj(S)} \tag{13}$$

- Object Overlap Ratio

$$ObjOverlap = \frac{Comm(Obj(S), Obj(T))}{Obj(S)} \tag{14}$$

- Subject Noun Overlap Ratio

$$SubjNounOverlap = \frac{Comm(Subj(S), Noun(T))}{Subj(S)} \tag{15}$$

All these features are calculated with respect to the source sentence. $Subj, Obj, Noun$ represents the number of subjects, objects, and nouns respectively. $Comm$ gives the number of common elements.

Stanford CoreNLP Dependency Parse Annotator (*depparse*) [81] is used here to identify subjects and objects. All the subject types including nominal subject, clausal subject, their passive forms and controlling subjects are taken into account in calculating the number of subjects. Direct and indirect objects are considered when calculating the number of objects. All subject and object types are referred from *Stanford typed dependencies manual* [87].

9. Number of Entities

Ratio between number of named entities can be used as a measurement of relationship between two sentences.

$$NERatio = \frac{NE(S)}{Max(NE(S), NE(T))} \tag{16}$$

$NE(X)$ represents the number of named entities in a given sentence X. Stanford CoreNLP Named Entity Recognizer (NER) [93] was used to identify named entities which belong to 7 types; PERSON, ORGANIZATION, LOCATION, MONEY, PERCENT, DATE and TIME.

3.9.1.3 Feature Calculation and Discourse Type API

Annotating the sentences through the Stanford CoreNLP pipeline and calculating the features, were high memory and CPU consuming tasks, for which our local computers were not enough. So virtual machines from Google Compute Engine of the Google Cloud Platform [94] were used for feature calculation.

A REST API, to take the discourse relation type of any given two sentences, was developed and hosted on Digital Ocean [95], where it takes a JSON in the format,

```
{
    "target-sent":"Lee has demonstrated that he was prejudiced.",
    "source-sent":"Lee could not show that he was prejudiced."
}
```

and returns a JSON containing the discourse relation type number (There was a prior agreement about the number and relation type mapping), in the below format.

```
{
    "type":5
}
```

3.9.2 Contributions to the study - Shift-of-Perspective Identification within Court Cases

3.9.2.1 Discovering Inconsistencies among Triples

Here we adopted the methodology presented in the study which discovers inconsistencies in PubMed abstracts[10] from the medical domain. In this study, they have introduced an Ontology-Based Information Extraction method to detect occurrences of inconsistencies in microRNA research paper abstracts using open information extraction. They have found 102 inconsistencies relevant to the microRNA domain form the downloaded 36877 abstracts from the PubMed database.

In this study a dictionary for the medical domain is developed. Following the same methodology, a legal term dictionary was constructed to be served as a *Semantic Lexicon* for the system. 200+ court case transcripts were used to extract words for the process. Then a word list consisting 17,000+ unique words were developed by removing stop words. A TF-IDF algorithm [88] based method is used to calculate a value for each term in the dictionary.

$$TermValue = \frac{\sum_{i=1}^{casecount} \frac{f_{t,d}}{termcount}}{D.F} \tag{17}$$

Raw count ($f_{t,d}$) for each term is taken, considering each court case transcript as a document. Term frequency value for a term is calculated by dividing the raw term count by the total number of terms in the case. Term frequency value for each case is added together and the result value is divided from the document frequency ($D.F$), to calculate the value for a term in the dictionary. Then all the term values are normalized according to the equation 18.

$$NormalizedTV = \frac{(TV - TV_{min}) * (1 - TV_{min})}{TV_{max} - TV_{min}} + TV_{min} \tag{18}$$

Here TV_{min} and TV_{max} represent the minimum and maximum values of the term values respectively. This normalized value is used to be served as the semantic weight for the system.

First, coreference resolving is done on the sentence pairs using the Stanford CoreNLP CorefAnnotator (“coref") [66] and the pairs with *Transition Words and Phrases* are filtered out. Then OLLIE [80], open information extraction system, is used to extract triples, in (*Subject; Relationship; Object*) format, from sentences. When comparing two sentences, for the *Shift-in-View* relationship, only triple pairs with same subject or object are considered, as the *Shift-in-View* relationship talks about different perspectives on the same topic or entity. The stop words removed relationship strings of a triple pair are then compared with each other word by word. The comparison is performed in three ways in the **PubMed Study**.

1. Words which are exactly the same
2. Exactly same words with one word negated with “not" (eg. “increased" - “not increased")
3. Different words

In our study we modified above 1 and 2, types to consider words with similar meanings, instead of considering only the words which are exactly the same. Modified approaches are,

1. Words which have **similar meaning** (Lin score >= 0.86)
2. **Similar words** with one word negated with “not" (eg. “showed" - “not demonstrated")
3. Different words (Lin score < 0.86)

When both the comparing words have similar meaning, the weight of the word is taken from the legal dictionary and it is raised to the power of two and then multiplied by the constant “yes weight" (W_{yes}), added the resultant to the similarity amount ($simil_T$) and the similarity number counter (s_n) is increased by one. When the comparing words

are similar and one word negated with “not" same procedure is followed and and the resultant value is then multiplied by the constant “no weight" (W_{no}) and added to the difference amount (dif_T) and the difference number counter (d_n) is increased by one.

