Internet Technologies and Information Systems (International Master's Program)

Master Thesis

Submitted per degree requirements, for Master of Science at Leibniz University Hannover

N-ary Relation Extraction from News Articles

Gul Jabeen

*L*3S Research Center
Appelstrasse 4
30167 Hannover, Germany

L3S Research Center

27th February 2020

Internet Technologies and Information Systems (International Master's Program)

Master Thesis

Submitted per degree requirements, for Master of Science at Leibniz University Hannover

N-ary Relation Extraction from News Articles

Author: Gul Jabeen
Thesis Advisor: Besnik Fetahu
1st Examiner: Prof. Dr. Wolfgang Nejdl
Second Examiner: Prof. Dr. Avishek Anand
Submission Date: 27th February 2020

I, hereby, declare that this thesis is entirely the result of my own work, except where otherwise indicated. The resources used are given in the list of references.

February 2020 Gul Jabeen

Acknowledgements

First and foremost, I would like to thank my thesis advisor, Besnik Fetahu, from the department of L3S Research Center, without whose guidance, I wouldn't have made the best out of the opportunity to work on this topic. The topic is quite credible and attractive, during the research of which, I gained a lot of knowledge and experience. I would also like to thank Lijun Lyu, from the department of L3S Research Center, without whose guidance and input, this thesis could not have been successfully completed.

People sometimes ask me if it is a sin in the Church of Emacs to use vi. Using a free version of vi is not a sin; it is a penance. So happy hacking.”
-Richard Stallman

Abstract

Information Extraction (IE) involves extracting information from unstructured data such as entities, relations and events. IE and its sub-task relation extraction, play a central role in data processing, which enables and enhances the visibility of the hidden knowledge. Entity extraction has a task to extract entities from the text. These entities could be an organization or a location and the like, whereas relation extraction finds a semantic link between such entities which can be either binary or n-ary. In this dissertation, the focus is on n-ary relation extraction, to identify n-ary (n>2) relations among multiple arguments from sentences. This thesis presents three main contributions to the essential questions of the corresponding research.

The first contribution is data scraping from Wikipedia news articles using web-scraper, which further deals with extraction of entities and relations from news events by using constituency parser from natural language processing toolkit. The second contribution presents the annotation interface for annotation purposes. Finally, the third contribution pertains to annotated arguments and entities from the annotation interface, passed through the relation extraction framework based on Long Short Term Memory (LSTM) and Multi-Task Learning (MTL). The model has been evaluated on datasets and its performance is measured on the evaluation metrics, demonstrating the model's effectiveness with supervised learning and MTL, which has significantly improved the extraction accuracy.

Keywords: Information extraction, relation extraction, entity extraction, n-ary relation extraction, constituency parser, annotation interface, Long Short Term Memory (LSTM) and Multi-Task Learning (MTL).

Contents

List of Figures

List of Tables

1 Introduction

Information extraction is a process of finding patterns and collecting data from unstructured data. The purpose of information extraction is to transform raw data into meaningful information that can be used for analytical purposes. It extracts entities and relations among those entities and links the information together into a more coherent framework. Examples include natural language processing, information retrieval, machine learning, databases, document analysis and web mining.

Name Entity (NE) and Relation Extraction (RE) are two fundamental tasks of information extraction. NE has a task of finding entities in a text such as a person, location, organization and country; relation extraction is a task of determining semantic links between those entities. RE is typically addressed as a classification problem for pairs of entities. There are different methods proposed to address RE. In contrast to the traditional rule-based method, this dissertation explores machine learning techniques for the extraction of relations. Machine learning is aimed to analyze patterns in data and then train a model, in the case of extraction of entities and relationships. There are two types of RE, binary relations and n-ary relation extraction. Binary relation extraction is between two entities, whereas n-ary relation extraction defines more than one relation among multiple entities. In this dissertation, we have focused on n-ary relation extraction.

This section is further elaborated into four sub-sections; Section 1.1 illustrates the introduction of n-ary relation extraction, Section 1.2 defines the problem statements of this thesis. Section 1.3 provides a thesis outline and Section 1.4 demonstrates the contributions of the thesis such as dataset extraction, annotation interface and implementation of neural networks for n-ary relation extraction.

1.1 Introduction to the N-ary Relation Extraction

The purpose of relation extraction is to identify relationships between name objects (e.g., hiredby (PersonX, OrganizationY)). This relationship interprets the existing predicates associated with any two argument. An arguments represents any object, concept or people in the actual world and the relation predicate describes the type of stative connection or interaction that exists between the things represented by the arguments. N-ary relation extraction detects relations among n-entities across multiple sentences (Song et al., 2018). The task of relation extraction (RE) aims to identify a relation between two entities. Research in the field of relation extraction is primarily focused on binary relation extraction as compared to n-ary relation extraction (Bach and Badaskar, 2017). The difference between binary and n-ary relation, is that a binary relation exists between two entities in a single sentence, whereas n-ary relation can exist between more than two entities.

Modern analytical tools, question answering chatbots, and summarizing systems require a higher order of relations to provide enriched information. Binary relation gives precise information, which is not efficient enough for such systems, because of which, n-ary relation extraction is needed to improve the credibility of such systems. Figure 1 illustrates the comparison between binary relation extraction and n-ary relation extraction. It shows a binary relation existing between two entities, *'first nominating contest'* (e1) and *'Democratic Party '* (e2), whereas complex n-ary relation existing between five (n = 5) entities mention namely *'first nominating contest'* (e1) , *'Democratic Party '* (e2), *'Republican Party '* (e3), *'place'* (e4) and *'U.S. state '* (e5) which gives more enriched information compared to binary relation extraction. The n-ary relation extraction can extract more relations compared to binary, which limits the information extraction. Figure 1 also highlights the importance of n-ary relation extraction, which extracts more than one relation among multiple entities as compared to the binary relation extraction, which extracts only one relation between two entities. It is crucial to extract all the arguments in the sentence because each argument contains some information. If any argument is missing, then the data extraction becomes less knowledgeable. For instance, in Figure 1, binary relation has two arguments, and n-ary extraction has all the related arguments and relations in a sentence, it is not missing any information. Hence, n-ary relation gives a more enriched data and complete information.

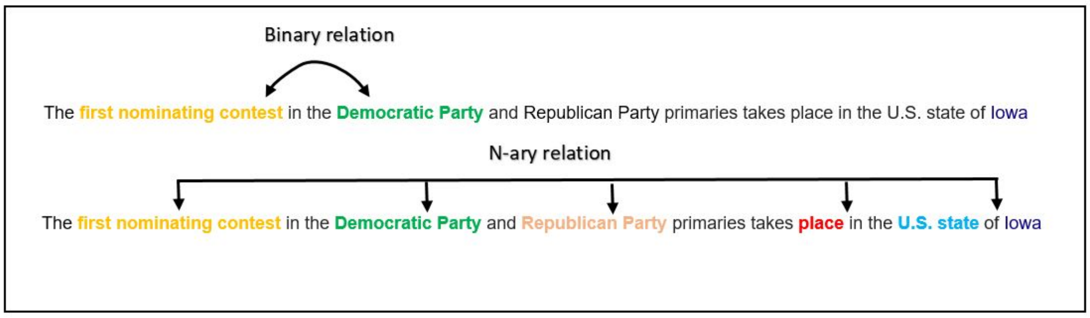

Binary relation
The first nominating contest in the Democratic Party and Republican Party primaries takes place in the U.S. state of Iowa
N-ary relation
The first nominating contest in the Democratic Party and Republican Party primaries takes place in the U.S. state of Iowa

Figure 1: Binary vs n-ary relation extraction

n-ary relation extraction is challenging in many ways; first, the existence of multiple entities in a sentence, having correlated relations, makes the task more challenging. Figure 1 demonstrates that the sentence has multiple entities in n-ary relation extraction, a valid affiliation relation instance can be given as a binary $<$ e1, e2 $>$, ternary $<$ e1, e2, e3 $>$, qua-ternary $<$ e1, e2, e3, e4 $>$ or a 5ary $<$ e1, e2, e3, e4, e5 $>$ tuple with possibilities of different permutations of entities for each tuple type. Additionally, it shouldn't consider irrelevant tuples like $<$ e1, e4 $>$. The second challenge is to extract relevant information from a sentence having a different syntactic structure, and third, to improve the accuracy of extraction as the best accuracy obtained by relation and event extraction systems hovers in between the range of 70-80%.

In this thesis, we consider deep learning approaches and manual annotation process in order to overcome the above-mentioned challenges. First, we developed a human-based annotation tool to annotate entities and relations in a sentence. The resulting annotations are used as training data for the deep learning model to learn n-ary relations among multiple entities. These approaches have increased the credibility and accuracy of n-ary relation extraction task, which has been mentioned in Section 9.

1.2 Problem Setup

In order to motivate the specific contributions of the dissertation, this section introduces the research problems of n-ary relation extraction. These problems are concerned with the utilization of linguistic patterns to expose relational information in texts. Patterns are rules that define template of sentences. Patterns determine what type of relations exist in a text and relational arguments which are located in sentences. Relation extraction systems often exploit extraction rules with different underlying sentence models.

Research Problem 1: How can we extract arguments and relations from sentences? Argument extraction is an event extraction task, which finds entities in a text, whereas, relation extraction defines a relation among those entities which can be binary or n-ary relation. It is crucial to extract arguments and relations from the sentence for the training of the models to extract n-ary relations.
Research Problem 2: How can we model the task of n-ary relation extraction through machine learning or information extraction models? The second problem is to model the task of n-ary relation extraction using some model which can find n-ary relation among sentences using some dataset which contains sentences, arguments and relations label.

Research Problem 3: How can these models assess efficiency, quality, relevance, and diversity? The third problem is to improve the accuracy of relation extraction and provides enriched data using n-ary relation extraction, compared to binary relation extraction.

1.3 Thesis Outline

Section 2 This section provides all the necessary background information required to understand the research basis. Firstly, it describes supervised machine learning techniques and deep learning models which have been used for the purpose of relation extraction in this thesis. Secondly, it highlights the natural language processing tasks which have been used for the extraction of noun-phrases from the constituency parser for this research. Lastly, it demonstrates the background knowledge of word embedding and its types.

Section 3 The related work of binary relation extraction and n-ary relation extraction are discussed in this section. It reviews related work in n-ary relation extraction using different models and techniques.

Section 4 This section describes the tasks behind dataset collection. Firstly, it shows the analysis of Wikipedia current events portal. Secondly, it illustrates the event extraction task and pre-processing of the data. Lastly, it explains the event type distributions present in the dataset collection.

Section 5 This section demonstrates the architecture and intuitions behind the annotation interface. Firstly, it introduces all the necessary steps for the preparation of the interface; then it describes the annotation protocol of how annotation can be done using this interface. Lastly, it reviews all the drawbacks and challenges held during the preparation of the interface.

Section 6 This section explains the ground-truth collection which includes expert annotations and crowdsourced annotations. Firstly, it describes the purpose of expert annotations. Secondly, it explains the crowdsourced ground-truth collection which has been annotated by crowdsourced workers using annotation interface mentioned in the previous section.

Section 7 The major contribution of this research is explained in this section which explains intuitions behind n-ary relation extraction task. Firstly, it discusses the importance of multi-task learning and the benefit of using it. Then, it introduces the model setup and model training for the n-ary relations extraction which mention how two models can work simultaneously in order to extract n-ary relations.

Section 8 The demonstration of the experimental setup is explained in this section which includes evaluation metrics, evaluation models including with/without pre-trained word embedding, and evaluation of datasets.

Section 9 This section provides evaluation results and their discussion. Furthermore, it explains the model robustness using different test and training sizes. Lastly, it describes ablations and limitations of these models.

Section 10 The conclusion and future work of this research are mentioned in this section.

1.4 Thesis Contributions

In this section, we described main contributions of this thesis.

1.4.1 n-ary Dataset

The first contribution presented in Section 4 is a dataset collection of Wikipedia current events, which has been extracted using web-scraper using two libraries named Requests and bs4 (Beautiful Soup), which serves to scrape all the required data from the website, for example, event summary, date, event title, etc.

Furthermore, the event summary is used for relation extraction. The noun-phrases have been extracted from the constituency parser as arguments, which, in turn, are used in the annotation interface to identify which essential arguments should be kept and what relation exists among these arguments. This aspect is discussed in Section 2.2.4.

1.4.2 Annotation interface

Section 5 proposes the second contribution to this dissertation which is the implementation of annotation interface for crowd workers to select important arguments and identify the relation between those arguments. This solution aims to identify relevant and possible arguments in a sentence. The selected arguments and relations are not solely based on machine annotations anymore.

Section 6 presents the third contribution, which is based on two parts; expert annotations and crowdsourced ground-truth collection. We have generated some annotations from the interface as expert annotations, to identify the quality of crow-sourced annotations. Second, we set some ground-truth conditions for the collection of crowdsourced data, in order to improvise the quality of the annotations; each annotation is annotated by three annotators and must have selected expert annotations, as mentioned above.

1.4.3 Insights on how different neural networks work

The fourth contribution in Section 7 is a benchmark for n-ary relation extraction methods. We have implemented the multi-task learning model in order to achieve goals. We defined an experimental setup to be used on all methods in Section 8. This way we expect to impartially evaluate the selected methods in Section 9 and analyze their performance.

Figure 2 shows the overall architecture of the system, which shows a combined representation of all contributions in this thesis. Data preparation represents the first main contribution of the thesis, which includes data extraction, pre-processing tasks and the annotation interface. The main contribution of this research is n-ary relation extraction which has used multi-task learning for identifying n-ary relations among relations and arguments which have been annotated by crowd-workers.

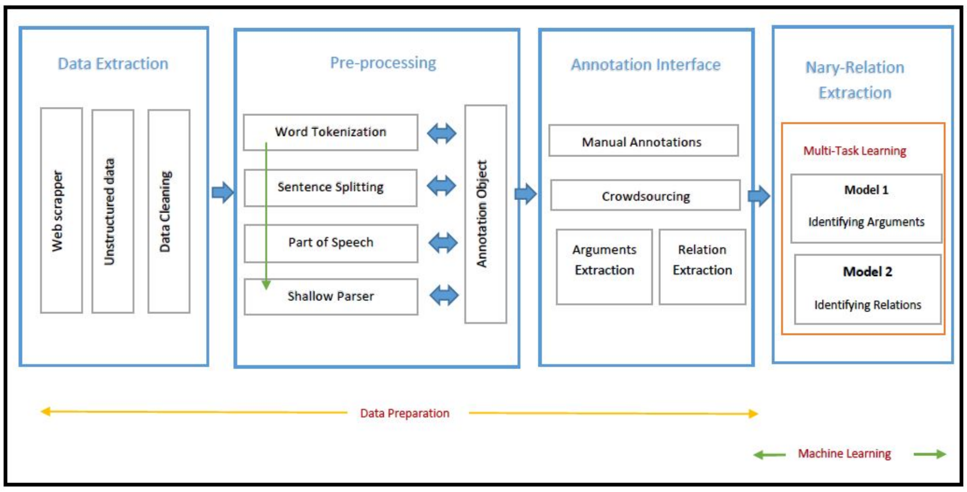


Figure 2: System architecture

2 Background

This section contains the background information necessary for understanding the rest of the dissertation. First, we describe the supervised machine learning tasks and basic terminologies—next, the methods used in the research- RNN and Bi-LSTM. Finally, the syntactic and semantic natural language processing tasks will be elaborated upon, used in the dissertation for the entity extraction task and preparation of semantic annotation tool.

2.1 Supervised Machine Learning

2.1.1 Basic definitions and terminologies

Supervised machine learning

The main aim of supervised learning is to learn from the labeled dataset. The dataset consists of inputs and targets which can be numeric values or string labels and make predictions in the future about unseen data. Here the labeled data implies tagged data with one or more labels identifying specific properties. Supervised algorithms try to model the relationships and dependencies between the target variable and the input variable in order to predict the target variable in the future based on the input variable (Mohri et al., 2012). Supervised learning has been used in many practical applications, such as sentiment analysis (movie review rating), news classification, video surveillance, image classification, fraud detection, email classification and spam filtering. For instance, sentiment analysis (movie reviews) is one of the common application to identify whether a review is good or bad. The model first learns inputs and their corresponding labeled review, which correctly classifies as bad or good. Then it predicts new review label from the learned data.

Types of supervised learning

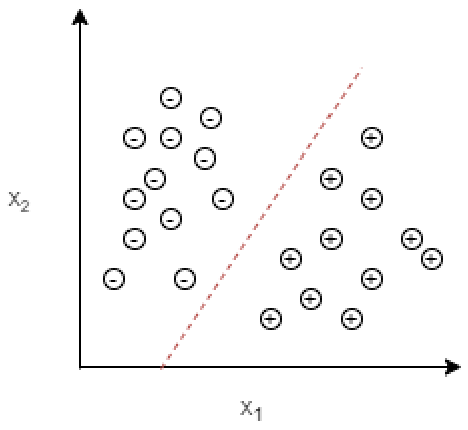
Supervised learning has two types, classification and regression problems. In regression problems, the target remains continuous, whereas in classification problems, the target is a qualitative variable .

1. **Classification Problem**

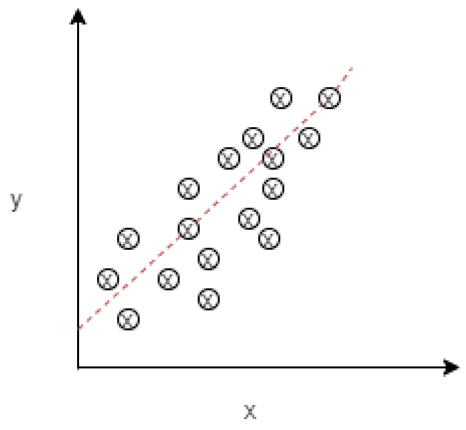
   Classification is a type of problem where we predict categorical labels which could either be binary or multi-class labels. Figure 3a illustrates the binary classification problem of the two-dimensional dataset.

2. **Regression problem**

   Regression, on the other hand, is a type of problem where we predict the continuous target variable (outcome or target) based on the given number of explanatory variables, for example, predicting the salary of a person based on the years of experience, qualification, age, and university. In Figure 3b, x is the predictor variable and y is the response variable. A straight line is fit to this data that minimizes the mean squared distance between the sample data points and the fitted line. Using the intercept and slope from the data, outcome variable or unknown data can be predicted (Mohri et al., 2012).



(a) Classification



(b) Regression

Figure 3: Supervised learning

The focus of this thesis is based on supervised learning classification problem to classify the data against argument or relation labels. The overall focus is on binary classification. We have used a classification problem with deep learning methods, RNN and LSTM models to do the tagging in order to get enriched data.

2.1.2 Deep neural networks

Deep learning belongs to a family of artificial intelligence and machine learning methods, which typically use many hierarchical layers of non-linear processing (Deng and Yu, 2014). It uses the multi-layered artificial neural network that use training data to train the model, which gives state-of-the-art accuracy in the task. In this thesis, the primarily focus is on RNN and LSTM which are explained below.

Recurrent Neural Networks (RNN)

Neural networks try to mimic how the brain functions, designed to recognize patterns in data using past and current knowledge. These networks recognize numerical patterns in all real-world data (images, sound, text or time series) which consist of vectors. Neural networks are the composition of the highly interconnected neurons that work together to solve a problem or to make a decision. Recurrent neural networks process the sequential data such as text, which can send feedback signal, as shown in Figure 4. (Chen, 2016)

Neural networks have three major problems.

1. They cannot capture long-term dependencies
2. It has vanishing and exploding gradient problem because weights become zero or explode due to the product of partial differentials.
3. Its inference is slow.

RNNs are more capable of capturing long term dependencies than neural networks, for instance, Multi-layer Perceptron (MLP). Neural networks are composed of neurons like the human brain; they use their previous knowledge to produce current decisions whereas simple neural networks cannot store previous knowledge. Recurrent neural networks have an internal memory to store the past computations. In contrast, the output of the current input depends on the previous input. After producing the output, it sends back a signal into the recurrent network. For making a decision for the next output, RNN considers considers both the current input and previous knowledge.