When comparing different words (Lin score < 0.86) a oppositeness value ($oppo$) is calculated using the Lin similarity measure, antonyms of words with the equations provide in the **PubMed study**. If the calculated $oppo$ is greater than or equal zero, it is multiplied by the constant “yes weight" (W_{yes}), added the resultant to the similarity amount ($simil_T$) and the similarity number counter (s_n) is increased by one. If the $oppo$ value is less than zero then, it is multiplied by the constant “no weight" (W_{no}) and added to the difference amount (dif_T) and the difference number counter (d_n) is increased by one.

Then, if $simil_T$ value is greater than dif_T, $simil_T$ is returned as the similarity measure for the two triples. Otherwise dif_T multiplied by -1 is returned as the difference measure for the two triples. Finally, the maximum absolute value from, the $simil_T$ and dif_T values taken from each triple combination, is considered to be the similarity or difference measure for two sentences. A threshold based on the similarity or difference values is introduced empirically to select sentence pairs which have the *Shift-in-View* relationship.

Only a precision of 0.27 could be observed in this approach. The following reasons may have contributed to the poor performances of that approach. From the 2150 sentence pairs which were considered, oppositeness values were not calculated for 1570 pairs. Containing at least one sentence within a sentence pair in which the triples could not be extracted by OLLIE[80] is a major reason for not having an oppositeness value. Even if the triples are extracted from both sentences, if there is no matching between either subjects or objects of the two sentences, an oppositeness value will not be calculated for a sentence pair. The error in identifying similar words with **Lin** measure would also get propagated to this approach also.

3.9.3 Contributions to the study - Extracting Argumentative Sentences from Court Case transcripts

The objective of this study is to determine if a given sentence from a court case transcript, is an argument or not. Methodologies in previous literature can not be used in extracting arguments in US court cases, because of the lack of annotated argument sets and unidentified argument patterns.

So we have to come up with a novel approach to identify arguments in US court case transcripts. We consulted a legal expert to identify determined structures that can be used to identify argumentative sentences in court case transcripts.

3.9.3.1 Argument extraction using verbs

First, I read court cases and identified argumentative sentences, which were later clarified by discussing with the legal expert. Then I prepared a list of words which are usually been used to represent arguments. As examples, argue, agree, conclude, rejected, contest, contend,testify, concede and claim. Then we extracted sentences considering these words and annotated as arguments and non-arguments. There we saw that some wrong sentences were detected because comparing just words. As example,

- The D. C. Superior Court rejected petitioners' Brady **claims**, finding that the withheld evidence was not material.

This is a decisions of a court which is not an argument presented by a party. To eliminate these sort of wrong detections, we decided only to compare verbs extracted from the sentences with the list of pre-identified word. So verbs in a sentence is extracted and lemmatized using Stanford CoreNLP pipeline[92], and then compared with the verb list.

3.9.4 Contributions to Result Calculation and Data Annotation

In the process of evaluating the **Sentence Relationship Identifying** system for the legal domain, first the legal sentence pairs were run through the system and the calculated features values and the discourse relation type numbers were stored in a database. Out

of the 600+ manual annotations of sentence pairs, more than 400 were done by the team members, where I also participated. In the study of **Legal Argument Extraction** we have annotated around 200 sentences. In the study **Fast Approach to Develop a Sentiment Annotator for Legal Domain**, I also participated in manual annotation of sentiment phrases, which was a collaborative task of our group members manually annotated the sentiment of approximately 500 phrases.

3.9.5 Contributions to Research Papers

I have contributed for the two research papers *"Identifying Relationships among Sentences in Court Case transcripts"* and *Shift of Perspective Identification in Legal Cases* by writing the tasks which I performed.

3.10 Menuka Warushavithana (140650E) - Contributions

3.10.1 Contribution to the Study - Identifying Discourse Relations Among Sentences

3.10.1.1 A Web Application for Annotating Sentences Pairs

As mentioned in the **Experiments** section of the study on using discourse relations to identify relationships between sentences in legal cases, we needed to create an application to get the help of our friends with the data annotation tasks. The web application was created in PHP using the Laravel framework [96]. The source code for this application can be found here [3].

The website was hosted in a DigitalOcean [95] virtual machine. I also incorporated OAuth 2.0 login using Google to let the users have a seamless experience with the application. The database records show that there were more than 50 users registered on the application and contributed the data annotation process.

3.10.1.2 Database Design for the Web Application

The most critical part of this application was the design of the database. After thorough discussions with our supervisors, we arrived at the conclusion that the most effective way to measure the accuracy of the discourse analysis system is to adhere to the following criteria.

- The pairs of sentences should be divided into clusters such that a cluster contains 5 pairs.
- The sentence pairs in a single cluster should be annotated by only a single user. i.e. there should be a one-to-one mapping between the clusters and users.
- Each cluster should be annotated by at least two users.
- A user should not be able to submit the annotations without annotating all the sentence pairs (five) in a single cluster. i.e. annotating a single cluster is considered an atomic transaction in terms of database design.

[3] `http://bit.ly/discourse-annotator`

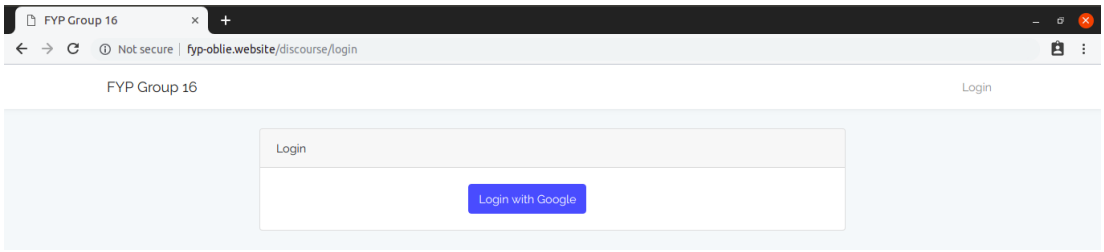

Fig. 7. Login Screen of the Data Annotation Application

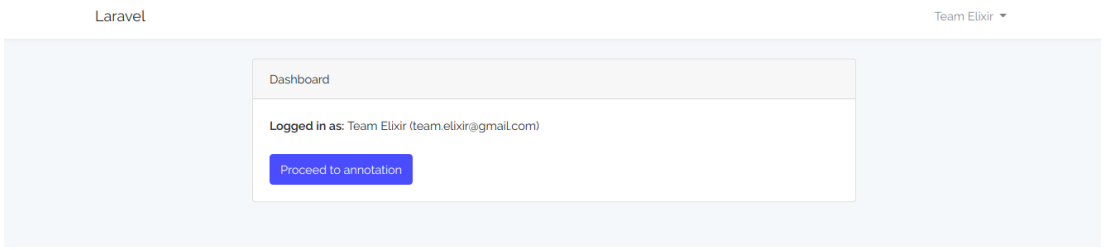

Fig. 8. Annotation Application after the User has Logged In

The task of identifying discourse relationships among sentences is somewhat subjective. Therefore, we could not have annotated a pair of sentences to state whether they had a particular relationship. Two different users could have different opinions on the relationship between two sentences. The only plausible way to measure the accuracy of the system was to compare the human-human correlation values with human-computer correlation values.