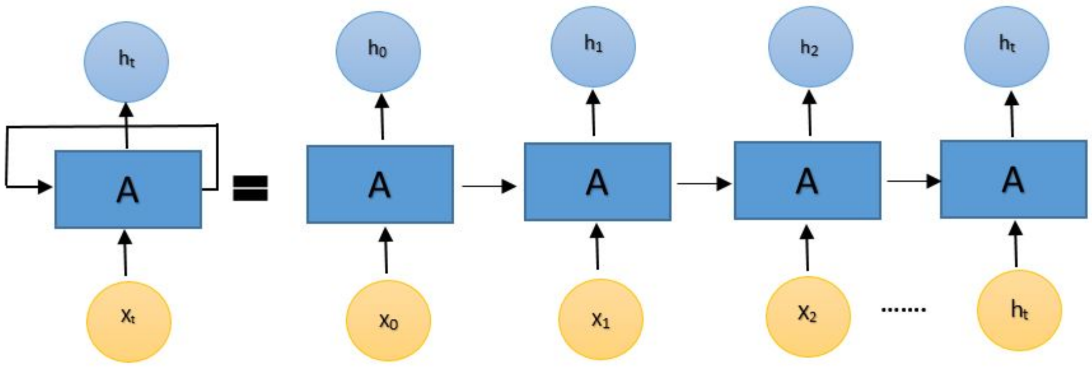


Figure 4: An unrolled recurrent neural network (Chen, 2016)

RNN can take more than one input vector as well as it can produce more than one output vector. The outputs from the RNN are influenced just not by weights but also by the hidden layer state vector, which represents the partial differential of inputs and weights. Therefore, input produces more outputs based on the previous given knowledge. As shown in the above figure, first RNN sends first input sequence of the X(0) to the output h(0), which will combine with X(1) of input, and become the input for the next step. It will pass from the activation function, and generate output. An activation function has the aim to determine whether a neuron should be activated or not; the activation function calculates the weighted sum and adds further bias to it. That introduces non-linearity into the output of a neuron — there are different kind of activation functions, e.g., ReLU, sigmoid. Similarly, h(1) will combine with the next input X(2) for the next step. In this way, it will remember the sequences of the previous inputs while training the data.

The formula for the current state is:

$$h_t = f(h_{t-1}, X_t)$$

Applying activation function:

$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}X_t)$$

Where;

W: weight
h: hidden vector
W_{hh}: weight at previous hidden state
W_{hx}: weight at current input state
tanh: Activation funtion

Output state is y_t :

$$y_t = W_{hy} h_t$$

Sometimes RNN does have gradient vanishing and exploding problem, which means gradient/derivative of weight and inputs gets closer to zero or get larger. It is not able to process very long sequences if using activation functions tanh or Relu. In order to resolve this issue, this research has focused on Long Short-Term Memory (LSTM) which can overcome these issues because of the addition of extra gates to the model. It is a special kind of recurrent neural network which works better than a standard version.

Long Short-Term Memory (LSTM)

LSTM is the modification and extension of the recurrent neural networks, which resolves the issue of long term dependencies and vanishing or exploding gradient. It uses the back-propagation for the training of the model. LSTM is composed of the cell, which has three different gates, input gate, forget gate and output gate, as shown in Figure 5.

1. **Input Gate**

   The input gate decides which input value should be used to modify the memory. It uses two activation functions for this purpose, sigmoid function and tanh function. The sigmoid function generates output between 0 and 1, deciding which value would be likely to update. In the next step, tanh function squishes values to always be between -1 and 1, which describes the level of importance of each word. Then both gates combine to create an update to the state.

2. **Forget Gate**

   In this step, forget gate decides which data should be removed from the block. This decision is taken by a sigmoid function, which looks at the previous state (h_{t-1}) and input (X_t) and gives output between 0 and 1 or each number in the cell state C_{t-1}. The closer to 0 means to forget, and the closer to 1 means to keep.

3. **Output Gate**

   The output state decides the next hidden states. First, it passes the previous hidden state and the current output state into a sigmoid function that decides which values would pass through 0 and 1. The newly modified cell state passes to tanh function. Then the tanh output and sigmoid output multiply to decide what information the hidden state should carry.

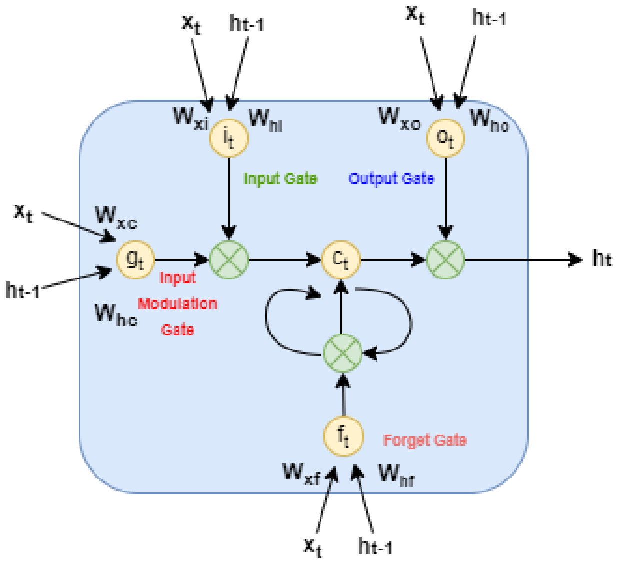


Figure 5: LSTM Networks (Chen, 2016)

The computations of these gates and cell state are given below: (Chen, 2016)

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$$

$$g_t = tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg})$$

$$O_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$$

$$C_t = f_tC_{t-1} + i_tg_t$$

$$h_t = O_t tanh(C_t)$$

Where as;
x_t The memory cell input at time step t;
h_t The hidden layer value of memory cell at time step t;
σ Sigmoid function, output a value between 0 to 1;

Bidirectional LSTM

Bi-directional LSTM consists of two LSTMs, a forward and backward based sequence, further connected to the output layer. It accesses the past (left) and future (right) context of the word. However, simple LSTM can only save past information h_{t-1}. But Bi-LSTM can track the data on the both sides of the word, which omits the problem of fixed context. Both LSTMs, forward and backward concatenate their outputs, the hidden state sequence $(\overrightarrow{h_i})$ of forward-based LSTM and hidden state sequence $(\overleftarrow{h_i})$ of backward-based LSTM get combined $h_t = [\overrightarrow{h_i}, \overleftarrow{h_i}] \in R^m$ at each time, then the complete hidden state sequence is formed: $(h_1, h_2, ..., h_n) \in R^{n \cdot m}$

In the next step, the m-dimensional hidden state vector is mapped to k-dimensional (k is the number of labels). The extracted sentence features expressed as matrix $O = (o_1, o_2, ..., o_n) \in R^{n \cdot m}$.

Each dimension o_{ij} of $o_i \in R^k$ can be marked as the scoring value that the word x_i is classified as j_{th} tag.

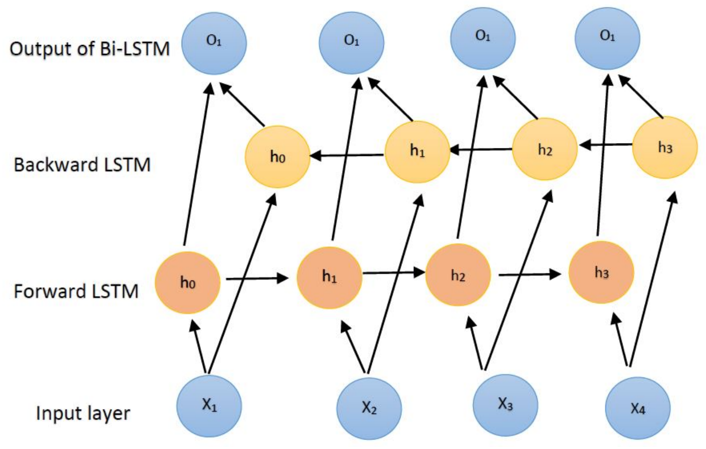


Figure 6: Bi-directional LSTM (Huang et al., 2015)

2.2 Syntactic and Semantic Natural Language Processing Tasks

Natural Language Processing (NLP) is a part of artificial intelligence that deals with the interaction between computers and humans using the natural language. NLP perform several automated tasks such as automatic summarization, sentiments analysis, speech recognition (Siri, Google Assistant, Alexa), bank systems to analyse credit worthiness assessment, chatbots and many other applications pertaining to the real-world. The application of NLP can find names of people and companies in free texts and can link the names to public records or a directory of the company. (Jackson and I., 2007)

In this dissertation, we have used Stanford CoreNLP Natural Language Processing Toolkit (Stanford CoreNLP) for following tasks; tokenization, sentence splitting, part of speech tagging, and shallow parser, also known as constituency parser which is the most important part in this thesis for the event extraction task.

The following subsections describe syntactic and semantic natural language processing tasks that are used for the annotation of text corpora with relevant features.

2.2.1 Tokenization and sentence splitting

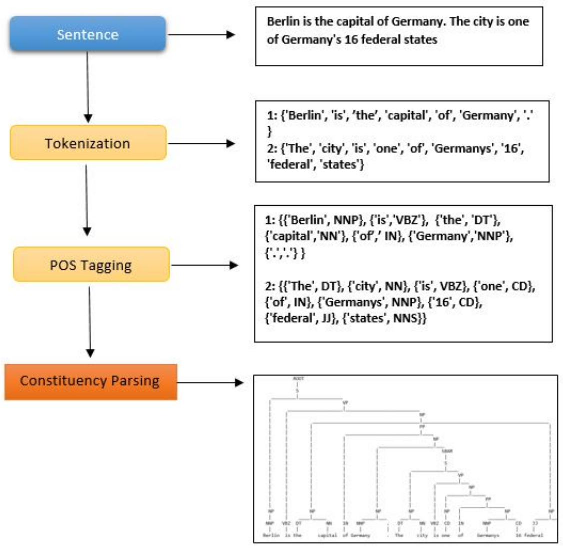
Tokenization is the first step of the natural language processing task which tokenizes the sentence into the sequence of tokens. Sentence splitting is often used in combination with the tokenization at the same time. The benefit of using Stanford Corenlp is it can split the text into sentences and apply all functions on it separately as shown below in Figure 7.

2.2.2 Part of Speech Tagging

Part of speech tagging annotates each token with their part of speech, such as noun, verb, adjective based on its content and definition. Figure 7 demonstrates the step of POS tagging in which each token has a part of speech.

2.2.3 Constituency Parsing

Constituency parsing assigns syntactic structure to the tokens. Constituent parsing transmits constituent parts of sentences (noun, verb, adjective) to the constituency parser which connects to the higher-order units that have more grammatical meaning such as noun phrases, preposition phrases and verb phrases as shown in Figure 7. In this thesis, we are more concerned about noun phrases in order to do the extraction of the argument task that has been explained in Section 5.



```
Berlin is the capital of Germany. The city is one
of Germany's 16 federal states

1: {'Berlin', 'is', 'the', 'capital', 'of', 'Germany', '.'
}
2: {'The', 'city', 'is', 'one', 'of', 'Germanys', '16',
'federal', 'states'}

1: {{'Berlin', NNP}, {'is','VBZ'}, {'the', 'DT'},
{'capital','NN'}, {'of',' IN}, {'Germany','NNP'},
{'.','.'}}

2: {{'The', DT}, {'city', NN}, {'is', VBZ}, {'one', CD},
{'of', IN}, {'Germanys', NNP}, {'16', CD},
{'federal', JJ}, {'states', NNS}}
```

Figure 7: Step by step natural language processing

2.2.4 Extraction of noun phrases

In order to extract arguments from sentences, we have extracted the noun phrases from the constituency parser, considering only noun phrases that do not constitute other noun phrases. These extracted noun phrases have been further used in the preparation of the semantic annotation tool, which was described in Section 5. For instance, in Figure 8, all highlighted boxes are the noun phrases which do not contain other noun phrases such as Berlin, the capital, Germany, the city, one, Germanys 16, Federal.

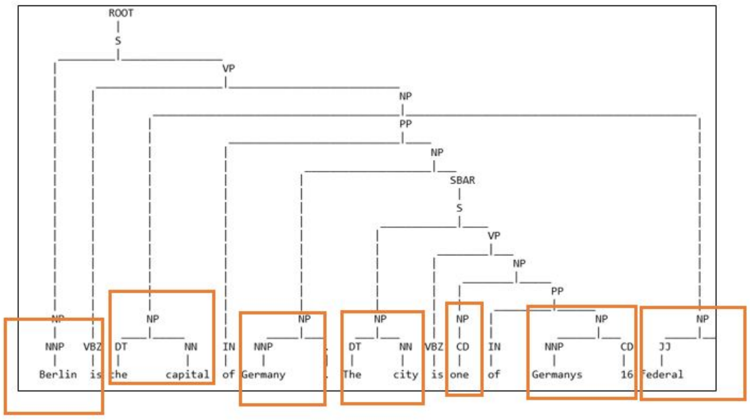


Figure 8: Extraction of noun phrases closer to POS tags

2.3 Word Representations

Word embeddings are numerical representations of words that allows words with similar meaning to have a similar representation. Why do we need word representations? Machine learning algorithm and deep learning architectures are incapable of processing text; they can only take numerical data as input to perform algorithm tasks such as classification and regression.

There are two standard ways of representing words as tokens, namely one-hot and word vector representations in which vector is defined for the entire vocabulary with value 1 at the index of the corresponding word. This representation cannot keep track of the growing dimensions and can build a higher-dimensional vector. A solution to this problem is to use low dimensional vectors that store most of the critical information in a fixed and a small number of dimensions (usually around 25-100). This idea gives rise to the introduction of word vectors (Collobert and Weston, 2008). Ever since, this idea has been used in many word vector models like (Bengio et al., 2009), Word2Vec (Turian et al., 2010a), (Mikolov et al., 2013a), (Mikolov et al., 2013b.) and GloVe (Pennington et al., 2014). The word vector model is the faster approach which easily incorporates new words and predicts surrounding words of every word in a window of length m.

In 2009, Mikolov et al. proposed an approach to train a model in two steps: first, vector representation of words using a simple model and second, train a language model on the top of these distributed representations of words. There are two types of predictive models in word2vec models; skip grams and Continuous Bag-of-Words (CBOW), further discussed below.

2.3.1 Skip-Gram word2vec model

Mikolov et al. 2013a; 2013b. took an inspiration from the above idea (Mikolov et al., 2009) , and proposed a model, Skip-Gram word2vec model which is only focused on the learning of word embedding. The main idea behind this skip-gram model is to predict surrounding words in a context window of length c for every given word w in a corpus, for every possible context windows t in a corpus. Formally, to maximize the average log probability of any context word given the current center word:

$$L(\theta) = 1T \sum_{t=1}^{T} \sum_{c \leq j \leq c, j \neq 0} \log P(w_{t+j} | w_t; \theta)$$

In the above formula, the second summation represents the sum over all the words in contexts windows up to c many words away from the center word, ignoring the center word.

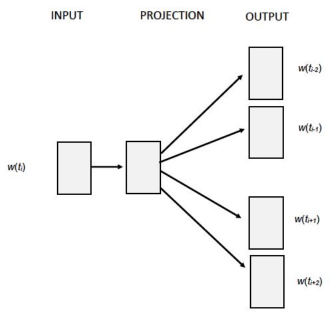


Figure 9: Skip-Gram word2vec model

2.3.2 CBOW word2vec model

Similar to the Skip-Gram, the Continuous Bag-of-Words (CBOW) model, which aggregates the outside words and predicts the center word (Mikolov et al., 2013a), instead of predicting the surrounding words around a given center word. Both the Skip-Gram and the CBOW have a single-layer architecture based on the inner product between two word vectors, and both use the context of a word, instead of only the preceding words.

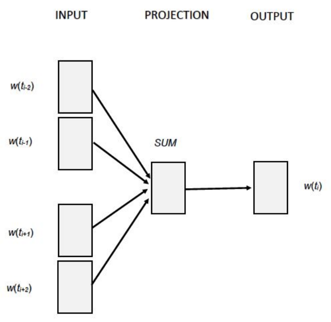


Figure 10: CBOW word2vec model

In this thesis, we used glove embedding (Pennington et al., 2014) for the multi-task learning in Section 7. Both CBOW and Skip-Ngrams take local context into account; they do not take advantage of the global context. Glove embedding has the same intuition behind its tasks; the co-occurrence matrix uses distributed embeddings, then it decomposes the matrix into more dense vectors using neural methods. GloVe embeddings are a kind of pre-trained embeddings which are faster and easier to use.

3 Related work

In this section, we examined the related work of binary and n-ary relation extraction with the aim of identifying state-of-the-art methods and current challenges. The prior work on relation extraction has been primarily focused upon binary relations, on a single sentence. In this thesis, we have extracted multiple entities and relations (n>2) within a sentence or consecutive sentences using Bi-LSTM, natural language processing toolkit and multi-task learning which has been explained further in Section 2. However, this section demonstrates the related work on binary relations and n-ary relation extraction.

3.1 Binary relation extraction

In *binary relation extraction*, the aim is to extract relations between a pair of entities within a sentence. Kambhatla (2004) proposes a statistical technique for extracting relations, which combines diverse, lexical, syntactic and semantic features. GuoDong et al. (2005) presented a feature-based approach for relation extraction which combines diverse, lexical, syntactic and semantic knowledge in which the exploration is based on phrase chunking information instead of a full parse tree, which improves the performance of the model.

Suchanek et al. (2006) has extended the pattern matching approach for information extraction by using deep learning linguistic structures, instead of shallow text patterns which gives deep linguistic structures. Chan and Roth. (2010) has improved the relation extraction by using background knowledge such as relationship among target relations. Etzioni et al. (2011) has focused on identifying more meaningful relationships in a text. There is an introduction of two open information extraction systems, Reverb and R2A2, which requires only shallow syntactic features and both systems double the area under precision/ recall curve compared to previous first-generation open information extraction systems. Nguyen and Grishman (2014) has solved the problem of performance loss when the model is applied to any out-of-domain data, by evaluating word embedding and clustering an adapting feature-based relation extraction system which demonstrates the effectiveness of regularisation for the adaptability of relation extractors.

3.2 N-ary relation extraction

N-ary relation extraction has a task to find relations between more than two entities. This relation extension makes the mathematical definition of 'relations' a little more widely useful, and it is also a foundation for relational databases in computing. Chinchor (1998) has extracted news articles using n-ary relations using deep learning techniques. McDonald et al. (2005) applied n-ary relation extraction to biomedical text, which has followed two stages. First, the creation of a graph from pairs of related entities and then a recreation of the complex relations by finding maximal cliques in a graph that represents relations between the entities.

FitzGerald et al. (2015) and Roth and Lapata. (2016) have applied neural networks to semantic labelling. Following work has decomposed the n-ary relation into binary relation using feed-forward networks.

3.2.1 Cross sentence relation extraction

The cross sentence relation extraction has given an advantage to many relation extraction tasks. Swampillai and Stevenson. (2011) has used cross extraction for MUC facts and also event extraction which demonstrates a composite kernel approach to inter-sentential relation extraction, can achieve comparable results with intra-sentential relation extraction. Wick et al., 2006 also used the same technique on a different dataset; they extracted records from webpages. Yoshikawa et al. (2006) have extracted facts for biomedical domains, which incorporates co-reference relations through the concept of salience in discourse and transitivity. It followed the implementation of two models, SVM pipeline and MLN joint, which improves the intra-sentence and cross sentence related to co-reference relations. Gerber and Chai (2010) used cross sentence relation extraction to extend the semantic role that recovers implicit inter-sentential arguments with a supervised classification model. This recovery could benefit many Natural Language Processing (NLP) applications.

The cross sentence relation has been recently learned with distant supervision. However, it fixates only on binary relations. Numerous papers have been published, which solely work on relation extraction via distant supervision, which has applied to both binary and n-ary relation extraction.

3.2.2 N-ary Relation Extraction using Distant Supervision

Distant supervision occurs in domains where unlabeled data is plentiful and there exists a source of structured labeled data. Mintz et al. (2009) has applied distant supervision to extract such binary relations which accumulates a large number of reasonable relation having high precision patterns. Poon et al. (2014) also proposed a method which extracts binary relation using distant supervision. The technique has been applied on PubMed to extract biological pathways which proved to be workable; which have extracted many pathways interaction in a context. Reschke et al. (2014) and Xu et al. (2015) have extracted n-ary relations using distant supervision, which gives better performance along with more relations.

3.2.3 N-ary relation extraction using multi-task learning

Collobert and Weston (2008) proposed a unified architecture for natural language tasks such as part of speech tags, chunks, named entity tags, and semantic role. The entire network is trained jointly on these tasks using weight sharing, an instance of multi-task learning. Pengl et al. (2017) explored the relation extraction framework using long short term memory networks (LSTM graph) in which it extracts more than two relations in multiple sentences. The framework has been generalized for cross sentence n-ary relation extraction based on graph LSTM using multi-task learning. However, the framework has evaluated two main domains in precision medicine and contributed its effectiveness with both supervised learning and distant learning.