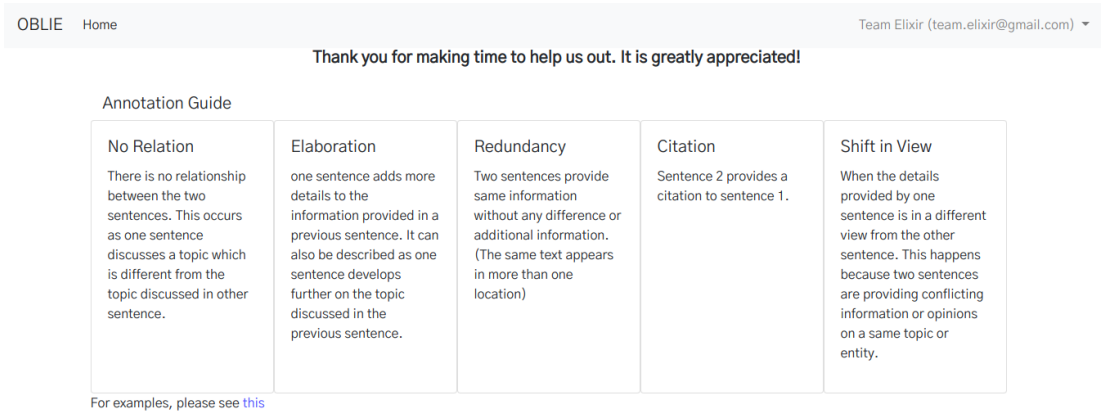

Fig. 9. Guide for Annotating Discourse Relations

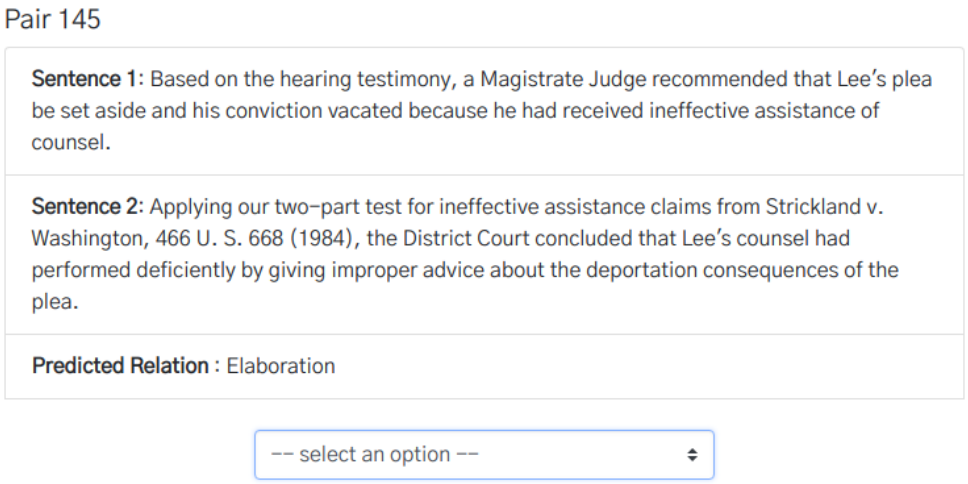

Fig. 10. Example Discourse Annotation

3.10.2 Crawling Legal Cases

As our target was on criminal law, I implemented a crawler in Python (using the Scrapy [97] Framework) and crawled all available criminal cases from CaseLaw-FindLaw site [2]. Then I separated the crawled cases into sentences and extracted triples (subject, relationship, object) to be used in further research.

3.10.3 Contribution to the Study - Sentiment Analysis in the Legal Domain

After crawling around 250 cases (all criminal cases that were tried by the United States Supreme Court which were available), I created a corpus of unique words found in the all those cases. It was essential to come up with a way to reduce the number of words in the corpus (vocabulary) before proceeding to annotation because of the limited availability of resources including manpower. First, stopwords were removed from the list of extracted words. The list mentioned in the Van stop-list [75] was used for this task. There were about 45,000 words at the end of this step. Afterwards, it was necessary that a graph was drawn against the words and their frequencies to determine a way to reduce the number of words to be annotate without having to compromise the accuracy of the model. It could be observed the graph that only about 7500 words were occupying (considering the cumulative frequencies) 95% of the selected words. It was

concluded that these 7500 should be annotated to continue with the experiment. Then, I annotated a portion of the 7500 words (the other 3 members of the group annotated the rest).

After substituting the word vectors with a selected list of words for the RNTN model (not done by myself), I extracted around a thousand phrases from the previously mentioned 250 criminal cases to check the accuracy of the improved model. Again, as the sentiments of these phrase not annotated, we had to manually annotate them. We could annotate the sentiments of around 500 phrases within the time that was available to finish the study. Then the precision and recall values could be measured separately for the original RNTN model mentioned in the Socher model [44] and our model which was tuned for the legal domain using transfer learning.

I also contributed this study by modifying the web application used for annotating the discourse relations among sentences from the previous study.

3.10.4 Contribution to the Study - Shift-of-Perspective Identification within Court Cases

One of the approaches we tried in the study Shift-of-Perspective Identification within Court Cases was, using verb similarity to determine verbs that convey similar meanings.

First, I used the Wu-Palmer semantic similarity [16] score since it was the most widely used similarity score in past studies. I extracted 1000 pairs of verbs from court cases (criminal) and annotated them on their similarity. Then I calculated Wu-Palmer scores on those 1000 verb pairs. Since semantic similarity scores output a value between 0 and 1, it was necessary to select a threshold score above which we could consider the verbs as having similar meanings. i.e. for instance, all verb pairs that have semantic similarities above a Wu-Palmer of 0.75 could be considered as having similar meanings. In order to systematically determine this threshold score, I first checked the precision, recall, and F-measures of the annotated pairs of verbs (1000) and then incremented the threshold by intervals of 0.1 up to 0.95 and tabulated (Table 3.11) the precision and recall values for each threshold.

In addition to Wu-Palmer, I also carried out the aforementioned experiments for

all other semantic similarity scores mentioned in Wordnet [84], which were **Hirst St Onge**, **Jian Conrath**, **Leacock Chodorow**, **Lesk**, **Lin**, **Path**, and **Resnik**. Among these scores, **Lin** and Jiang Conrath were the only ones (along with **Wu-Palmer**) that generated promising results. The graph drawn against the thresohold values and the F-measures for **Wu-Palmer**, **Lin**, and **Jiang Conrath** can be found in Fig. 6. It could be seen that **Lin** outperforms the other two measures when the F-measure are considered. A **Lin** score of 0.75 had the highest F-measure of 63.95%. However, it was due to considerably high recall and undesirably low precision values. As our target was to have a proper compromise between precision and recall values, **Lin** score of 0.86 was selected for the experiment.

3.10.5 Contribution to Creating the Argument Tree

Creating an argument tree which is a systematic way of representing the information we have extracted with the aid of the systems we built, was the final objective of our project. I created the front-end of the argument tree application using JavaScript and the graphing library GoJS [98]. It was necessary to input arguments in a JSON structure to correctly render the tree of arguments using GoJS. The following structure was used in representing a single argument in the tree.

```
{
  "key": "",
  "parent": "",
  "name": ",
  "type": ""
}
```

3.10.6 Contribution to Writing the Papers

Throughout this project, we conducted several studies and wrote a couple of papers. I contributed to writing the research papers by writing the parts from scratch on the parts I did alone, plus improving the language wherever possible. I also helped in formatting most parts of the papers using LaTeX[99].

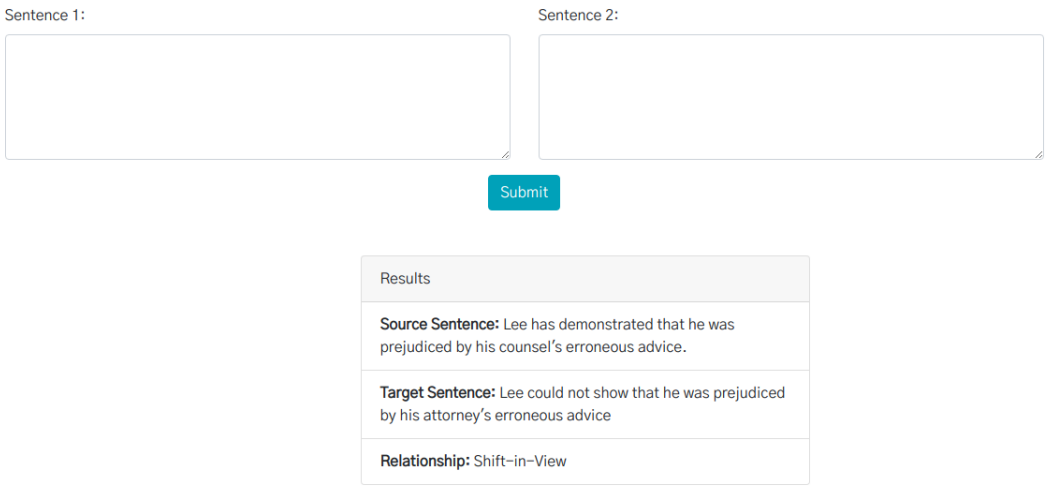

Fig. 11. Screenshot of the Discourse Analyzer Application

3.10.7 Using Cloud Virtual Machines for Calculations

A lot of processing power and time was required to get the results of the multiple studies we conducted. It was not enough run the systems in our local computers to get the results. I configured multiple virtual machines to be used for calculations. I used the Google Compute Engine of the Google Cloud Platform [94] and Digital Ocean [95] Droplets for setting up the virtual machines.

3.10.8 Contributing to Creating the Demonstrations

I contributed to creating a demonstration the discourse analyzer system by setting up a front-end user interface input two sentences and display the relationship between the sentences. An HTTP call is made to the backend REST API through an AJAX call and the response is shown in the user interface. A screenshot of the demonstration is shown in Fig. 11.

My contributions can be found at the GitHub repositories under the user handle **menuka94** [4].

[4] Visit `http://bit.ly/fyp-contributors-github-1` and `http://bit.ly/fyp-contributors-github-2`

4 CONCLUSION

The main research contribution of this study is coming up with a successful methodology to identify relationships between sentences in court case transcripts. It can be considered as a major step in the process of representing information related to court cases in a structured manner. It will enable a system to automatically detect facts which are elaborating on a certain argument. Also, it will provide an idea on the information flow within a legal case by identifying the changes in discussion topics. Our methodology to detect legal citations and to identify citation relationship among sentences proved to be highly successful. Additionally, we have introduced novel linguistic approaches to detect situations where two texts are providing different opinions on the same discussion topic. Identification of such situations, where two sentences are providing different opinions on the same discussion topic is useful in legal domain, especially when it comes to identifying counter arguments to a particular argument.

Our study demonstrates how rule based approaches can be combined together with a machine learning model to increase the accuracy of identifying relationships among sentences. Further, we have demonstrated that discourse relations can be successfully applied in the legal domain, suggesting that discourse relations can be redefined in order to facilitate the intended research purpose. Also, the proposed methodologies have the potential to be applied in various other applications such as discussion platforms and question answering systems.

We have also come up with rule based approaches that can successfully extract argumentative sentences from court case transcripts. This methodology can be used to differentiate arguments and non-arguments. Integrating this approach with the sentence relationship identifying methodology will enable the identification of supporting facts for a argument. This will also facilitate the representation of information flow within a legal court case transcript, in a structured manner.

Coming up with a fast approach to develop a Sentiment Annotator for Legal Domain can be considered as another major research contribution of this study. The methodology we have followed demonstrates how transfer learning can be used to develop a sentiment annotator in an environment where very limited resources are available. The proposed

methodology can be easily applied to develop a sentiment annotator for any domain where resources are very limited. The manually annotated dataset which was prepared during this study can be considered as another major research contribution. The dataset can be used for training and testing purposes in future research related to US court case transcripts.

The sentence relationship identification system together with argument extraction can be used to identify facts and citations that support a particular legal argument. Proposed approaches to detect arguments and *shift of perspective* can facilitate the task of identifying counter arguments to a given legal argument. Sentiment analysis can be used to identify whether a particular argument or a fact given in a sentence supports the plaintiff or the defendant. The sentiment annotator developed in this study can be used to improve the effectiveness of such a task. The above mentioned tasks can also be considered as the major future work when it comes to developing a system which can automatically suggest legal arguments.