Furthermore, Multi-task learning significantly improves extraction accuracy. This approach has proved that extraction beyond sentence boundary, served to produce better linguistic knowledge which provides a consistent gain of the model. This technique has been applied only to the medical field. In this thesis, some techniques from this paper are followed. Ernst et al. (2018) proposed an approach to harvest higher-arity facts from texts, considering ternary and higher-arity relations. It uses the fact-pattern duality principle to gather fact candidates with high recall, and for high precision, it devises a constraint-based reasoning method to eliminate a false candidate. The approach can harvest facts with higher-arity as well as high precision.

In contrast to the studies above, we reviewed binary relation extraction, n-ary relation extraction, and its further approaches. A thorough literature review was conducted in order to identify the common approaches, state-of-the-art methods and challenges pertaining to each component. This was followed by introducing the task of complex relation extraction and discussing the previous approaches, state-of-the-art methods and accompanied challenges in order to provide motivation for the work that follows in the next sections. We need to find relations and entities in the text corpus which can further be trained to obtain n-ary relation extraction. However, we took some ideas from above conducted research for n-ary relation extraction using supervised learning and multi-task learning.

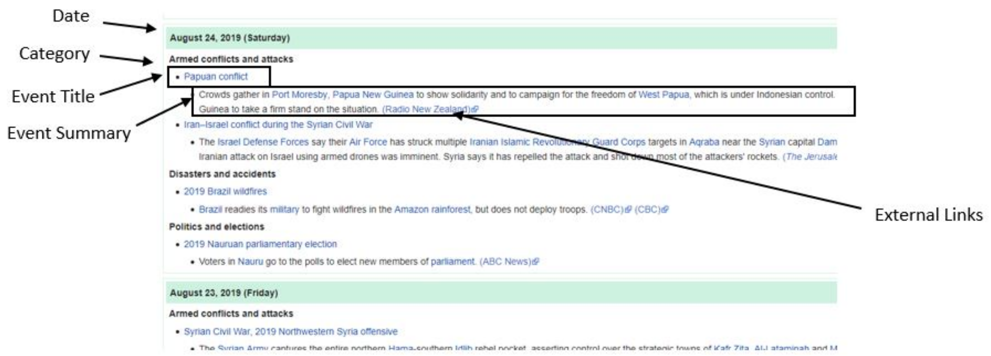
4 Dataset Collection

This section is divided into three sub-sections. Section 4.1 describes Wikipedia current event portal and Section 4.2 presents event extraction and preprocessing of extracted features. Finally, the dataset distributions and description of the features, as well as the descriptive statistics, are listed in Section 3.3.

4.1 Wikipedia current events portal

The dataset used in this project belongs to Wikipedia's Current Event Portal (WCEP). It provides a platform for creating and archiving daily events summaries of news from several external links. Users add news in the portal daily to the main page of the portal. This portal creates daily entities and adds them to the portal homepage. All the events are listed in the right container which is summed up at the end of the month in one Wikipedia page which has a unique identifier of the form [1].

WCEP is a platform for collaboratively creating and archiving daily event summaries for the crowd and by the crowd. It adds a small overview of daily events and relevant news. The main advantage of WCEP is the manual annotation of the event summaries as can be seen in Figure 11.



August 24, 2019 (Saturday)

Armed conflicts and attacks

- Papuan conflict
  - Crowds gather in Port Moresby, Papua New Guinea to show solidarity and to campaign for the freedom of West Papua, which is under Indonesian control. Guinea to take a firm stand on the situation. (Radio New Zealand)
- Iran–Israel conflict during the Syrian Civil War
  - The Israel Defense Forces say their Air Force has struck multiple Iranian Islamic Revolutionary Guard Corps targets in Aqraba near the Syrian capital Dam Iranian attack on Israel using armed drones was imminent. Syria says it has repelled the attack and shot down most of the attackers' rockets. (*The Jerusale*

Disasters and accidents

- 2019 Brazil wildfires
  - Brazil readies its military to fight wildfires in the Amazon rainforest, but does not deploy troops. (CNBC) (CBC)

Politics and elections

- 2019 Nauruan parliamentary election
  - Voters in Nauru go to the polls to elect new members of parliament. (ABC News)

August 23, 2019 (Friday)

Armed conflicts and attacks

- Syrian Civil War, 2019 Northwestern Syria offensive
  - The Syrian Army captures the entire northern Hama-southern Idlib rebel pocket, asserting control over the strategic towns of Kafr Zita, Al-Lataminah and M

Figure 11: A screenshot from Wikipedia's Current Events portal (Tran and Alrifai, 2014)

Each event is assigned to the specified category in the portal (e.g., Armed conflicts, Business and economy, International Relations etc.) and is linked to the event title when its applicable (e:g Papuan conflict), which is described in a separate Wikipedia article. These entities are linked to the Wikipedia articles. Additionally, it links to the external resources related to the events. Because of that, WCEP can be used in many applications nowadays as it is a combination of archived data provided by crowd and journalists all together in one page.

[1]Month_Year (e.g. http://en.wikipedia.org/wiki/March_2019)

4.2 Event Extraction and Pre-processing

The Wikipedia current events have been extracted using web scraper which sends a request to the URL and extracts all the raw data, which is then converted into a useful format by using beautiful soup. Once it finishes parsing, the scraper searches the data in the HTML tags , which requires to be extracted and subsequently converted into the specified format as shown in Figure 12.

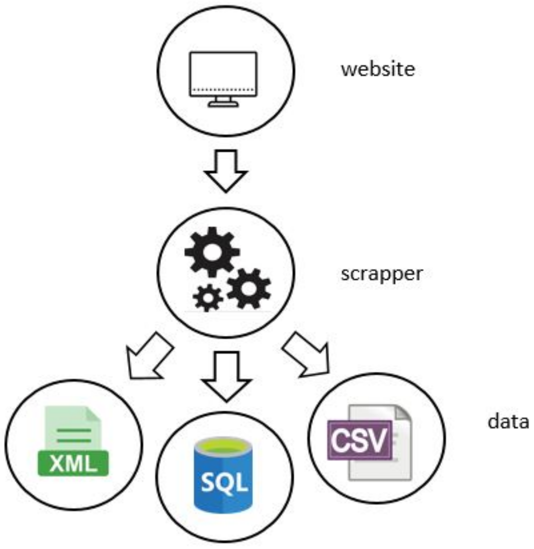


Figure 12: Web scraping

The following features have been extracted from the website; date, category, event title, event summary, entities, and external link. We used WCEP in this project to extract binary and n-ary relation extraction from the event summary to find interactions between several news events to build an easily searchable knowledge base. Given below are the assumptions and associated information regarding the data specific to the scope of the project:

- The data used in the project is from in between 2016-2018. It covers all the events that occurred over the past two years.
- In total, the web scraper extracted 12k news events found in WCEP, dating in between January 2016 and December 2018. The portal is still active and gaining more and more content every day, as shown in figure 13.

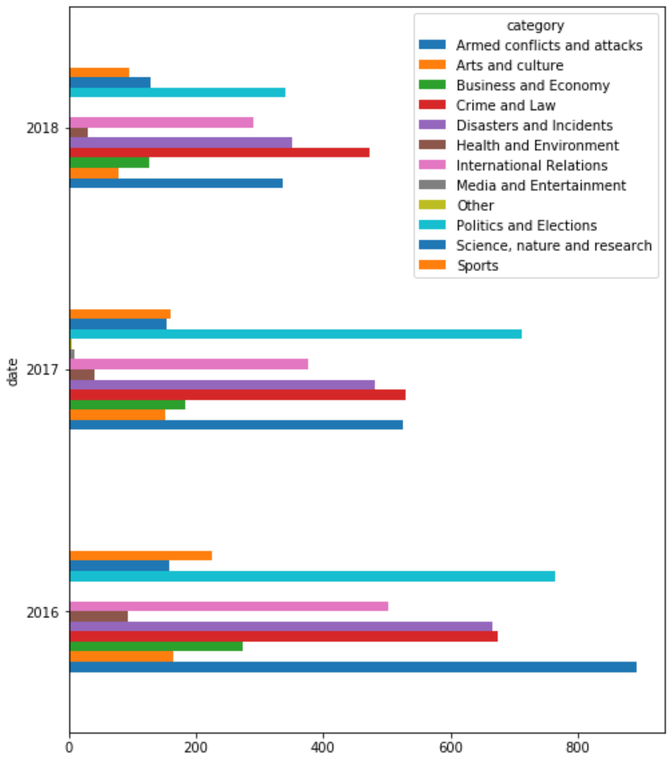


Figure 13: Number of events per year

- While extracting data from web scraper, we realized, the HTML format of the webpage had changed after September 2017, and that all the data was saved in divs instead of tables, thus requiring two different implementations of the web scrapers.
- The data consists of different attributes such as categorical attributes like event categories, date, and string attributes (such as event summary). Each category has several events with their descriptions in another attribute 'event summary'.
- Before starting an analysis of the distribution of each category on the dataset, the data preprocessing step is crucial to be applied on the categories of the events in order to maintain n-to-1 mapping for each category such as Armed conflicts, disasters, crimes, business, health, media, politics, and sport. For instance, any category in {'Armed Conflicts', 'armed conflicts and attack', 'attacks' , 'Attacks and conflicts', 'Attacks and armed conflicts' } should map to one collective category, such as, 'Armed conflicts and attacks'.

4.3 Event Type Distributions

The categories of the events are shown in Figure 14, which shows the wide coverage of topics. However, most of the news events belong to armed conflicts and attacks, crime, disasters, international relations and politics while fewer events have been reported in the sports, media and art. Those events which do not belong to any category are listed as 'other'.

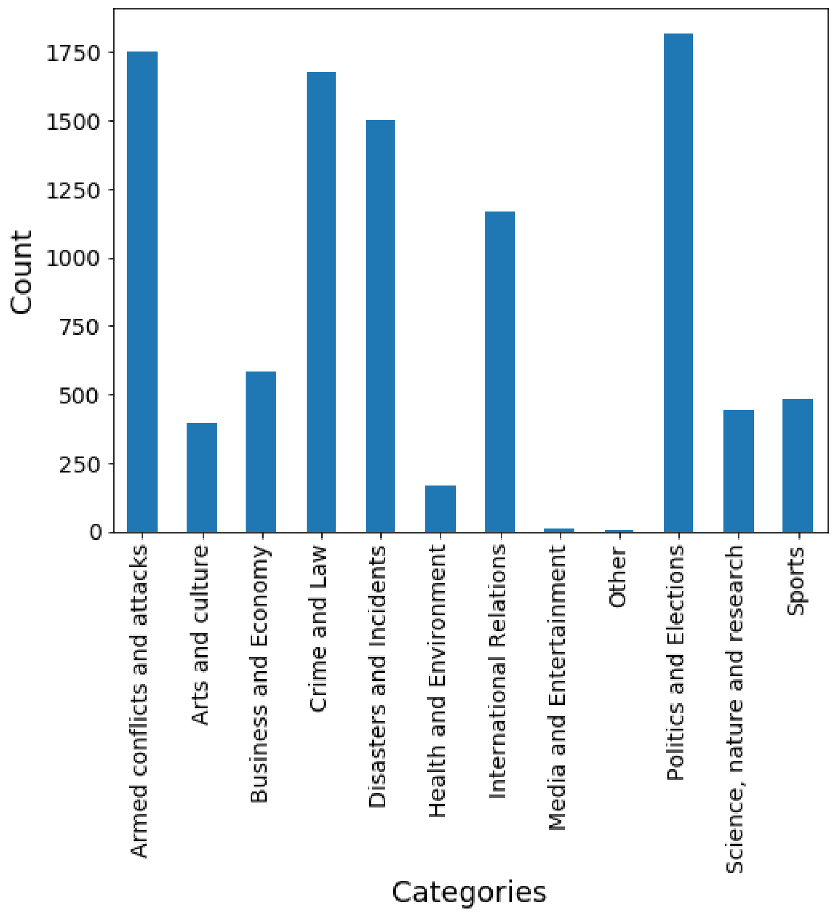


Figure 14: Distribution of events over major categories

Table 1 shows all the attributes of the news portal which are extracted by webscraper from WCEP. In this thesis, we have used event summaries for the task of n-ary relation extraction.

Concept	Description
Date	the date at which the event occurred
Category	the high level (topic) category the event belongs to (e.g. Armed conflicts, Internation Relations, Business etc.)
Event Summary	a brief summary of the news event
Event Title	event title that spans a period longer than one day, this is typically a story that is described in Wikipedia in a separate article
Entities	a list of entities mentioned in the description (e.g. person, location, or simply anything that is described by a Wikipedia article)
External Link	links to external online news articles

Table 1: Event information indexed by WikiTimes (Tran and Alrifai, 2014)

5 Annotation Interface

This section is divided into three subsections; Section 5.1 incorporates all the implementation process behind the task of the annotation interface. Section 5.2 describes the functionality of the annotation interface and how it works and the challenges and drawbacks during the implementation of the interface mentioned in Section 5.3.

5.1 Annotation Architecture

Annotation interface is a manual annotation tool with the purpose of identifying important arguments and its associated relations. Figure 15 demonstrates the composition of the annotation interface, having a sentence which is composed of spans and tokens. Spans are the noun phrases which can be selected as important or unimportant and tokens are the words which are outside of spans, converted into clickable buttons; in this way, the user can identify relation phrase for the selected arguments. However, this interface is generating two outputs, important arguments and relation to those arguments.

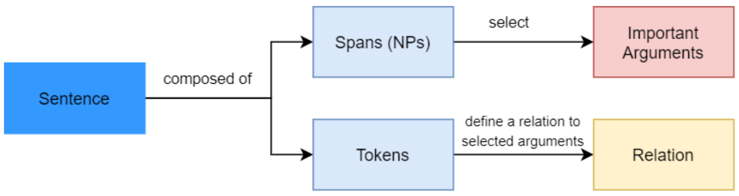
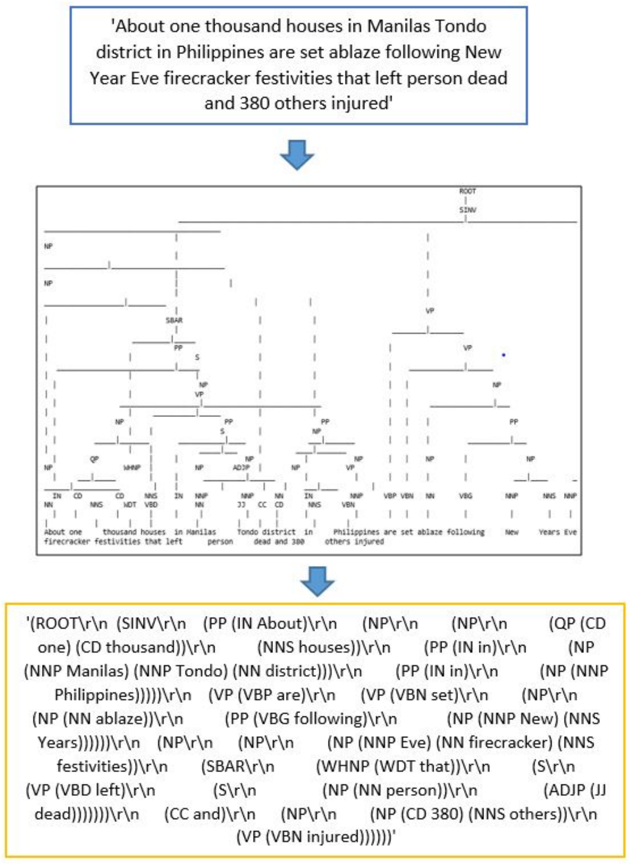


Figure 15: Composition of annotation architecture

Moreover, the constituency parse trees have been used for the extraction of Noun Phrases (NPs), which are transmitted into the interface as spans for the selection of important or unimportant arguments within a sentence, as shown in Figure 16.

In this thesis, CoreNLP has been used for the process of semantic and synthetic natural language tasks such as tokenization and sentence splitting, part of speech and constituency parser, which is further discussed in Section 2.2.

Constituency parser has been used for the extraction of noun phrases from the sentences which provide full syntactic analysis. First, it identifies nouns, verbs, and adjectives which are further categorized as higher-order units such as noun phrases, verb phrases and preposition phrases. The constituent-based NPs, VPs, and PPs are then saved in tree annotation. The annotation interface considers only those noun phrases which do not contain other noun phrases; otherwise, it causes overlapping in an interface, as shown in Figure 16.



'About one thousand houses in Manilas Tondo district in Philippines are set ablaze following New Year Eve firecracker festivities that left person dead and 380 others injured'

```
'(ROOT\r\n  (SINV\r\n    (PP (IN About)\r\n      (NP\r\n        (NP\r\n          (QP (CD one) (CD thousand))\r\n          (NNS houses))\r\n        (PP (IN in)\r\n          (NP (NNP Manilas) (NNP Tondo) (NN district)))\r\n        (PP (IN in)\r\n          (NP (NNP Philippines)))))\r\n    (VP (VBP are)\r\n      (VP (VBN set)\r\n        (NP\r\n          (NP (NN ablaze))\r\n          (PP (VBG following)\r\n            (NP (NNP New) (NNS Years))))))\r\n    (NP\r\n      (NP\r\n        (NP (NNP Eve) (NN firecracker) (NNS festivities))\r\n        (SBAR\r\n          (WHNP (WDT that))\r\n          (S\r\n            (VP (VBD left)\r\n              (S\r\n                (NP (NN person))\r\n                (ADJP (JJ dead)))))))\r\n      (CC and)\r\n      (NP\r\n        (NP (CD 380) (NNS others))\r\n        (VP (VBN injured)))))'
```

Figure 16: Constituency parsing

Table 2 shows the selection of NPs from the parse tree, which does not contain other NPs. If we take into account all NPs of the parse tree, then it can create overlapping in annotation interface. Therefore, we are considering only those NPs which do not constitute more than one NP. This is not an issue because in the interface user can select these NPs which can constitute together later while training of the model as in Table 2 each long argument consists of more than one NPs.

Considering all NPs	**Considering only NPs that not contain other NPs.**
['A', 'new', '28foot', 'tall', 'statue', 'of', 'Jesus']	
['A', 'new', '28foot', 'tall', 'statue']	['A', 'new', '28foot', 'tall', 'statue']
['Jesus']	['Jesus']
['Jesus', 'de', 'Greatest', 'is', 'unveiled', 'on', 'New', 'Years', 'Day', 'outside', 'St.', 'Aloysius', 'Catholic', 'Church', 'in','Abajah', 'village', 'Nigerias', 'Imo', 'state', 'which', 'is', 'described','as', 'tallest', 'Jesus', 'statue', 'in', 'Africa']	
['Jesus', 'de', 'Greatest']	['Jesus', 'de', 'Greatest']
['New', 'Years']	['New', 'Years']
['St.', 'Aloysius', 'Catholic', 'Church', 'in', 'Abajah', 'village', 'Nigerias', 'Imo', 'state', 'which', 'is', 'described', 'as','tallest', 'Jesus', 'statue', 'in', 'Africa']	
['St.', 'Aloysius', 'Catholic', 'Church', 'in', 'Abajah', 'village', 'Nigerias', 'Imo', 'state']	
['St.', 'Aloysius', 'Catholic', 'Church']	['St.', 'Aloysius', 'Catholic', 'Church']
['Abajah', 'village', 'Nigerias', 'Imo']	['Abajah', 'village', 'Nigerias', 'Imo']
['tallest', 'Jesus', 'statue', 'in', 'Africa']	
['tallest', 'Jesus', 'statue']	['tallest', 'Jesus', 'statue']
['Africa']	['Africa']

Table 2: Selection of Noun Phrases (NPs)

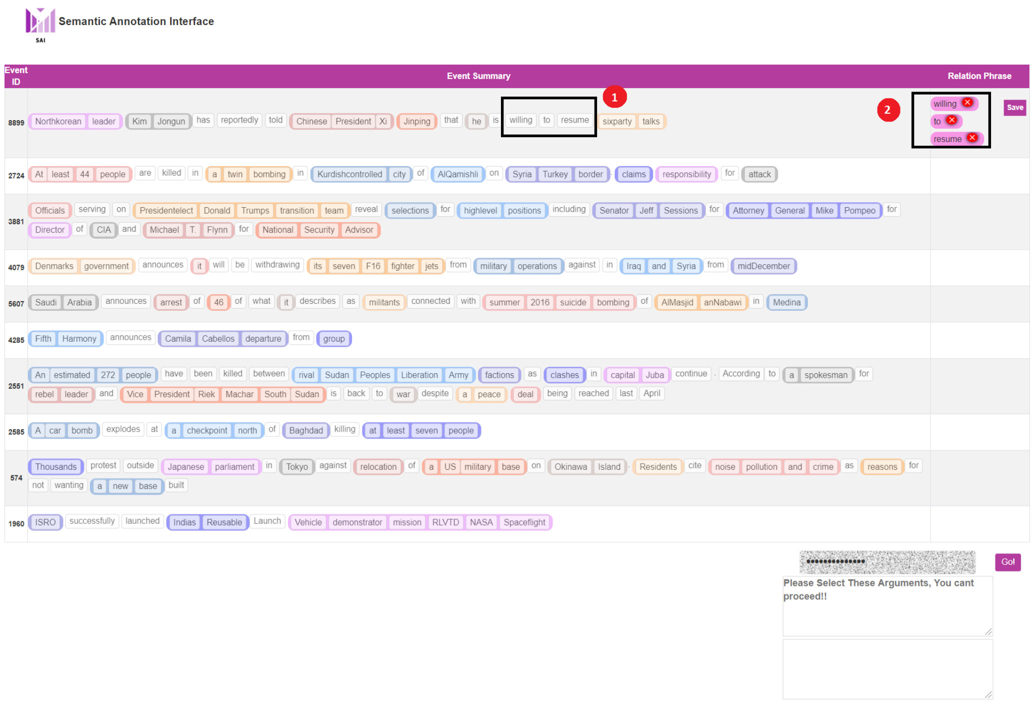
5.2 Annotation Protocol

A semantic annotation interface is a tool for selecting important arguments and the relation phrases connected to such selected arguments in a sentence, which then can be provided to crowd-workers for identification of important arguments and the relation between them. This interface has increased the validity and quality of the relation extraction tasks by being more enhanced in deep learning algorithms for identifying more and more arguments and respective relations within a dataset. Following are the steps to follow to do annotation tasks for the annotation interface.