REFERENCES

[1] "The day-to-day stresses and challenges of being a lawyer âĂŞ slaw," http://www.slaw.ca/2015/10/05/the-day-to-day-stresses-and-challenges-of-being-a-lawyer/, (Accessed on 02/06/2018).

[2] "Caselaw: Cases and codes - findlaw caselaw," https://caselaw.findlaw.com/, (Accessed on 05/20/2018).

[3] "Westlaw uk - online legal research from sweet & maxwell - westlaw uk," https://legalresearch.westlaw.co.uk/, (Accessed on 03/09/2018).

[4] "Bailii | ials," http://ials.sas.ac.uk/digital/bailii, (Accessed on 03/09/2018).

[5] "Ross intelligence - artificial intelligence meets legal research ross intelligence," https://rossintelligence.com/, (Accessed on 03/09/2018).

[6] "Ai based case research assistant | casemine," https://www.casemine.com/caseiq, (Accessed on 03/09/2018).

[7] "What is case law? definition and meaning - businessdictionary.com," http://www.businessdictionary.com/definition/case-law.html, (Accessed on 05/17/2018).

[8] D. C. Wimalasuriya and D. Dou, "Ontology-based information extraction: An introduction and a survey of current approaches," *Journal of Information Science*, vol. 36, no. 3, pp. 306–323, 2010.

[9] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[10] N. de Silva, D. Dou, and J. Huang, "Discovering inconsistencies in pubmed abstracts through ontology-based information extraction," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2017, pp. 362–371.

[11] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of*

52nd annual meeting of the association for computational linguistics: system demonstrations, 2014, pp. 55–60.

[12] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, "Open information extraction: The second generation." in *IJCAI*, vol. 11, 2011, pp. 3–10.

[13] M. Schmitz, R. Bart, S. Soderland, O. Etzioni *et al.*, "Open language learning for information extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 523–534.

[14] G. Angeli, M. J. J. Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 344–354.

[15] "Natural person - definition, examples, cases, processes," https://legaldictionary.net/natural-person/, (Accessed on 03/11/2018).

[16] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1994, pp. 133–138.

[17] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[18] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *arXiv preprint cmp-lg/9709008*, 1997.

[19] N. de Silva, "Safs3 algorithm: Frequency statistic and semantic similarity based semantic classification use case," in *Advances in ICT for Emerging Regions (ICTer), 2015 Fifteenth International Conference on*. IEEE, 2015, pp. 77–83.

[20] H. Shima, "Wordnet similarity for java (ws4j)," *Retrieved November*, vol. 23, p. 2005, 2016.

[21] M. A. Tayal, M. Raghuwanshi, and L. Malik, "Word net based method for determining semantic sentence similarity through various word senses," in *Proceedings of the 11th International Conference on Natural Language Processing*, 2014, pp. 139–145.

[22] D. R. Radev, "A common theory of information fusion from multiple text sources step one: cross-document structure," in *Proceedings of the 1st SIGdial workshop on Discourse and dialogue-Volume 10*. Association for Computational Linguistics, 2000, pp. 74–83.

[23] P. Rashmi, D. Nihkil, L. Alan, M. Eleni, R. Livio, J. Aravind, W. Bonnie *et al.*, "The penn discourse treebank 2.0," in *Lexical Resources and Evaluation Conference*. -, 2008.

[24] W. C. Mann and S. A. Thompson, *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute, 1987.

[25] N. A. H. Zahri, F. Fukumoto, and S. Matsuyoshi, "Exploiting discourse relations between sentences for text clustering," in *24th International Conference on Computational Linguistics*, 2012, p. 17.

[26] A. Louis, A. Joshi, and A. Nenkova, "Discourse indicators for content selection in summarization," in *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2010, pp. 147–156.

[27] S. Verberne, L. Boves, N. Oostdijk, and P. Coppen, "Discourse-based answering of why-questions," 2007.

[28] P. Piwek and S. Stoyanchev, "Generating expository dialogue from monologue: motivation, corpus and preliminary rules," in *Human Language Technologies: The*

2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010, pp. 333–336.

[29] N. A. H. Zahri, F. Fukumoto, M. Suguru, and O. B. Lynn, "Exploiting rhetorical relations to multiple documents text summarization," *International Journal of Network Security & Its Applications*, vol. 7, no. 2, p. 1, 2015.

[30] M.-F. Moens, C. Uyttendaele, and J. Dumortier, "Information extraction from legal texts: the potential of discourse analysis," *International Journal of Human-Computer Studies*, vol. 51, no. 6, pp. 1155–1171, 1999.

[31] M.-C. Marneffe, A. N. Rafferty, and C. D. Manning, "Finding contradictions in text," *Proceedings of ACL-08: HLT*, pp. 1039–1047, 2008.

[32] M. J. Paul, C. Zhai, and R. Girju, "Summarizing contrastive viewpoints in opinionated text," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 66–76.

[33] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, "The third pascal recognizing textual entailment challenge," in *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*. Association for Computational Linguistics, 2007, pp. 1–9.

[34] A. Wyner, R. Mochales-Palau, M.-F. Moens, and D. Milward, "Approaches to text mining arguments from legal cases," in *Semantic processing of legal texts*. Springer, 2010, pp. 60–79.

[35] C. Reed and G. Rowe, "Araucaria: Software for argument analysis, diagramming and representation," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 04, pp. 961–979, 2004.

[36] G. R. Chris Reed, Raquel Mochales Palau and M.-F. Moens, "Language resources for studying argument," in *Proceedings of the Sixth International Conference on*

Language Resources and Evaluation (LREC'08), B. M. J. M. J. O. S. P. D. T. Nicoletta Calzolari (Conference Chair), Khalid Choukri, Ed. Marrakech, Morocco: European Language Resources Association (ELRA), may 2008, http://www.lrec-conf.org/proceedings/lrec2008/.

[37] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proceedings of the 11th international conference on Artificial intelligence and law*. ACM, 2007, pp. 225–230.

[38] R. Mochales-Palau and M. Moens, "Study on sentence relations in the automatic detection of argumentation in legal cases," *Frontiers in Artificial Intelligence and Applications*, vol. 165, p. 89, 2007.

[39] Y. Hong, S. Zhu, W. Yan, J. Yao, Q. Zhu, and G. Zhou, "Expanding native training data for implicit discourse relation classification," in *Chinese National Conference on Social Media Processing*. Springer, 2014, pp. 67–75.

[40] A. Esuli and F. Sebastiani, "Sentiwordnet: a high-coverage lexical resource for opinion mining," *Evaluation*, vol. 17, pp. 1–26, 2007.

[41] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." in *Lrec*, vol. 10, no. 2010, 2010, pp. 2200–2204.

[42] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[43] B. Ohana and B. Tierney, "Sentiment classification of reviews using sentiwordnet," 2009.

[44] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[45] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 759–766.

[46] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in neural information processing systems*, 2013, pp. 935–943.

[47] A. Quattoni, M. Collins, and T. Darrell, "Transfer learning for image classification with sparse prototype representations," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[48] V. Jayawardana, D. Lakmal, N. de Silva, A. S. Perera, K. Sugathadasa, B. Ayesha, and M. Perera, "Word vector embeddings and domain specific semantic based semi-supervised ontology instance population," *International Journal on Advances in ICT for Emerging Regions*, vol. 10, no. 1, p. 1, 2017.

[49] K. Sugathadasa, B. Ayesha, N. de Silva, A. S. Perera, V. Jayawardana, D. Lakmal, and M. Perera, "Synergistic union of word2vec and lexicon for domain specific semantic similarity," *arXiv preprint arXiv:1706.01967*, 2017.

[50] V. Jayawardana, D. Lakmal, N. de Silva, A. S. Perera, K. Sugathadasa, and B. Ayesha, "Deriving a representative vector for ontology classes with instance word vector embeddings," in *Innovative Computing Technology (INTECH), 2017 Seventh International Conference on*. IEEE, 2017, pp. 79–84.

[51] K. Sugathadasa, B. Ayesha, N. de Silva, A. S. Perera, V. Jayawardana, D. Lakmal, and M. Perera, "Legal document retrieval using document vector embeddings and deep learning," 2018.

[52] V. Jayawardana, D. Lakmal, N. de Silva, A. S. Perera, K. Sugathadasa, B. Ayesha, and M. Perera, "Semi-supervised instance population of an ontology using word vector embeddings," *arXiv preprint arXiv:1709.02911*, 2017.

[53] L. Carlson, M. E. Okurowski, and D. Marcu, *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania, 2002.

[54] F. Wolf, E. Gibson, A. Fisher, and M. Knight, "Discourse graphbank," *Linguistic Data Consortium, Philadelphia*, 2004.

[55] "Lee v. United States," in *US*, vol. 432, no. No. 76-5187. Supreme Court, 1977, p. 23.

[56] Z. Zhang, S. Blair-Goldensohn, and D. R. Radev, "Towards cst-enhanced summarization," in *AAAI/IAAI*, 2002, pp. 439–446.

[57] V. Uzêda, T. Pardo, and M. Nunes, "A comprehensive summary informativeness evaluation for rst-based summarization methods," *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM) ISSN*, pp. 2150–7988, 2009.

[58] M. L. d. R. Castro Jorge and T. A. S. Pardo, "Experiments with cst-based multidocument summarization," in *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 74–82.

[59] D. Marcu, "From discourse structures to text summaries," *Intelligent Scalable Text Summarization*, 1997.

[60] D. R. Radev, H. Jing, M. Styś, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management*, vol. 40, no. 6, pp. 919–938, 2004.

[61] K. C. Litkowski, "Cl research experiments in trec-10 question answering." in *TREC*, 2001.

[62] N. A. H. Zahri, F. Fukumoto, and S. Matsuyoshi, "Exploiting discourse relations between sentences for text clustering," in *24th International Conference on Computational Linguistics*, 2012, p. 17.