Step 1: Select a relation phrase

Following are the steps for the selection of relation phrase.

1. The first step is to select the relation phrase in order to activate spans, which are arguments. User cannot select an argument until the relation phrase is selected.
2. When a user clicks on any word outside of span, then the word pops up in the relation phrase column.
3. After the creation of a relation phrase, a user must click on the Save button in order to save the relation phrase to the database, which also activates the functionality of spans as shown in Figure 17.



Event ID	Event Summary	Relation Phrase
8899	Northkorean leader Kim Jongun has reportedly told Chinese President Xi Jinping that he is willing to resume sixparty talks	willing to resume
2724	At least 44 people are killed in a twin bombing in Kurdishcontrolled city of AlQamishli on Syria Turkey border . claims responsibility for attack	
3881	Officials serving on Presidentelect Donald Trumps transition team reveal selections for highlevel positions including Senator Jeff Sessions for Attorney General Mike Pompeo for Director of CIA and Michael T. Flynn for National Security Advisor	
4079	Denmarks government announces it will be withdrawing its seven F16 fighter jets from military operations against in Iraq and Syria from midDecember	
5607	Saudi Arabia announces arrest of 46 of what it describes as militants connected with summer 2016 suicide bombing of AlMasjid anNabawi in Medina	
4285	Fifth Harmony announces Camila Cabellos departure from group	
2551	An estimated 272 people have been killed between rival Sudan Peoples Liberation Army factions as clashes in capital Juba continue . According to a spokesman for rebel leader and Vice President Riek Machar South Sudan is back to war despite a peace deal being reached last April	
2585	A car bomb explodes at a checkpoint north of Baghdad killing at least seven people	
574	Thousands protest outside Japanese parliament in Tokyo against relocation of a US military base on Okinawa Island . Residents cite noise pollution and crime as reasons for not wanting a new base built	
1960	ISRO successfully launched Indias Reusable Launch Vehicle demonstrator mission RLVTD NASA Spaceflight	

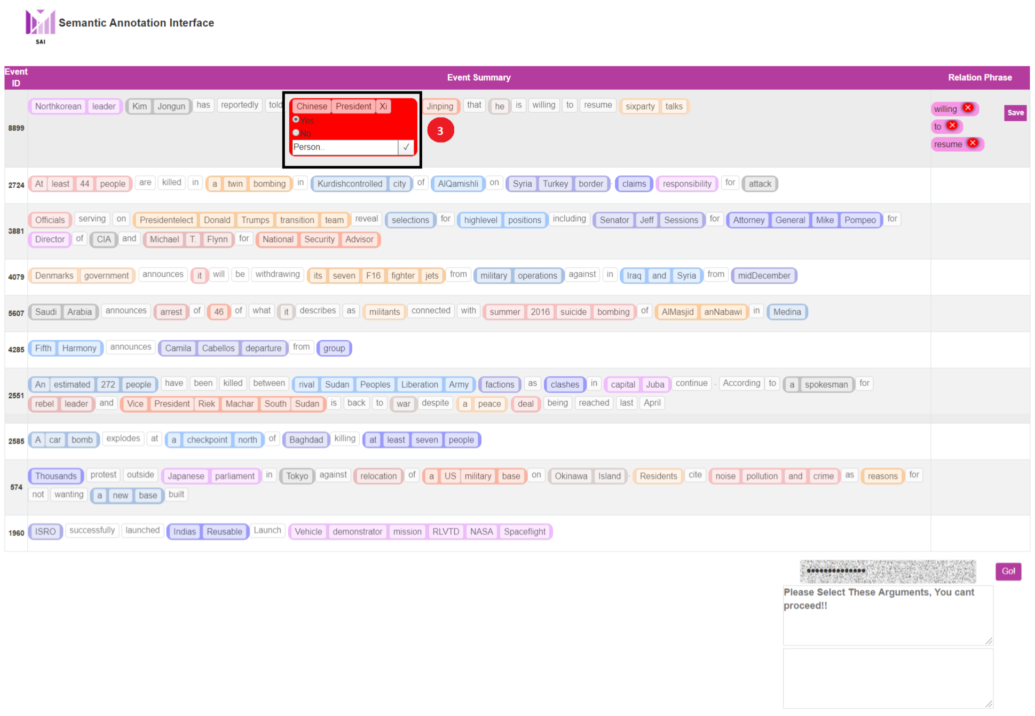
Please Select These Arguments, You cant proceed!!

Figure 17: Select a relation phrase (step 1)

Step 2: Select arguments related to the selected relation phrase

After selecting a relation phrase, the next task is to identify important arguments in correspondence with the previously identified relation phrase. Following are the steps for this given task.

1. User has to hover the cursor over the span to select one of the two options, important or not important as shown in Figure 18.
2. If the user clicks on the important, then a text-box appears within the hovered span where the user can add any related text to the corresponding argument if needed. It would get automatically saved.


Semantic Annotation Interface
SAI

Event ID	Event Summary	Relation Phrase
8899	Northkorean leader Kim Jongun has reportedly told Chinese President Xi Jinping that he is willing to resume sixparty talks	willing to resume
2724	At least 44 people are killed in a twin bombing in Kurdishcontrolled city of AlQamishli on Syria Turkey border . claims responsibility for attack	
3881	Officials serving on Presidentelect Donald Trumps transition team reveal selections for highlevel positions including Senator Jeff Sessions for Attorney General Mike Pompeo for Director of CIA and Michael T. Flynn for National Security Advisor	
4079	Denmarks government announces it will be withdrawing its seven F16 fighter jets from military operations against in Iraq and Syria from midDecember	
5607	Saudi Arabia announces arrest of 46 of what it describes as militants connected with summer 2016 suicide bombing of AlMasjid anNabawi in Medina	
4285	Fifth Harmony announces Camila Cabellos departure from group	
2551	An estimated 272 people have been killed between rival Sudan Peoples Liberation Army factions as clashes in capital Juba continue . According to a spokesman for rebel leader and Vice President Riek Machar South Sudan is back to war despite a peace deal being reached last April	
2585	A car bomb explodes at a checkpoint north of Baghdad killing at least seven people	
574	Thousands protest outside Japanese parliament in Tokyo against relocation of a US military base on Okinawa Island . Residents cite noise pollution and crime as reasons for not wanting a new base built	
1960	ISRO successfully launched Indias Reusable Launch Vehicle demonstrator mission RLVTD NASA Spaceflight	

Chinese President Xi
(● Yes) ○ No
Person.. ✓

Please Select These Arguments, You cant proceed!!

Figure 18: Select arguments for selected relation phrase (step 2)

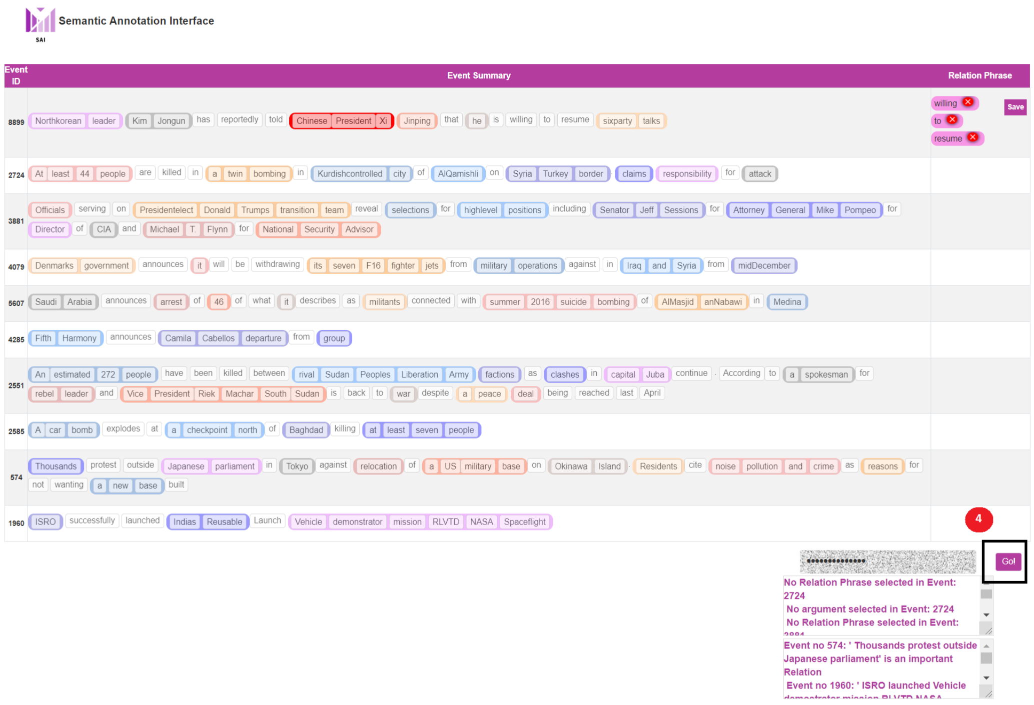
Step 3: Validation of the results

Repeat the above two steps for each event in the interface. After completion of the above tasks, the next step is to check the validity of the user's work.

1. After the selection of arguments and relation phrase, Go! button is to be selected, as shown in Figure 19.
2. There are some conditions and validity checks which mark the functionality of a button, so as to validate the results of the user. First, the user must select a "Relation Phrase" in each event, and second, at least one argument must be selected as an important argument

in the interface. Lastly, the user must also select expert annotations (which are annotated by experts), in order to ensure the validity and credibility of the user's work.

3. The semantic interface generates a validation code when a user uploads it which in turn is used for the payment of tasks. The code will not appear until the crowd worker accomplishes all these above-mentioned conditions and terms. The text-box below the code generator mentions all the validation requirements to be fulfilled for acquiring a validation code, as shown in Figure 19.


Semantic Annotation Interface

SAI

Event ID	Event Summary	Relation Phrase
8899	Northkorean leader Kim Jongun has reportedly told Chinese President Xi Jinping that he is willing to resume sixparty talks	willing ✕ to ✕ resume ✕ Save
2724	At least 44 people are killed in a twin bombing in Kurdishcontrolled city of AlQamishli on Syria Turkey border . claims responsibility for attack	
3881	Officials serving on Presidentelect Donald Trumps transition team reveal selections for highlevel positions including Senator Jeff Sessions for Attorney General Mike Pompeo for Director of CIA and Michael T. Flynn for National Security Advisor	
4079	Denmarks government announces it will be withdrawing its seven F16 fighter jets from military operations against in Iraq and Syria from midDecember	
5607	Saudi Arabia announces arrest of 46 of what it describes as militants connected with summer 2016 suicide bombing of AlMasjid anNabawi in Medina	
4285	Fifth Harmony announces Camila Cabellos departure from group	
2551	An estimated 272 people have been killed between rival Sudan Peoples Liberation Army factions as clashes in capital Juba continue . According to a spokesman for rebel leader and Vice President Riek Machar South Sudan is back to war despite a peace deal being reached last April	
2585	A car bomb explodes at a checkpoint north of Baghdad killing at least seven people	
574	Thousands protest outside Japanese parliament in Tokyo against relocation of a US military base on Okinawa Island . Residents cite noise pollution and crime as reasons for not wanting a new base built	
1960	ISRO successfully launched Indias Reusable Launch Vehicle demonstrator mission RLVTD NASA Spaceflight	4

Go!

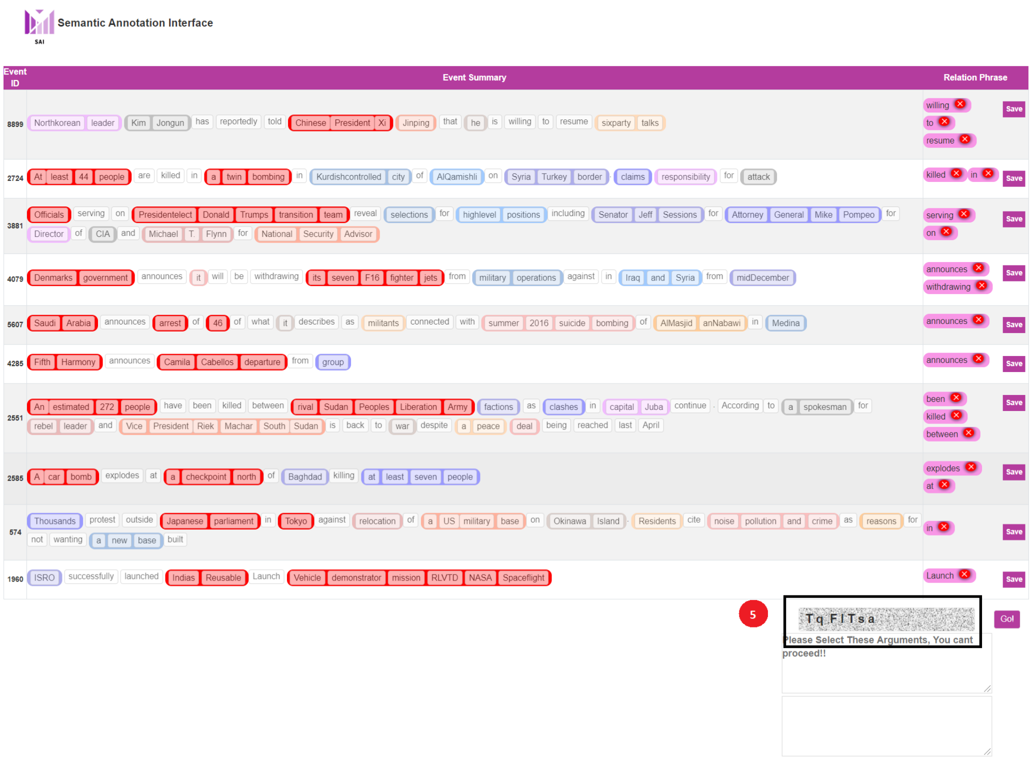
No Relation Phrase selected in Event: 2724
No argument selected in Event: 2724
No Relation Phrase selected in Event:

Event no 574: ' Thousands protest outside Japanese parliament' is an important Relation
Event no 1960: ' ISRO launched Vehicle

Figure 19: Validation of the results (step 3)

Step 4: Submission of the results

Last but not the least, after following all the validation checks and conditions, the user has to click the 'Go!' button, which again checks the validation of users' input. Once the validation is passed, then the code appears on the screen, as shown in Figure 20.


Semantic Annotation Interface

Event ID	Event Summary	Relation Phrase	
8899	Northkorean leader Kim Jongun has reportedly told Chinese President Xi Jinping that he is willing to resume sixparty talks	willing ✕ to ✕ resume ✕	Save
2724	At least 44 people are killed in a twin bombing in Kurdishcontrolled city of AlQamishli on Syria Turkey border · claims responsibility for attack	killed ✕ in ✕	Save
3881	Officials serving on Presidentelect Donald Trumps transition team reveal selections for highlevel positions including Senator Jeff Sessions for Attorney General Mike Pompeo for Director of CIA and Michael T. Flynn for National Security Advisor	serving ✕ on ✕	Save
4079	Denmarks government announces it will be withdrawing its seven F16 fighter jets from military operations against in Iraq and Syria from midDecember	announces ✕ withdrawing ✕	Save
5607	Saudi Arabia announces arrest of 46 of what it describes as militants connected with summer 2016 suicide bombing of AlMasjid anNabawi in Medina	announces ✕	Save
4285	Fifth Harmony announces Camila Cabellos departure from group	announces ✕	Save
2551	An estimated 272 people have been killed between rival Sudan Peoples Liberation Army factions as clashes in capital Juba continue . According to a spokesman for rebel leader and Vice President Riek Machar South Sudan is back to war despite a peace deal being reached last April	been ✕ killed ✕ between ✕	Save
2585	A car bomb explodes at a checkpoint north of Baghdad killing at least seven people	explodes ✕ at ✕	Save
574	Thousands protest outside Japanese parliament in Tokyo against relocation of a US military base on Okinawa Island · Residents cite noise pollution and crime as reasons for not wanting a new base built	in ✕	Save
1960	ISRO successfully launched Indias Reusable Launch Vehicle demonstrator mission RLVTD NASA Spaceflight	Launch ✕	Save

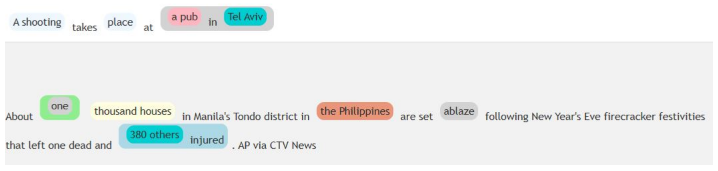
Please Select These Arguments, You cant proceed!!

Figure 20: Submission of the results (step 4)

5.3 Drawbacks and Challenges

During the preparation of the interface, several challenges surfaced. First, the noun phrases overlapped in the interface, as they were extracted from a given sentence; for instance, some noun phrases consisted of more than one noun phrases, which hampered the implementation of the interface. Only those NPs were considered which didn't include other NPs, which are closer to the POS tags.

Secondly, another drawback of these predefined units was that it missed upon long arguments that comprised of more than one NPs in a NP, which made the formation of a relation between those arguments impossible. Hence, it would have made it difficult for the crowd workers to select a relation phrase and other essential arguments in a sentence, where noun phrases overlapped as shown in Figure 21. Hence it led to the development of only considering those NPs that didn't consist of other NPs.