[63] B. Hachey and C. Grover, "A rhetorical status classifier for legal text summarisation," *Text Summarization Branches Out*, 2004.

[64] ——, "Extractive summarisation of legal texts," *Artificial Intelligence and Law*, vol. 14, no. 4, pp. 305–345, 2006.

[65] D. Radev, J. Otterbacher, and Z. Zhang, "CSTBank: Cross-document Structure Theory Bank," http://tangra.si.umich.edu/clair/CSTBank, 2003.

[66] K. Clark and C. D. Manning, "Entity-centric coreference resolution with model stacking," in *Association for Computational Linguistics (ACL)*, 2015.

[67] E. Schweighofer and W. Winiwarter, "Legal expert system kontermâĂŤautomatic representation of document structure and contents," in *International Conference on Database and Expert Systems Applications*. Springer, 1993, pp. 486–497.

[68] J. J. Nay, "Gov2vec: Learning distributed representations of institutions and their legal text," *arXiv preprint arXiv:1609.06616*, 2016.

[69] G. Ratnayaka, T. Rupasinghe, N. de Silva, M. Warushavithana, V. Gamage, and A. S. Perera, "Identifying relationships among sentences in court case transcripts using discourse relations," *arXiv preprint arXiv:1809.03416*, 2018.

[70] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, no. 2010, 2010, pp. 1320–1326.

[71] J. G. Conrad and F. Schilder, "Opinion mining in legal blogs," in *Proceedings of the 11th international conference on Artificial intelligence and law*. ACM, 2007, pp. 231–236.

[72] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.

[73] Y. Lu, H. Wang, C. Zhai, and D. Roth, "Unsupervised discovery of opposing opinion networks from forum discussions," in *Proceedings of the 21st ACM*

international conference on Information and knowledge management. ACM, 2012, pp. 1642–1646.

[74] R. T.-W. Lo, B. He, and I. Ounis, "Automatically building a stopword list for an information retrieval system," in *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*, vol. 5, 2005, pp. 17–24.

[75] C. Van Rijsbergen, "Information retrieval. dept. of computer science, university of glasgow," *URL: citeseer. ist. psu. edu/vanrijsbergen79information. html*, vol. 14, 1979.

[76] B. Santorini, "Part-of-speech tagging guidelines for the penn treebank project (3rd revision)," *Technical Reports (CIS)*, p. 570, 1990.

[77] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[78] V. Gamage, M. Warushavithana, N. de Silva, A. S. Perera, G. Ratnayaka, and T. Rupasinghe, "Fast approach to build an automatic sentiment annotator for legal domain using transfer learning," *arXiv preprint arXiv:1810.01912*, 2018.

[79] G. Ratnayaka, T. Rupasinghe, N. de Silva, M. Warushavithana, V. Gamage, and A. S. Perera, "Identifying relationships among sentences in court case transcripts using discourse relations," *arXiv preprint arXiv:1809.03416*, 2018.

[80] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "Open language learning for information extraction," in *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*, 2012.

[81] D. Chen and C. Manning, "A fast and accurate dependency parser using neural

networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740–750.

[82] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in neural information processing systems*, 2013, pp. 926–934.

[83] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 173–180.

[84] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet:: Similarity: measuring the relatedness of concepts," in *Demonstration papers at HLT-NAACL 2004*. Association for Computational Linguistics, 2004, pp. 38–41.

[85] D. Lin *et al.*, "An information-theoretic definition of similarity." in *Icml*, vol. 98, no. 1998. Citeseer, 1998, pp. 296–304.

[86] J. A. Harris and C. Potts, "Perspective-shifting with appositives and expressives," *Linguistics and Philosophy*, vol. 32, no. 6, pp. 523–552, 2009.

[87] M.-C. De Marneffe and C. D. Manning, "Stanford typed dependencies manual," Technical report, Stanford University, Tech. Rep., 2008.

[88] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.

[89] D. R. Radev, J. Otterbacher, and Z. Zhang, "Cst bank: A corpus for the study of cross-document structural relationships." in *LREC*, 2004.

[90] "Turner v. United States," in *US*, vol. 396, no. No. 190. Supreme Court, 1970, p. 398.

[91] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[92] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[93] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.

[94] "Google cloud including gcp & g suite âĂŤ try free Âă|Âă google cloud," https://cloud.google.com/, (Accessed on 11/11/2018).

[95] "Digitalocean: Cloud computing, simplicity at scale," https://www.digitalocean.com/, (Accessed on 11/07/2018).

[96] "Laravel - the php framework for web artisans," https://laravel.com/, (Accessed on 11/07/2018).

[97] "Scrapy | a fast and powerful scraping and web crawling framework," https://scrapy.org/, (Accessed on 11/07/2018).

[98] "Gojs diagrams for javascript and html, by northwoods software," https://gojs.net/latest/index.html, (Accessed on 11/11/2018).

[99] "Latex - a document preparation system," https://www.latex-project.org/, (Accessed on 11/13/2018).