A shooting takes place at a pub in Tel Aviv

About one thousand houses in Manila's Tondo district in the Philippines are set ablaze following New Year's Eve firecracker festivities that left one dead and 380 others injured . AP via CTV News

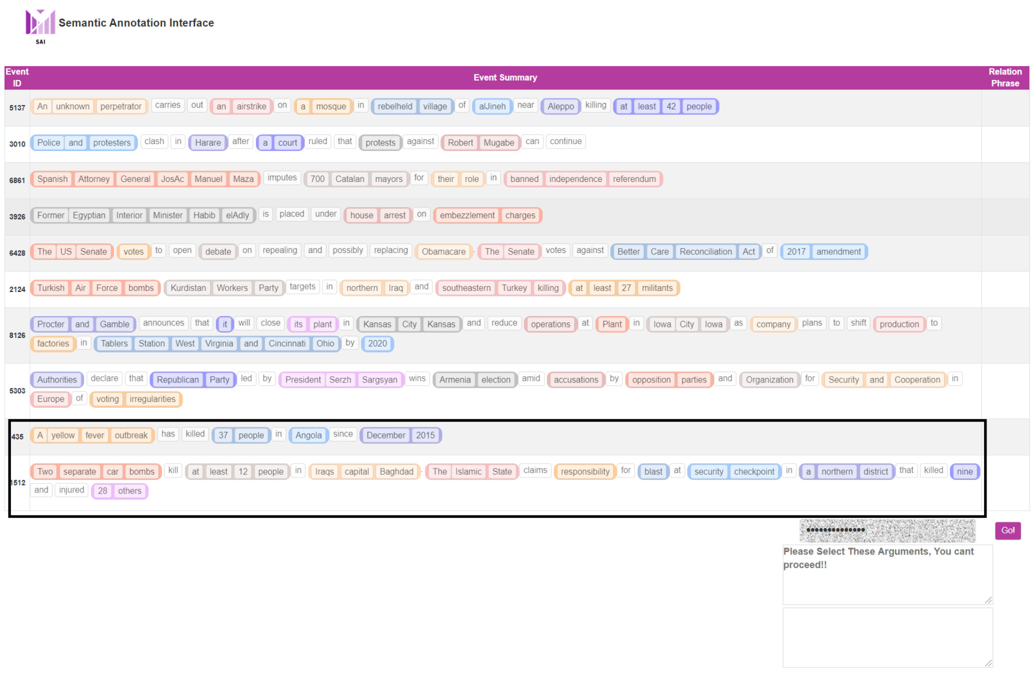
Figure 21: Overlapping of noun-phrases in interface

6 Ground-truth Construction

For the annotation process, its important to construct the gold labels in order to measure the quality of results from crowdsourced annotations. We have annotated around 2000 sentences, which checks the validity of the crow-sourced work.

6.1 Expert Annotations

Expert annotations is a work that is done by experts (us), which inspects the standard to check the quality of the crowdsourced result. It is crucial to check whether annotated results from crowdsourcing are accurate or not; otherwise model can end-up having bad results. Every time the 10 events appear on the interface as mentioned in Section 5, which holds 8 normal events and 2 events covertly annotated by expert which serve to check whether the annotator did the right annotation or not as shown in Figure 22.



Event ID	Event Summary	Relation Phrase
5137	An unknown perpetrator carries out an airstrike on a mosque in rebelheld village of alJineh near Aleppo killing at least 42 people	
3010	Police and protesters clash in Harare after a court ruled that protests against Robert Mugabe can continue	
6861	Spanish Attorney General JosAc Manuel Maza imputes 700 Catalan mayors for their role in banned independence referendum	
3926	Former Egyptian Interior Minister Habib elAdly is placed under house arrest on embezzlement charges	
6428	The US Senate votes to open debate on repealing and possibly replacing Obamacare . The Senate votes against Better Care Reconciliation Act of 2017 amendment	
2124	Turkish Air Force bombs Kurdistan Workers Party targets in northern Iraq and southeastern Turkey killing at least 27 militants	
8126	Procter and Gamble announces that it will close its plant in Kansas City Kansas and reduce operations at Plant in Iowa City Iowa as company plans to shift production to factories in Tablers Station West Virginia and Cincinnati Ohio by 2020	
5303	Authorities declare that Republican Party led by President Serzh Sargsyan wins Armenia election amid accusations by opposition parties and Organization for Security and Cooperation in Europe of voting irregularities	
435	A yellow fever outbreak has killed 37 people in Angola since December 2015	
1512	Two separate car bombs kill at least 12 people in Iraqs capital Baghdad . The Islamic State claims responsibility for blast at security checkpoint in a northern district that killed nine and injured 28 others	

Please Select These Arguments, You cant proceed!!

Figure 22: Expert annotations are hidden in these two events in order to check the validity of crowdsourced's work

Table 3 demonstrates that the total no of annotations done by experts has been held in 2000 sentences, which contains 4094 arguments and 2038 relations. These arguments and relations have been used as ground-truth for crowdsourced annotations which are hidden at the back-end of the interface.

Corpora	Sentences	Arguments	Relations
Expert annotations	2000	4094	2038

Table 3: Expert annotations

6.2 Crowdsourced Ground-truth Collection

Crowdsourced annotations are done as a task by crowd-workers. There are about 10,000 sentences which have been used in the annotation interface, in order to get annotated arguments and relations from crowd workers. There are some conditions which crowd workers must follow while doing the annotation process:

1. Each event must be annotated by 3 annotators.
2. Annotators must have followed the validation checks, as mentioned in Section 5. For instance, expert annotation is hidden in the last two events, as shown in Figure 22. Annotator must follow these expert annotations as it shows the credibility of crowd workers' work.

7 n-ary Relation Extraction Models

This section is divided into three sections; Section 7.1 describes the objectives of Multi-Task Learning (MTL) and Section 7.2 demonstrates the model setup for the MTL. Finally, model training and ablation has been discussed in Section 7.3.

7.1 Why Multi-Task Learning?

Multi-task learning idea revolves around the notion of learning multiple tasks simultaneously such as POS tagging, semantic tagging. In this thesis, the main focus is to learn argument tags and relation tags from the training data. Multi-task learning aims to improve the performance of multiple related tasks.

1. MTL helps the model to generalize better when the data is noisy or limited. It introduces a useful inductive bias helping a model ignore task-specific noise which gives a better generalization of the data.

2. MTL can help the model learn relevant features from the training data through additional evidence.

3. MTL smoothes the loss function and minimizes the complexity of the model.

4. In the cases when there is no overlap between two tasks, MTL results in the (implicit) augmentation of the training data allowing the model to generalize better by averaging the task-specific noise patterns.

In this thesis, multi-task learning has been used for the tagging of arguments and relations simultaneously. The multi-task learning helps to find n-ary relation among different sentences or within a sentence.

7.2 Model Setup for Multi-Task Learning

The multi-task learning is one major contribution of this thesis which has been used for various sequence tagging tasks. Multi-task learning is the machine learning task which learns multiple tasks at the same time. It optimizes the loss functions of related tasks, which can achieve superior results compared to the models which are trained on a single task. The goal of multi-task learning is summarized by (Caruana, 1997) “MTL improves generalization by leveraging the domain-specific information contained in the training signals of related tasks”.

In this dissertation, the main intuition behind using MTL is to identify arguments and relation in a text simultaneously during the training of data. For instance, Figure 23 illustrates the model structure which has trained for the task of argument tagging and relation tagging. The sentences are converted to a sequence of word vectors using GloVe embeddings which are explained in Section 2.3; these embeddings have been passed through LSTM layers along with shared layers to train data on two labels simultaneously; argument tags and relation tags. Each task has its final output layer. Therefore, the model's parameters are all shared except for the weights from the hidden layer to the two final output layers. This way model was able to learn more complex tasks from simpler ones.

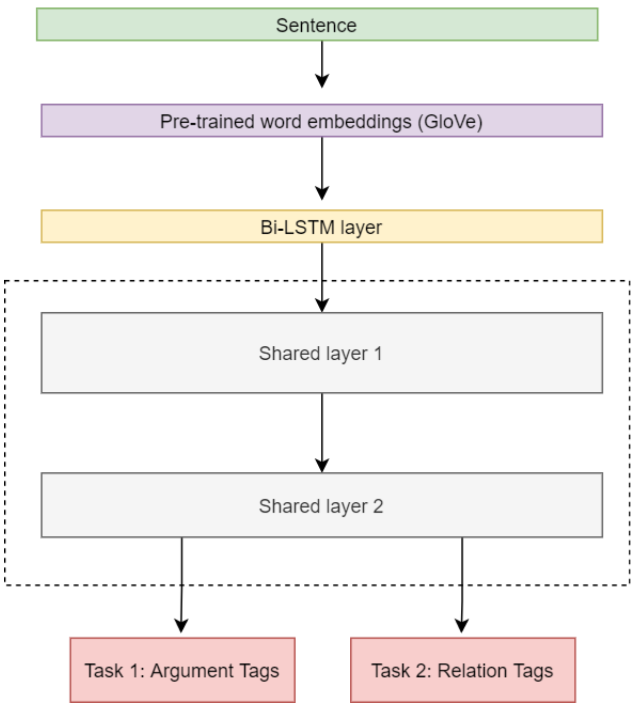


Figure 23: Model structure for Multi-Task Learning (MTL)

7.2.1 Utilizing Pre-Trained Word Embeddings

For this thesis, we have used GloVe word embeddings for the training of the model, which was explained in Section 2.3. Glove is an unsupervised learning algorithm which obtains vector representations for words. Utilizing these pre-trained word embeddings is a simple and effective method to improve the performance of machine learning systems in NLP (Turian et al., 2010b). GloVe pre-trained embeddings reduce training time and improved overall performance of the model. In this way, it can learn more common words or vector in a dataset.

7.2.2 Recurrent Neural Networks (RNN)

RNNs are used to solve sequence tagging tasks and their suitability for these tasks has been justified in Section 2.1.2. The system, therefore, uses bidirectional RNN layers (bi-LSTM) for the training of the data, as shown in Figure 23. Bi-LSTM can connect two hidden layers of opposite directions to the same output.

7.2.3 Parameter Sharing

There are two types of multi-task learning methods for deep learning, hard-parameter and soft parameter learning of hidden layers.

Hard parameter sharing

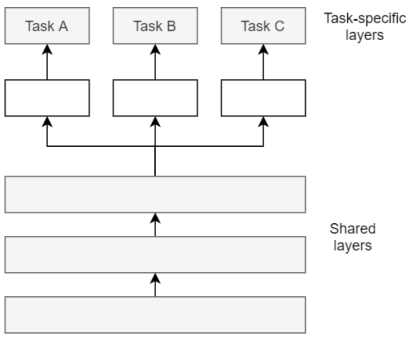
Hard parameter sharing is generally applied by sharing the hidden layers between all tasks while keeping several task-specific output layers, as shown in fig 24a. This is the most commonly used approach to MTL in neural networks. (Caruana, 1997)

Hard parameter sharing for multi-task learning helps to reduce the risk of overfitting. In fact,(Baxter, 1997) showed that the risk of overfitting the shared parameters is an order n – where n is the number of tasks – smaller than overfitting the task-specific parameters, i.e. the output layers. The model gets efficient by adding more tasks to it, which reduces the chance of overfitting.

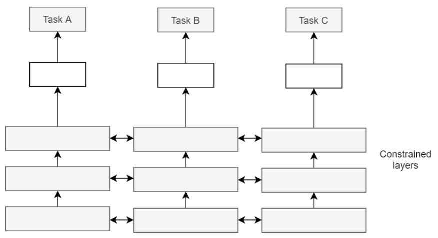
Soft parameter sharing

On the other hand, soft parameter sharing is composed of several tasks, but each task has its own model with its own parameters, as shown in Figure 24b. In order to make the parameters similar, the distance among the parameters of the model is then regularized, for instance, l2 norm for regularization (Duong et al., 2015). This method has been greatly inspired by regularization techniques for MTL.

This research has focused on hard parameter learning, as the model has shared hidden layers between all tasks while keeping several task-specific output layers such as argument tagging and relation tagging.



(a) Hard parameter sharing for multi-task learning



(b) Soft parameter sharing for multi-task learning

Figure 24: Parameter Sharing

7.2.4 Dropout

A dropout is an approach to regularization in neural networks which helps to reduce interdependent learning amongst the neurons. It dropouts a node by multiplying its output by zero. Only input and hidden nodes are dropped out from the model, as shown in Figure 25. In machine learning, regularization is a technique to prevent over-fitting . In order to that, it adds a penalty the loss function, that trained the model, in such a way that it does not learn the interdependent sets of features weights, for instance, L1 (Laplacian) and L2 (Gaussian) penalties.

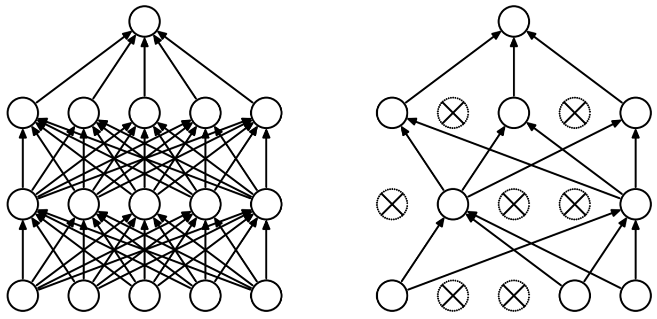
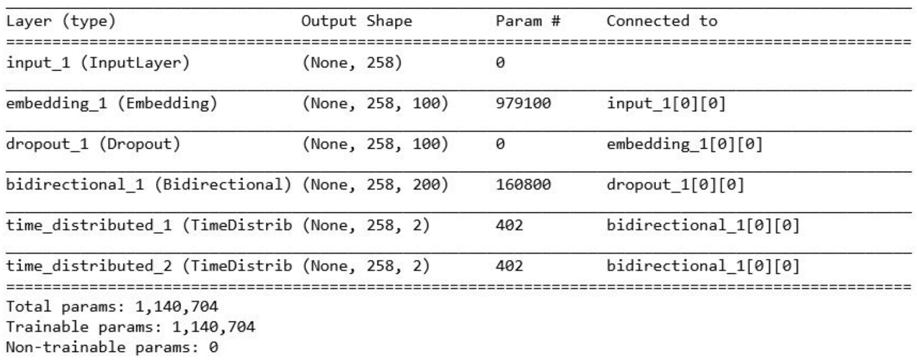


Figure 25: Dropout Neural Net Model. Left: A standard neural net with 2 hidden layers. Right: An example of a thinned out net produced by applying dropout to the network on the left. Crossed units have been dropped (Srivastava et al., 2014)

7.3 Model Training

Throughout the model training, hard parameter sharing has been used for multi-task learning. The model is trained jointly on both the arguments and relation tagging tasks by starting with one task, for example, the argument tagging task, which train the model using batches from the dataset, to minimize loss, then switching to the next sharing layer to perform relation tagging task and using the same dataset to minimize loss. In this way the model alternates between the tasks and thus, each batch leans a model that can solve both tasks.

Figure 26 presents a summary of the model. In multi-task learning, the two models are trained simultaneously, which gives 1,140, 704 parameters in total. Input is transmitted into embeddings, which results in a total of 979100 parameters, additionally bidirectional constitutes 258 input weights and have 200 units, hence it sums up to 160800 parameters. Both shared layers have 258 input weights and one weight of connection with bias, which gives 402 parameters. In a nutshell, there are 1,140, 704 trainable parameters. The role of this additional bias term is significant. It plays a role of increasing the model's capacity to a large extent.



```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_1 (InputLayer)             (None, 258)           0
____________________________________________________________________________________________________
embedding_1 (Embedding)          (None, 258, 100)      979100      input_1[0][0]
____________________________________________________________________________________________________
dropout_1 (Dropout)              (None, 258, 100)      0           embedding_1[0][0]
____________________________________________________________________________________________________
bidirectional_1 (Bidirectional)  (None, 258, 200)      160800      dropout_1[0][0]
____________________________________________________________________________________________________
time_distributed_1 (TimeDistrib  (None, 258, 2)        402         bidirectional_1[0][0]
____________________________________________________________________________________________________
time_distributed_2 (TimeDistrib  (None, 258, 2)        402         bidirectional_1[0][0]
====================================================================================================
Total params: 1,140,704
Trainable params: 1,140,704
Non-trainable params: 0
```

Figure 26: Model Summary

Figure 27 illustrates the model setup for multi-task learning in order to extract n-ary relations.

First, the word embedding layer represents each word by concatenating its word embedding (pre-trained vectors learned on text using word2vec). The pre-trained embeddings have remarkable capabilities for leveraging distributed similarities of words. This enables them to learn more extraction patterns and consequently extract more relationship instances.

The second layer adds the dropout to the embedding layer, which has passed through the bi-directional LSTM layer. Then the output of the previous layer passes to the next shared hidden layers to tag relations and arguments simultaneously, which gives two output layers, arguments tags and relation tags. The model was trained with Adam optimizer and trained for 5 epochs because more training will cause the model to overfit on the training data.

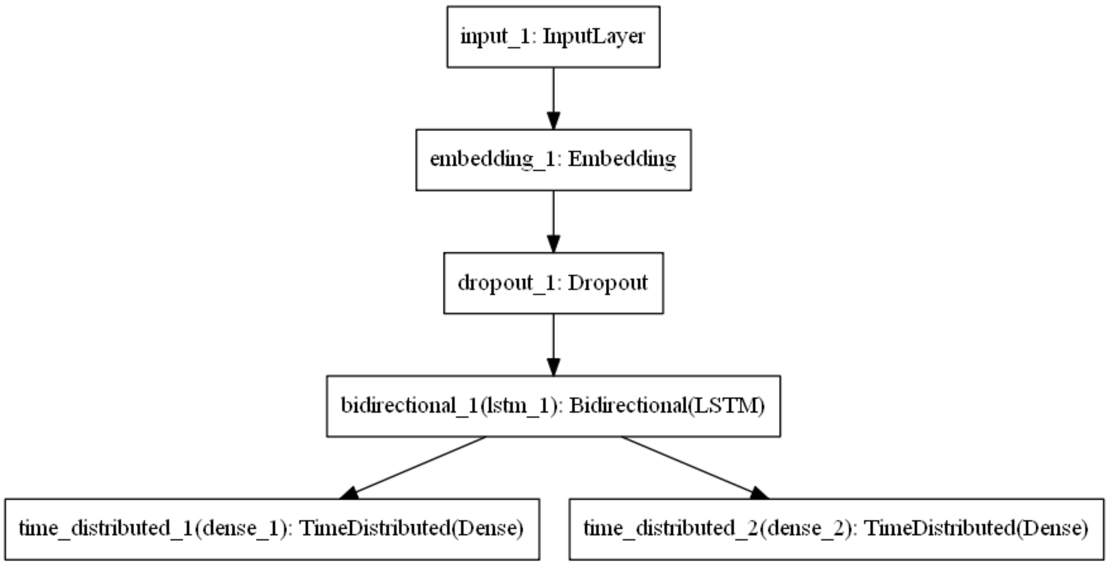

input_1: InputLayer
embedding_1: Embedding
dropout_1: Dropout
bidirectional_1(lstm_1): Bidirectional(LSTM)
time_distributed_1(dense_1): TimeDistributed(Dense)
time_distributed_2(dense_2): TimeDistributed(Dense)

Figure 27: Model setup for Multi-Task Learning (MTL)

In this thesis, the model was also tested by including character embeddings. The model can be seen as not dependent on the word embedding, as shown in Figure 28. By using GloVe embedding, words are represented as vectors. GloVe embedding is a vast dictionary which contains millions of words. However, those words might not be considered in the training which is not present in the GloVe embeddings. These embeddings deal with such words by assigning them some random values, which could end up confusing the model.

There is another mechanism that can handle out-of-vocabulary words, called character embeddings, which use the one-dimensional convolutional neural network (1D-CNN) to find the representation of words by looking at their character level compositions. The output of the character level embeddings step is similar to the production of word-level embedding step. This embedding has obtained 2 matrices, one for the context and other for the query with an equal length. This provides two sets of word representations for words, one from GloVe and the other from the 1D-CNN character level embeddings. In the next step, these two embeddings have been concatenated, as shown in Figure 28.

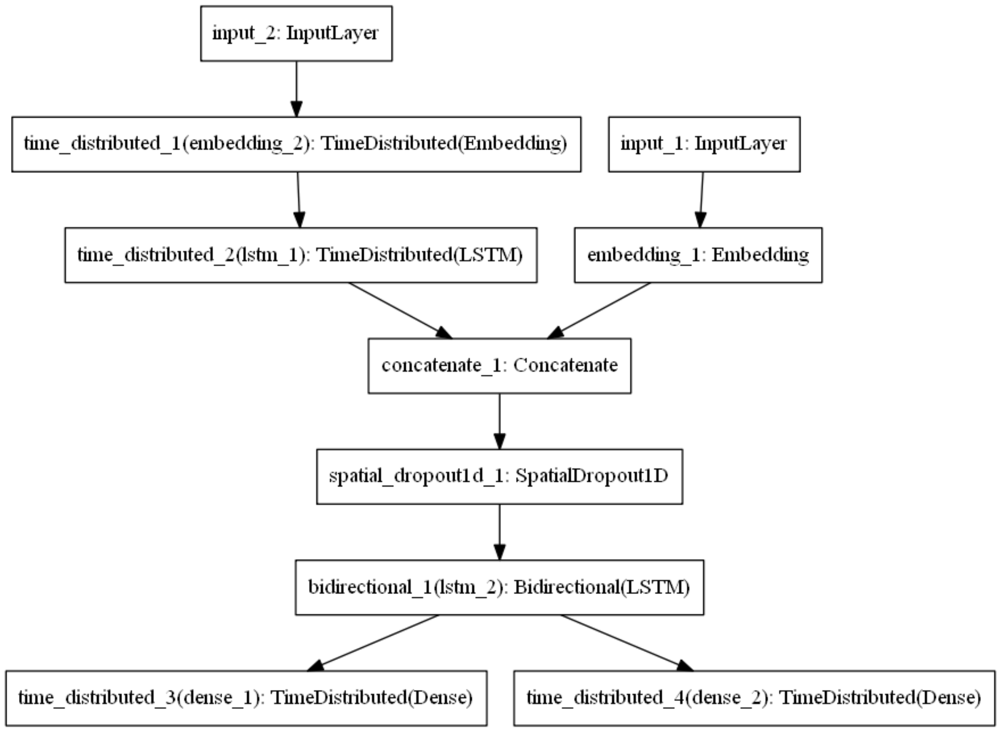

input_2: InputLayer
time_distributed_1(embedding_2): TimeDistributed(Embedding)
input_1: InputLayer
time_distributed_2(lstm_1): TimeDistributed(LSTM)
embedding_1: Embedding
concatenate_1: Concatenate
spatial_dropout1d_1: SpatialDropout1D
bidirectional_1(lstm_2): Bidirectional(LSTM)
time_distributed_3(dense_1): TimeDistributed(Dense)
time_distributed_4(dense_2): TimeDistributed(Dense)

Figure 28: Model setup for Multi-Task Learning (MTL)

8 Experimental Setup

This section describes the experiments that have been performed in this research. Section 8.1 describes evaluation metrics that have been used in this research for the evaluation of the results. Section 8.2 demonstrates the selection of hyperparameters for the models and Section 8.3 describes the evaluation of training/test split of datasets.

8.1 Evaluation Metrics

In a benchmark study, it is very crucial to provide success measures in a way that supports the comparison of methods and measure the performance of the model. This thesis is related to the usage of machine learning methods in a binary classification task. There are two types of labels for multi-task learning, arguments and relations. The argument has two classes 'arg' and 'O', where 'O' stands for others, likewise the relationship has two classes 'relation' and 'O'. The metrics used for the evaluation of the tasks are loss, accuracy, precision, recall, and F1-Score. Before understanding the metrics, the following definitions need to be known. In multi-task learning, the data has trained on two models, therefore let us define r as a type of relationship and s as a type of argument.

1. **True Positives (TP)**: The total number of accurate and successful predictions that were "Positive".

   (a) TP_r: as total number of successfully extracted relationships of type r.

   (b) TP_s: as total number of successfully extracted arguments of type s.

2. **False Positives (FP)**: The total number of inaccurate and incorrect predictions that were "Positive".

   (a) FP_r: as total number of incorrectly predicted relation tags that are said to be of type r but are not from type r

   (b) FP_s: as total number of incorrectly predicted argument tags that are said to be of type s but are not from type s.

3. **True Negative (TN)**: The total number of accurate predictions that were "Negative".

   (a) TN_r: as a total number of correctly predicted relation tags which are "Not Relevant" and were not of type r.

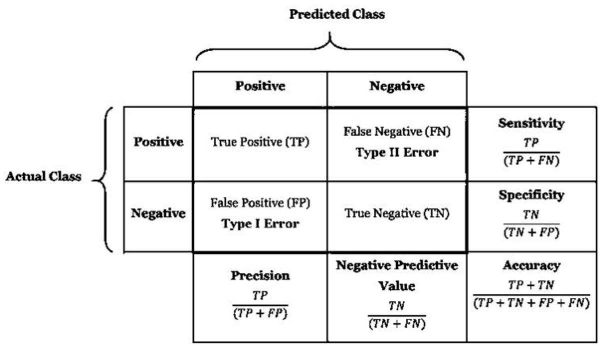
   (b) TN_s: as a total number of correctly predicted argument tags which are "Not Relevant" and were not of type s.

4. **False Negative (FN)**: The total number of inaccurate predictions that were "Negative". In this case, this is the total number of incorrectly predicted argument tags or relation tags which are "Not Relevant" and extracted relationships or arguments that are of type r or s but are said to be other type;

   (a) FN_r: as a total number of incorrectly predicted relation tags which are "Not Relevant" and extracted relationships that are of type r but are said to be other type.

(b) FN_s: total number of incorrectly predicted argument tags which are “Not Relevant” and extracted arguments that are of type s but are said to be other type.

5. **Confusion Metrics**: A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure 29: Confusion Metrics

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. In this case, the precision is the fraction of correctly retrieved relationships or arguments of type r or s respectively in the text, over the total number of retrieved relationships of type r or arguments of type s which explains how many are actually relevant. High precision relates to the low false-positive rate.

$$Precision = TP/(TP + FP)$$

Recall: Recall is the ratio of correctly predicted positive observations to all observations in an actual class. In this case, the recall is the fraction of relationships of type r or arguments of type s that are correctly extracted, over the total number of relationships of type r or arguments of type s present in the text. This describes all the words that are truly relevant to the labels or how many were labeled.

$$Recall = TP/(TP + FN)$$

F1-score : F1-score is the weighted average of Precision and Recall, that takes both false positives and false negatives into account. It’s better to look at both Precision and Recall if the cost of false positives and false negatives are very different.

$$F1Score = 2 * (Recall * Precision)/(Recall + Precision)$$

8.2 Evaluation Models

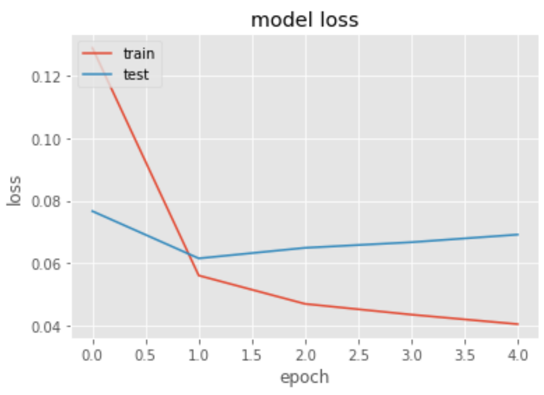
8.2.1 Evaluation of Model with pre-trained word embedding

	Hyperparameter	Choice	Experiment values
1	Learning rate	0.01	0.1, 0.01, 0.001
2	Epochs	5	3, 5, 10
3	Batch size	32	16, 32
4	Dropout	0.1	0.1, 0.2

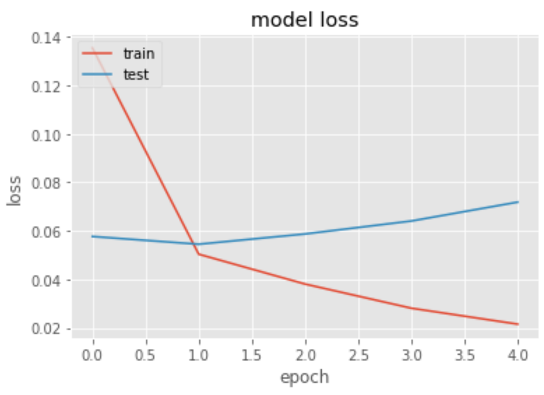
Table 4: Hyperparameters choice for LSTM with pre-trained embeddings

1. **Learning rate**

   The learning rate hyperparameter controls the speed at which model learns. Deep learning model uses stochastic gradient descent optimizer for training, which has various variations: Adam, RMSProp, Adagrad. These variations determine the learning rate, which tells the optimizer how far to move weights in the direction opposite of gradient for mini-batch. The training is more reliable if the learning rate is low, but optimizer takes a lot of time because of the smaller steps towards the minimum of the loss function. The training may not converge or even diverge if the learning rate is high, as weight changes can vary. This could lead the optimize to surpass/overshoot the minimum and aggravate the loss. The best way to find a suitable learning rate for the training is to pass it through different learning rate settings, in order to check which setting gives the least loss without sacrificing the speed of training. In this case, the learning rate was set at different values of 0.1, 0.01 and 0.001. The 0.01 has given the best performance and minimum model loss, as shown in Figure 30. The validation_split was set to be 0.2, which is the percentage of the size of training dataset. The output on each epoch shows the loss and accuracy of both the training dataset and the validation dataset.

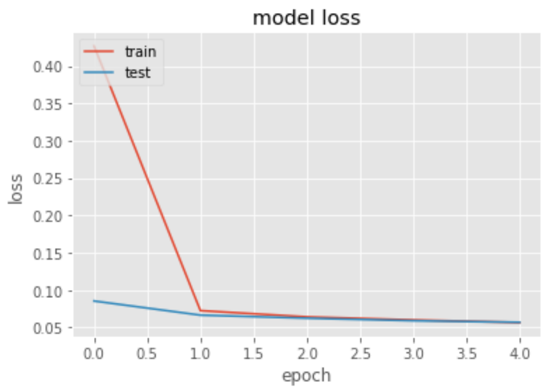


(a) Learning rate = 0.1



(b) Learning rate = 0.01

Figure 30: Selection of learning rate for the model

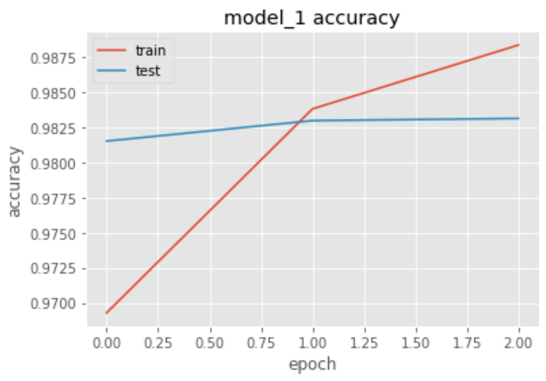


(c) Learning rate = 0.001

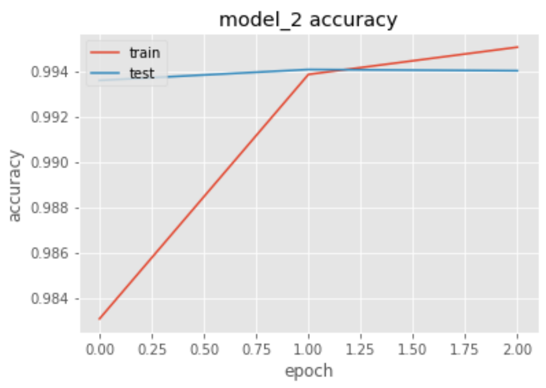
Figure 30: Selection of learning rate for the model

2. **Selection of Epochs**

When a dataset is passed back and forth through the neural network, one epoch is formed. The entire dataset must be passed multiple times through a neural network because of gradient descent which is an iterative process. As the number of epochs increases, several times the weight tends to change more in the neural network and the curve goes from under-fitting to optimal to over-fitting curve. In this case, the model is trained with Adam optimizer and for different number of epochs: 3, 5 and 10. Eventually, 5 epochs were selected, as training of more epochs would cause the model to overfit on the training data, and reduce the accuracy, as shown in the below figures.

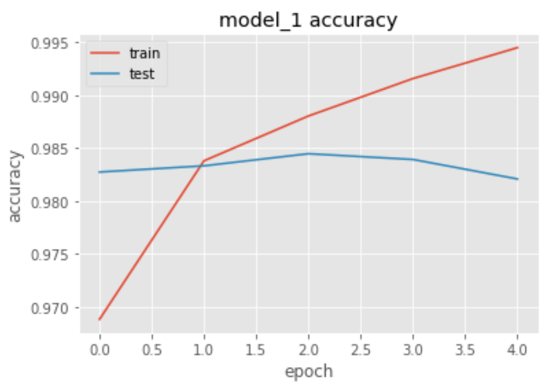


(a) Model 1

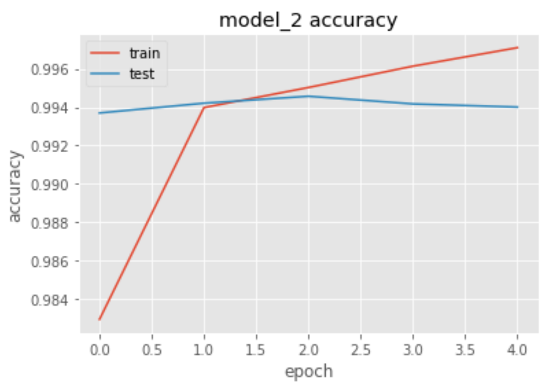


(b) Model 2

Figure 31: Epoch=3, learning rate=0.01

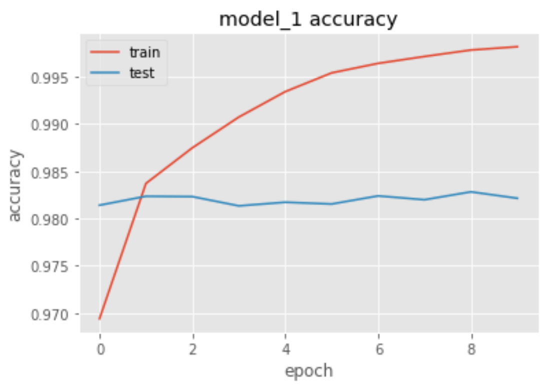


(a) Model 1

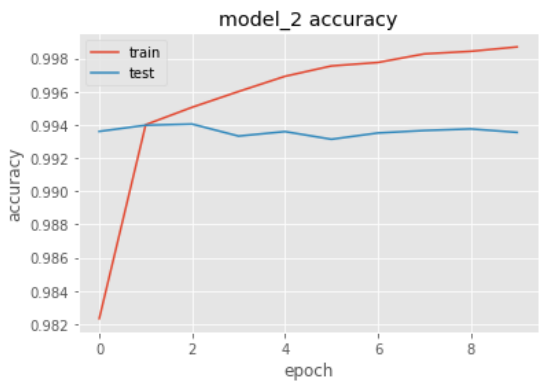


(b) Model 2

Figure 32: Epoch=5, learning rate=0.01



(a) Model 1

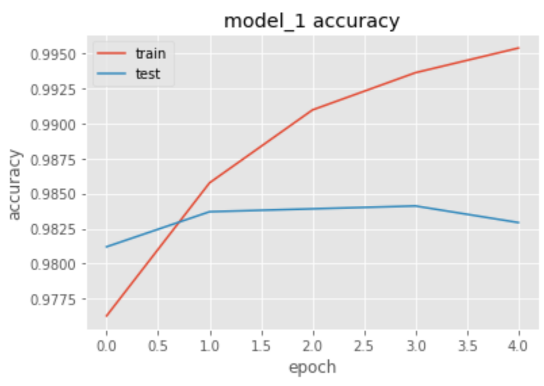


(b) Model 2

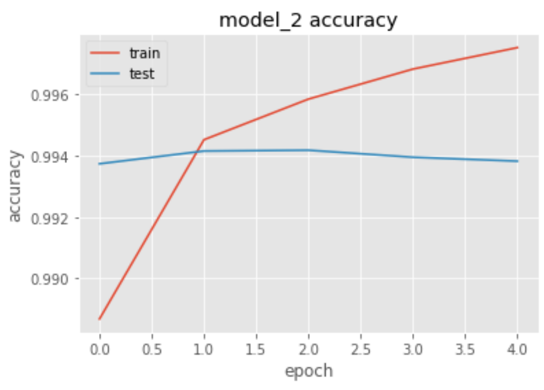
Figure 33: Epoch=10, learning rate=0.01

3. **Selection of batch size**

The batch size defines the number of samples that will be propagated through the network. If we have 1000 training examples, and the batch size is 500, then it will take 2 iterations to complete 1 epoch. Batch size is one of most hyper-parameters which should be chosen carefully. If the batch size is too large, it can lead to poor generalization, whereas smaller batch sizes have faster convergence to a good solution. In this case, we have trained the model with two different batch sizes – 16 and 32; the resulting outcomes were almost the same as shown in the below figures. Therefore, we selected batch 32 for this training process.

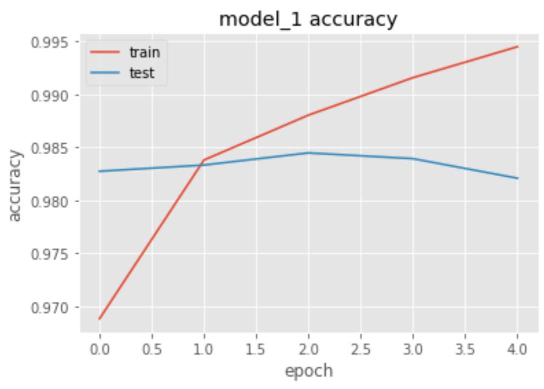


(a) Model 1

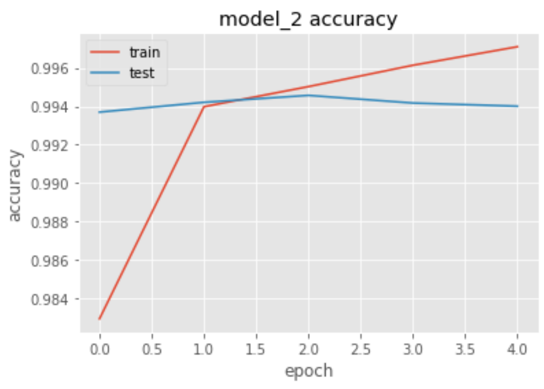


(b) Model 2

Figure 34: Epoch=5, learning rate=0.01, batch size=16



(a) Model 1

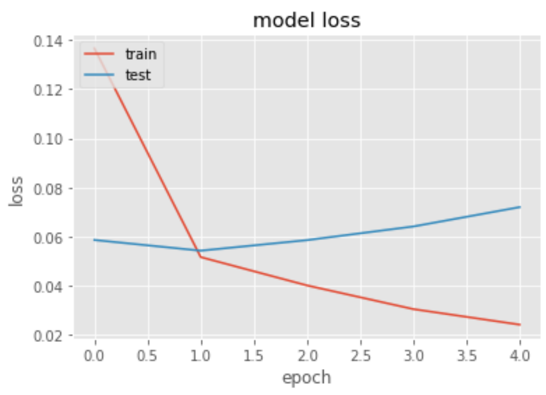


(b) Model 2

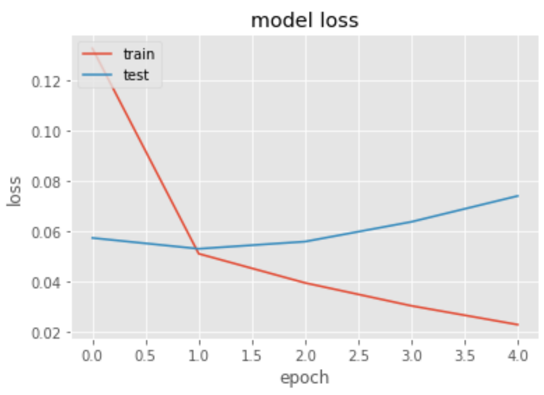
Figure 35: Epoch=5, learning rate=0.01, batch size=32

4. **Selection of dropout**

   Dropout is a regularization method to prevent overfitting of the model which approximately trains a large number of neural networks with different hidden layers in parallel. During training, a certain number of layer outputs are randomly ignored or “dropped out” which is a simple way to prevent over-fitting of the model. In this case, dropout has been set to 0.1 because it has a minimum loss. The dropout may or may not be necessarily used as the multi-task learning already took care of the regularization in the model, which reduced noise and prevented overfitting.



(a) Dropout=0.1



(b) Dropout=0.2

Figure 36: Selection of dropout for the model

8.2.2 Evaluation of Model without pre-trained word embedding

	Hyperparameter	Choice	Experiment values
1	Learning rate	0.01	0.1, 0.01, 0.001
2	Epochs	5	3, 5, 10
3	Batch size	32	16, 32
4	Dropout	0.1	0.1, 0.2

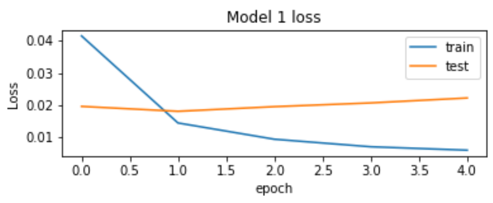
Table 5: Hyperparameters choice for LSTM without pre-trained embeddings

The model is also evaluated without pre-trained word embedding. We have created a dictionary for the words and tags, which is further trained using hyper-parameters mentioned in Table 5. Model 1 is trained on argument tags, whereas model 2 have trained on relation tags. It is crucial to select the right hyper-parameters for the model, because it would maximize the accuracy rate and minimize the loss of the model. The hyper-parameters, mentioned in Table 5, managed to get good accuracy and minimum loss, as shown in Figure 37. The overall loss of the entire model is 0.02, and the accuracy of both models reached to almost 0.99 at epoch 3, which shows the selection hyper-parameters are correct and effective.

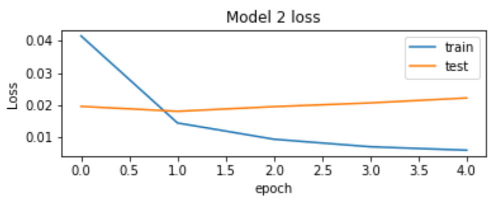
8.3 Evaluation Datasets

For the training of the model, it was evaluated on several train-test splits of the dataset. Two datasets were used for the extraction for relations and arguments, expert annotated, and crowd-sourced annotated dataset. The expert annotated dataset contains 2000 sentences with 2038 relations and 4094 arguments, as shown in Table 6. The model has been evaluated on different size of training and test sets, mentioned in Table 7.

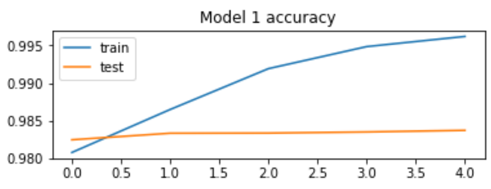
However, for the crowdsourced annotations, some issues emerged from the crowdsourcing site, such as not being able to capture the crowdsourced annotations, even though the interface worked fine and certain expert annotations were generated through it. This issue is further discussed in



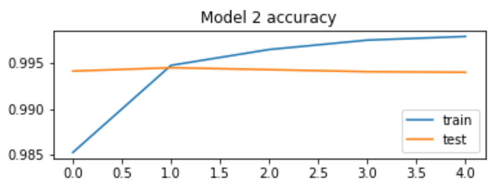
(a) Model 1- Argument tagger - Model loss



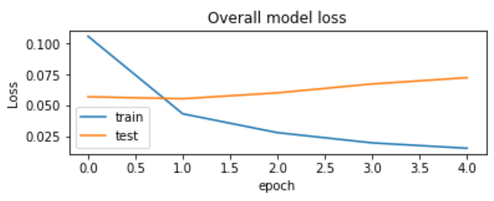
(b) Model 2- Relation tagger - Model loss



(c) Model 1- Argument tagger - Model accuracy



(d) Model 1- Relation tagger - Model accuracy



(e) Overall MTL model loss

Figure 37: Selection of hyper-parameters for the model without word embedding

Section 9.4.

Corpora	**Sentences**	**Arguments**	**Relations**
Expert annotations	2000	4094	2038
crowdsourced annotations	10000	-	-

Table 6: Evaluation Datasets

Test Size	Training Size	**No. of samples (validation set=0.2)**	**No. of samples (train set)**
0.1	0.9	364	1452
0.2	0.8	323	1291
0.3	0.7	283	1129
0.4	0.6	242	968
0.5	0.5	202	807

Table 7: Training/test data splits

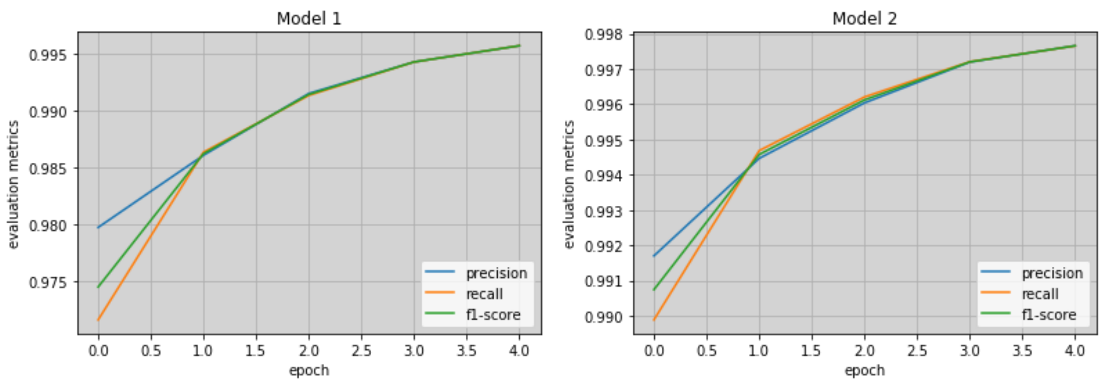
9 Evaluation Results

The evaluation results given in Section 9.1 pertains to the dataset containing sentences, relations and arguments. The model robustness is reported in Section 9.2. The discussions about the ablation and limitations of this research are discussed in Sections 9.3 and 9.4, respectively.

9.1 Results Discussion and Analysis

In this research, the sequence-to-sequence model is implemented along with Multi-Task Learning (MTL), which additionally tested with the two types of word embeddings, pre-trained and without pre-trained word embeddings. LSTM model was trained for the binary classification problem, which served to classify the relations and arguments using two models. In both cases, it has given good results concerning evaluation metrics. Figure 38 and Figure 39 demonstrate the comparison between two training models, with and without word embedding in which model 1 worked on the task of argument tagging and model 2 focused on the task of relation tagging.

Figure 38 illustrates that model 1 evaluation metrics improved after the first epoch. It reached to almost 0.99 after second epoch whereas model 2 metrics reached to 0.99 at epoch 1. Both models give high-grade demanding metrics.

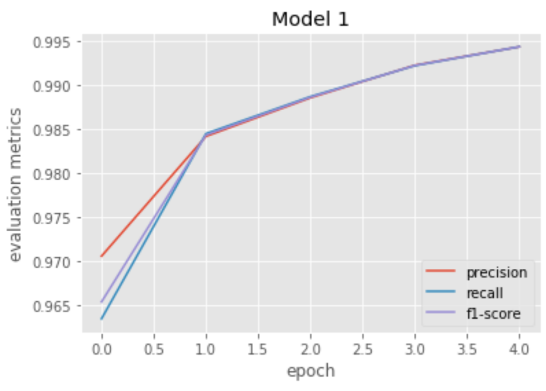


(a) Model 1 - evaluation metrics

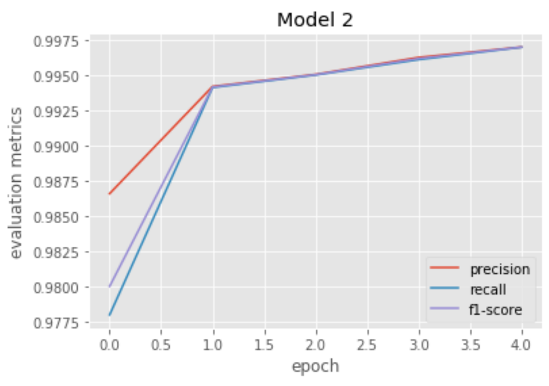
(b) Model 2 - evaluation metrics

Figure 38: Model results without pre-trained word embeddings

Figure 39 demonstrates that model 1 with pre-trained embedding which has a minimum 0.97 of recall, 0.98 of precision and 0.97 of f1-score. All these metrics collectively improved at second epoch and exhibited almost 0.98 score. which slightly increased at each epoch and reached to 0.99. This shows that a better performance of the model was accomplished.



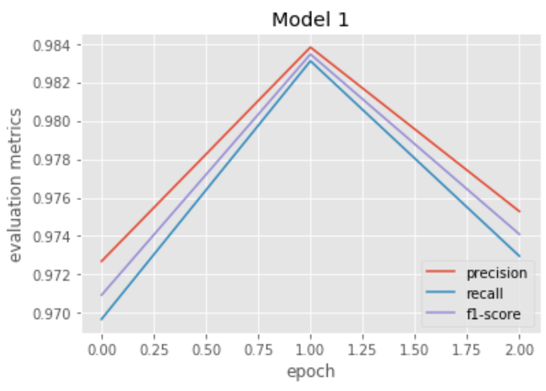
(a) Model 1 - evaluation metrics



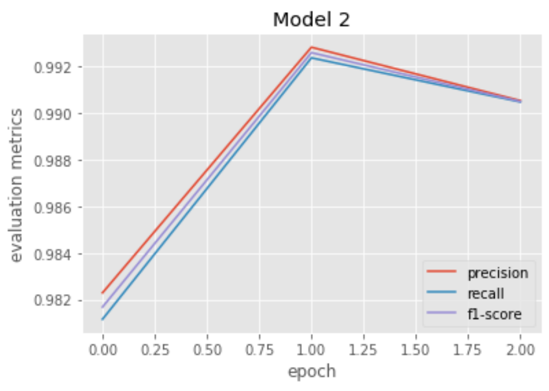
(b) Model 2 - evaluation metrics

Figure 39: Model results with pre-trained word embeddings

The model was also tested by including character embeddings, which were discussed in Section 7.3. Figure 40 shows that the model has over-fitted after second epoch due to less training data. In order to avoid this issue, the model used the early stopping technique to stop it from over-fitting, whereby the training comes to a halt prior to convergence, so as to avoid over-fitting. The overfitting had stopped at epoch 3 because from there on data over-fitted in both models.



(a) Model 1 - evaluation metrics



(b) Model 2 - evaluation metrics

Figure 40: Model results with pre-trained GloVe word embeddings and character embeddings

9.2 Model Robustness

Precision and recall have been used for the evaluation of extraction quality of relation and arguments, as mentioned in Section 8.1. Besides, these measurements estimate the generality of the approach by comparing accuracy, loss, recall, precision, and f1-score of different test sizes and training sizes. Precision describes the overall correctness of the approach, whereas recall defines overall completeness of the approach, furthermore, f1-score is the harmonic mean of recall and precision. However, training a binary classifier with expanded training sets improved metric scores, and underscoring its robustness to noise.

This section describes different experiments to evaluate the performance of models against different test sizes and training sets. The experiment compares the performance of the model by increasing training set size.

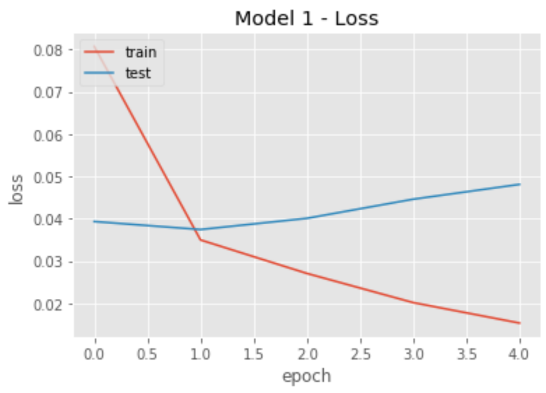
9.2.1 Test Size 0.1, Training Size 0.9

The model has been split into 0.1 of testsize and 0.9 of training set size. Here the training data is more than test data that has increased the quality of results.

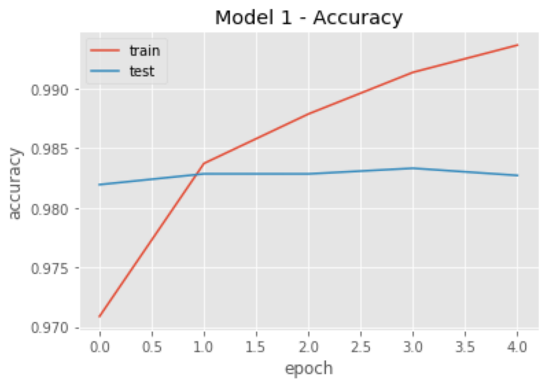
1. **Model 1 Results: Argument Tagging**

   The first model identifies those arguments in the sentences which give a maximum loss of 0.08, and a minimum loss of 0.01, The loss is quite low, mostly due to the accurate selection of the learning rate. This further means that the entire model loss is lower as well. The minimum accuracy of the model is 0.97 and the maximum accuracy of the model is 0.99 at epoch 3, which shows the generalization to be better as more data was present in the training set.

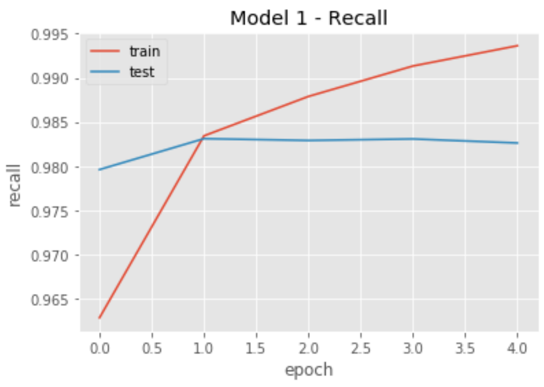
   The lowest recall value is 0.96 at epoch 0, which slightly increased to 0.99 at epoch 5. This shows that most of the returned results are relevant. The precision is 0.97 in the start; later it reaches to 0.99, which shows the returned results to be substantially more relevant than the irrelevant ones. F1-score explains the combined measurement of recall and precision that has minimum f1-score of 0.96 and a maximum of 0.99, which shows good accuracy of the model classification.



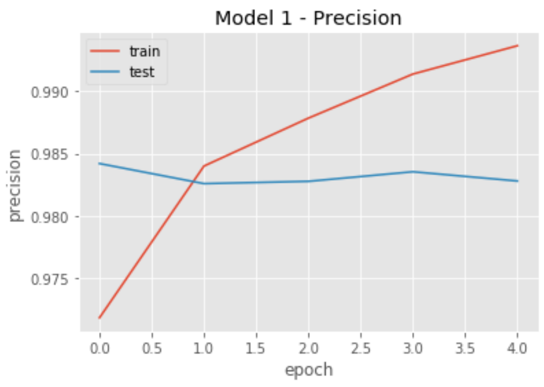
(a) Model 1-Loss



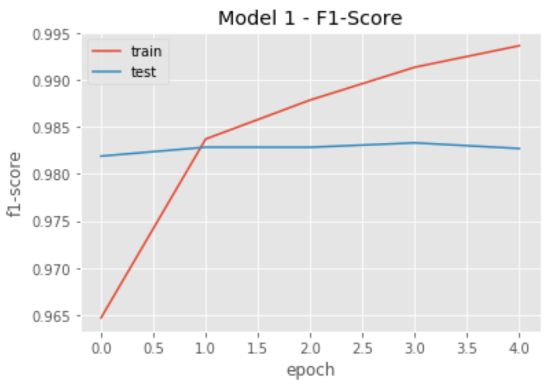
(b) Model 1-Accuracy



(c) Model 1-Recall



(d) Model 1- Precision

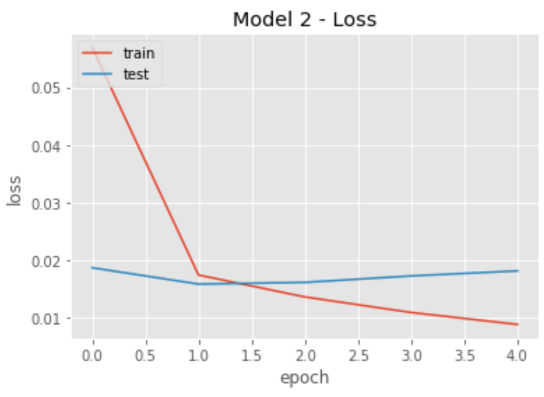


(e) Model 1- F1 Score

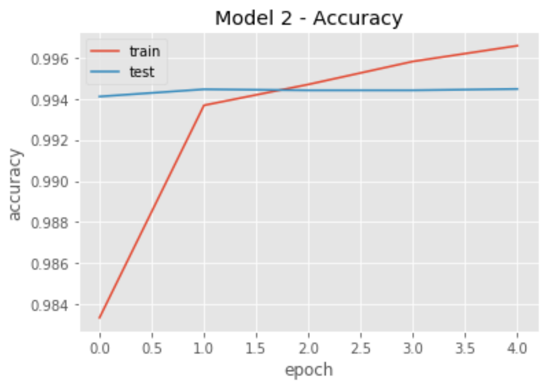
Figure 41: Model 1 results: argument tagging (training size=0.9)

2. **Model 2 Results: Relation Tagging**

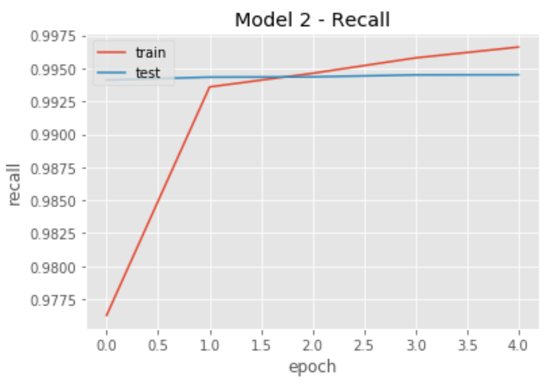
Similarly, model 2 has identified the relations among sentences using binary classification which gives high precision (0.99), high recall (0.99), and high f1-score(0.99) from epoch 2 to 5 which determines the better generalization and good classification of the model as shown in Figure 42.



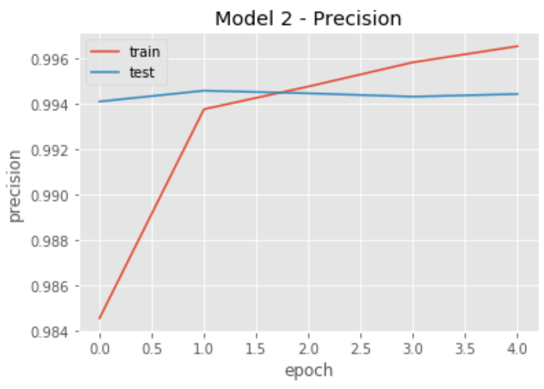
(a) Model 2-Loss



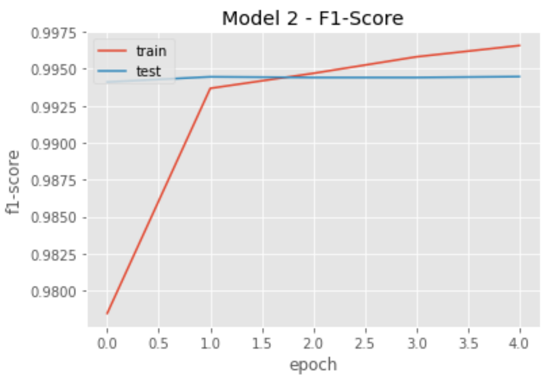
(b) Model 2-Accuracy



(c) Model 2-Recall



(d) Model 2- Precision



(e) Model 2- F1 Score

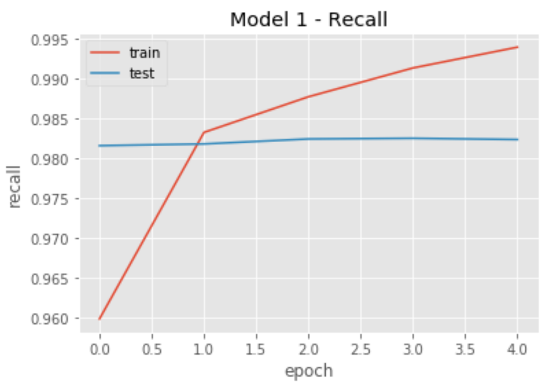
Figure 42: Model 2 results: relation tagging (training size=0.9)

9.2.2 Test Size 0.2, Training Size 0.8

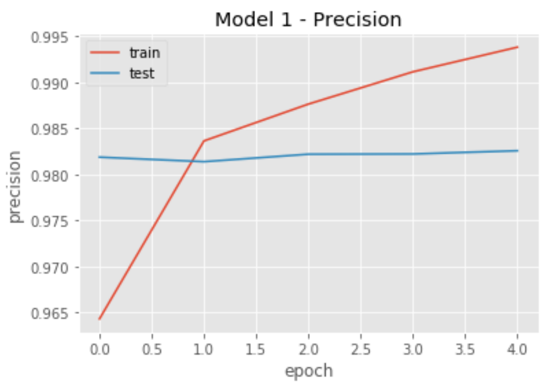
We have decreased the training set to 0.8 and increased the test size in order to analyze the robustness of the model.

1. **Model 1 Results: Argument Tagging**

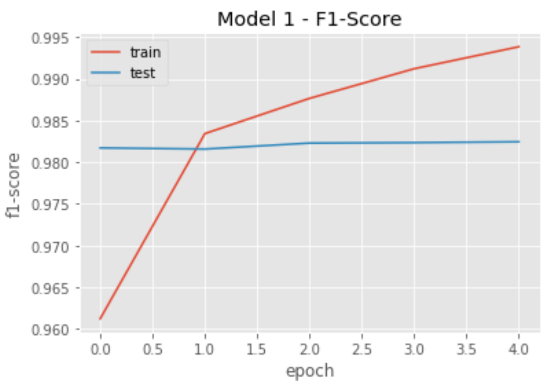
   Recall identifies the completeness of the approach, whereas precision describes the overall correctness of the approach. Figure 43 demonstrates the overall performance of the model which has minimum recall (0.96), minimum precision (0.96) and f1-score (0.96) where the maximum value is almost 0.99. This shows that the training data was classified adequately as both the recall and precision values are high. Higher precision means a lower false-positive rate. Higher recall means a lower false-negative rate.



(a) Model 1-Recall



(b) Model 1- Precision

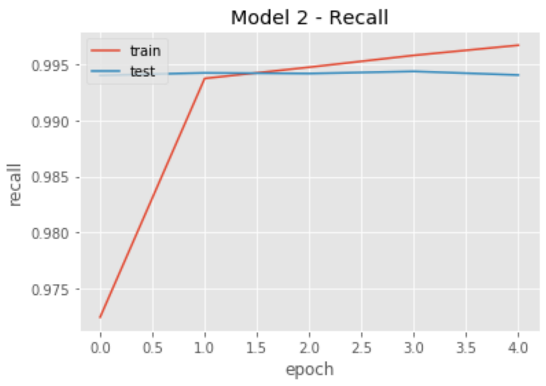


(c) Model 1- F1 Score

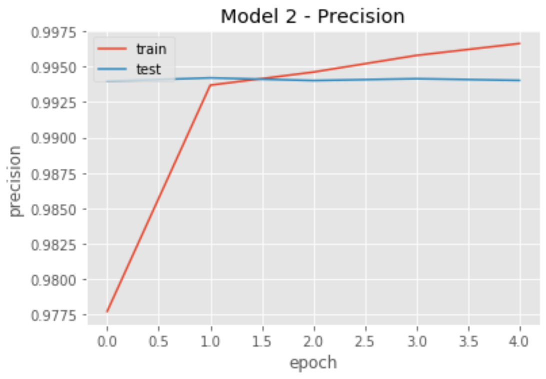
Figure 43: Model 1 results: argument tagging (training size=0.8)

2. **Model 2 Results: Relation Tagging**

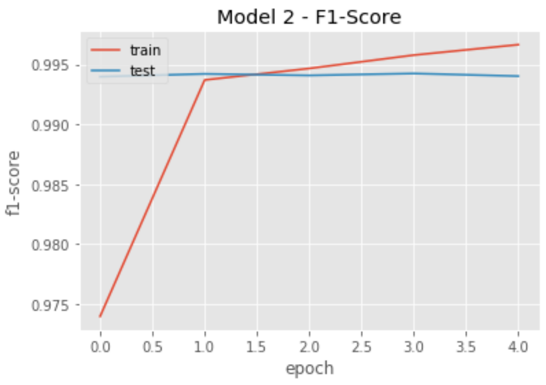
   Model 2 has a task of tagging relations among sentences which have high precision and high recall of 0.99, as shown in Figure 44. The least recall and least precision values are 0.97, meaning that the model executed well on the training data at the validation split of 0.2.



(a) Model 2-Recall



(b) Model 2- Precision



(c) Model 2- F1 Score

Figure 44: Model 2 results: relation tagging (training size=0.8)

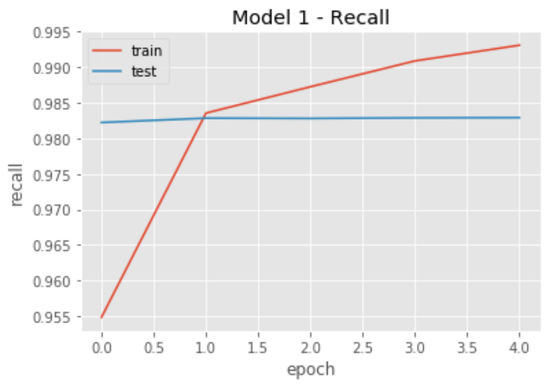
9.2.3 Test Size 0.3, Training Size 0.7

The model was trained again on the different sizes of test-train sets which were set at test size of 0.3 and train set size of 0.7. We have decreased the training set a little bit to check the performance of the model with less training data.

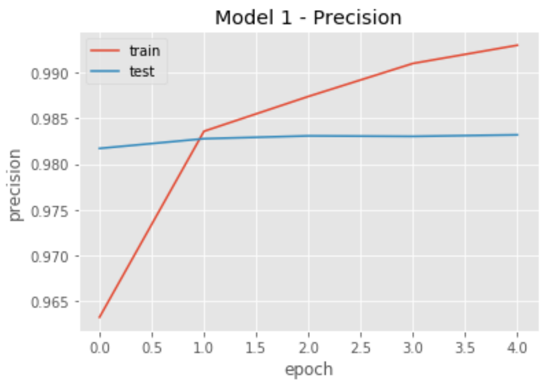
1. **Model 1 Results: Argument Tagging**

   By modifying the training size, the recall decreased to 0.95 from 0.97 (from the previous training set of size 0.8 and 0.9) at the start and later it got slightly improved at second epoch.

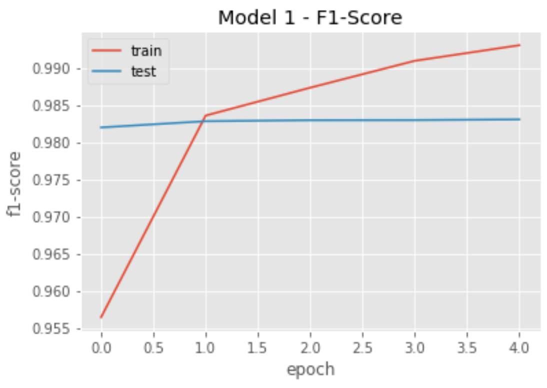
   Similarly, precision has minimized to 0.96 at second epoch, but later on it slightly extended to 0.99, as shown in Figure 45. Whereas minimum f1 score reached to 0.95 and the maximum reached to 0.99.



(a) Model 1-Recall



(b) Model 1- Precision

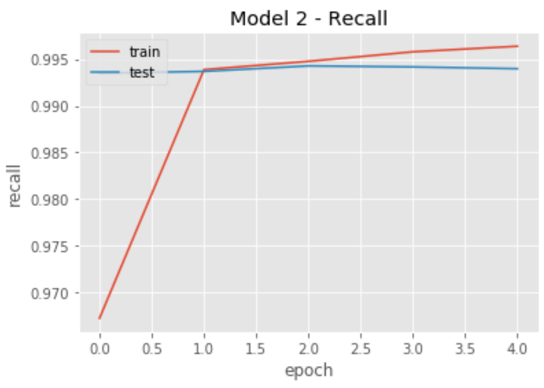


(c) Model 1- F1 Score

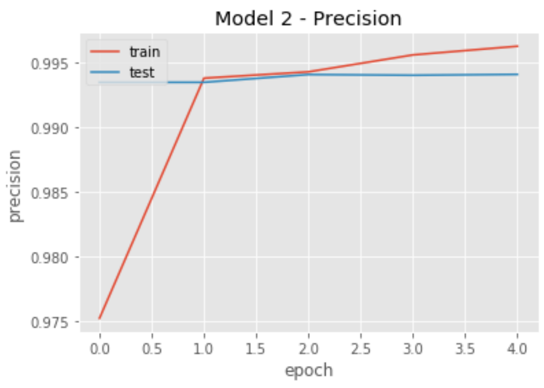
Figure 45: Model 1 results: argument tagging (training size=0.7)

2. **Model 2 Results: Relation Tagging**

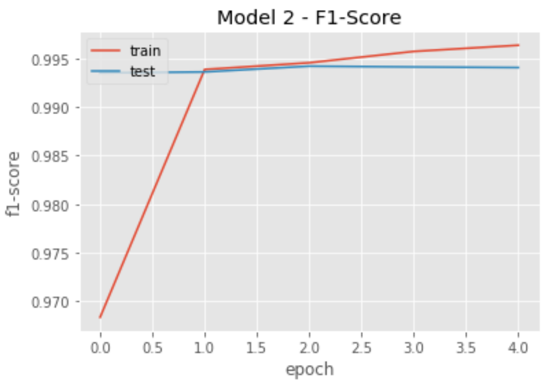
   Figure 46 illustrates the performance of relation tagging model in which minimum value reached to 0.97 and maximum value reached to 0.99 for all measures, which shows the model worked well at training size of 0.7.



(a) Model 2-Recall



(b) Model 2- Precision



(c) Model 2- F1 Score

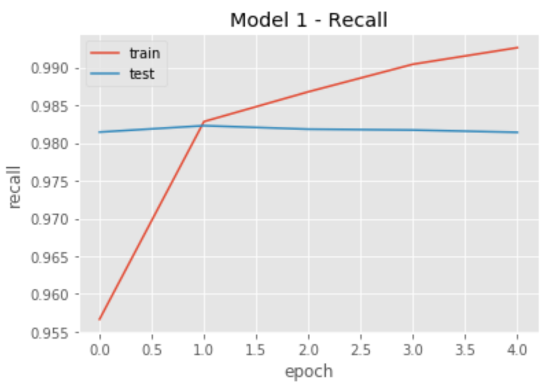
Figure 46: Model 2 results: relation tagging (training size=0.7)

9.2.4 Test Size 0.4, Training Size 0.6

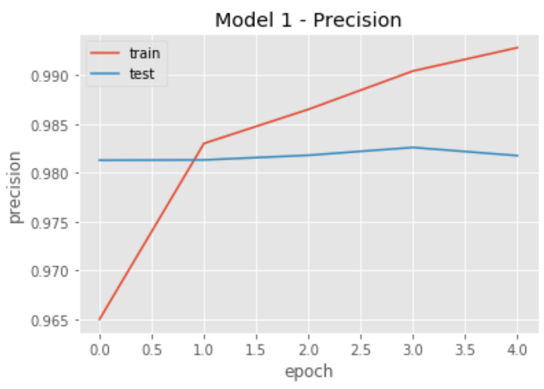
For the evaluation of the model, again, the training and test sizes were changed to 0.4 and 0.6 respectively.

1. **Model 1 Results: Argument Tagging**

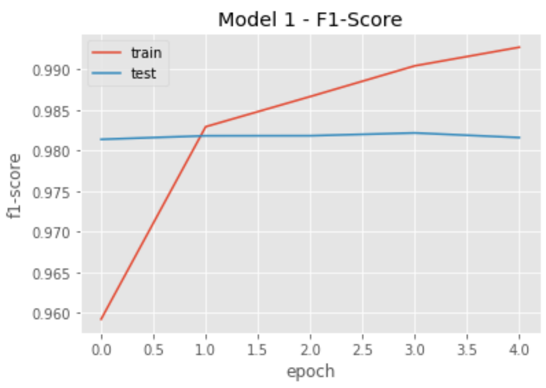
   Figure 47 demonstrates the minimization of the model-1 recall to 0.95, and of precision to 0.96. This led to a minimum f1 score of 0.96, demonstrating the impact of decreasing the training date. The model became more effective and better on reaching epoch 2. As for the minimized values of recall and precision, it shows the importance of increasing training data in a model.



(a) Model 1-Recall



(b) Model 1- Precision

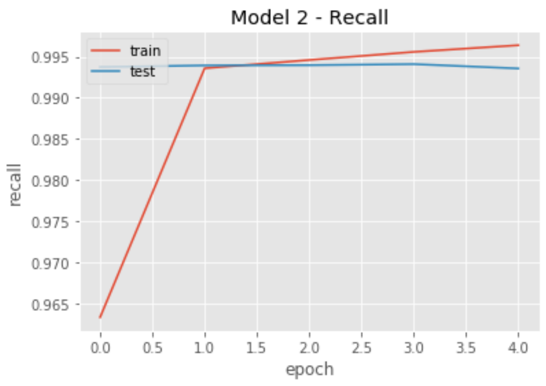


(c) Model 1- F1 Score

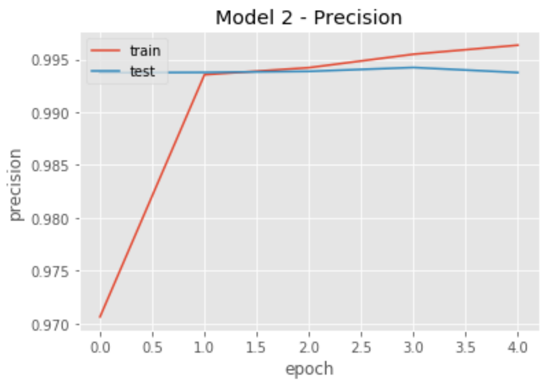
Figure 47: Model 1 results: argument tagging (training size=0.6)

2. **Model 2 Results: Relation Tagging**

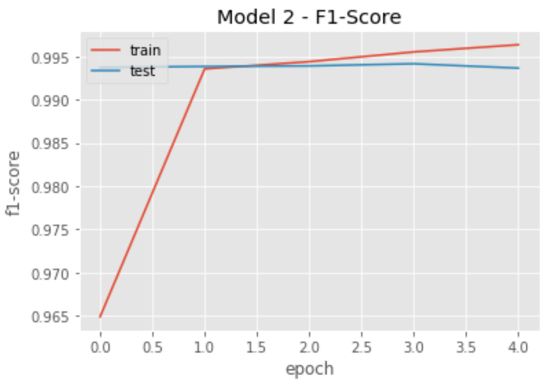
   Figure 48 provides the description of model 2 performance, which has the minimum recall (0.96), precision (0.97) and f1-score (0.96), which slightly maximized and reached to the 0.99.



(a) Model 2-Recall



(b) Model 2- Precision



(c) Model 2- F1 Score

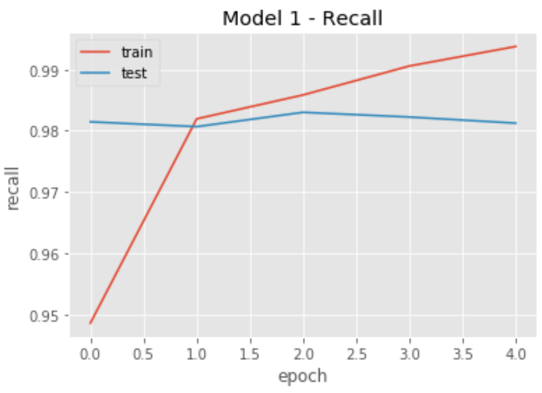
Figure 48: Model 2 results: relation tagging (training size=0.6)

9.2.5 Test Size 0.5, Training Size 0.5

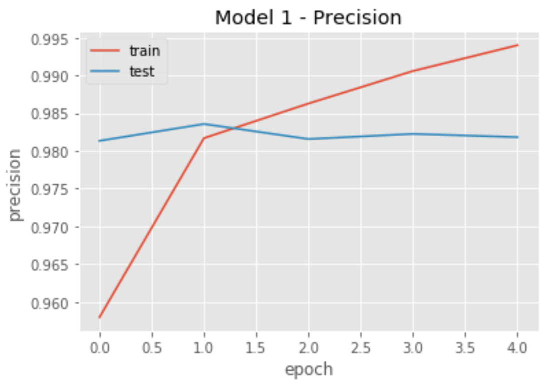
In the last test, we have divided both test and training size in half.

1. **Model 1 Results: Argument Tagging**

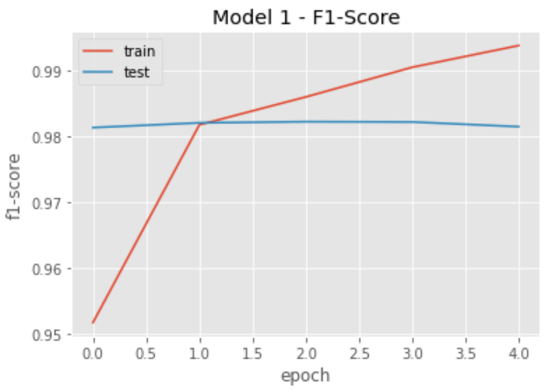
   Figure 49 describes the model having a training size of 0.5, which gives minimum recall of 0.95, and a minimum precision of 0.96. Furthermore, the maximum recall reached to 0.98 at second epoch, which slightly increased and reached to 0.99 whereas precision has also reached to 0.99. The maximum f1 score for this model is 0.99 and the minimum is 0.95, which demonstrates the model has been implemented well.



(a) Model 1-Recall



(b) Model 1- Precision

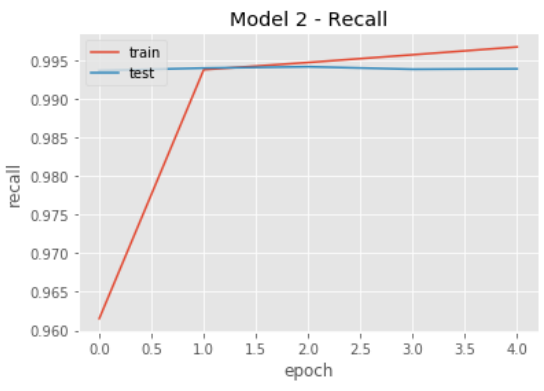


(c) Model 1- F1 Score

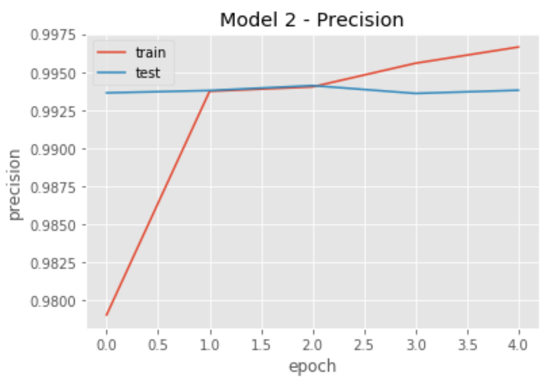
Figure 49: Model 1 results: argument tagging (training size=0.5)

2. **Model 2 Results: Relation Tagging**

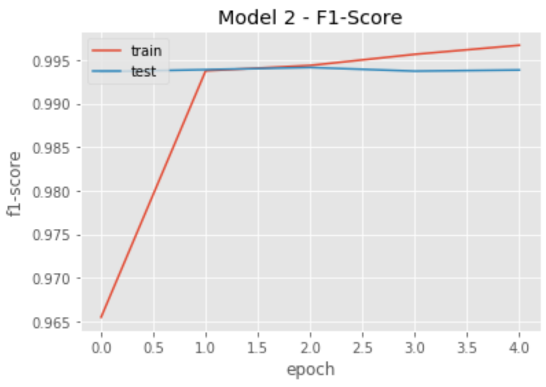
   Similarly, model 2 (relation tagging) has given good results, but its recall and precision have affected little bit however decreasing the recall and precision of the model to some extent. This occurred due to lesser training data, thus implicating that more training data can improve the model to generalize better and perform well. In machine learning, it is crucial to train the model with more training examples for the best results.



(a) Model 2-Recall



(b) Model 2- Precision



(c) Model 2- F1 Score

Figure 50: Model 2 results: relation tagging (training size=0.5)

9.3 Ablations

Following are the ablations of this research:

1. Annotation interface provides crowd-workers with the tool of argument and relation selection. These manual annotations has used as training data for the n-ary relation extraction model.

2. Argument and relation labels ,respective of their positions, are generated from the annotation interface in order to generate n-ary relation in MTL model.

3. The model has used GloVe pre-trained embeddings for the training of data.

9.4 Limitations

Following are the limitations of this research:

1. The first limitation of the model is to choose Noun Phrases (NPs) as arguments which has been discussed in Section 2.2.4.

2. Second limitation is the selection of those NPs which do not consist of other NPs, as such arguments would then hamper the implementation of the interface.

3. Another limitation is that the disjoint sets of arguments and relations in Multi-Task Learning MTL may mix up their labels.

4. The model had to be trained via two datasets, one from the expert annotations and the other from the crowdsourcing as crowdsourced annotations, mentioned in Section 6. The training was successful in the case of expert annotations; however, certain issues came forth with the crowdsourcing site. We did test the interface several times and were able to generate expert annotations from it. Nevertheless, mostly due to time-constraint, we were not able to capture the crowdsourced annotations. Hence, this work can be done on these annotations in the future.

10 Conclusions and Future Outlook

Relation extraction is a highly promising technology for converting unstructured data to structured data, that can be effectively used for querying and automated reasoning. Prior work has primarily focused on the binary relation extraction, but now n-ary relation extraction has come into demand, provided that it imparts more enriched data than the binary relation extraction. In this thesis, we have investigated multi-task learning with LSTM model for n-ary relation extraction. Several contributions have been made by this research for ensuring the formation of a well-rounded system for n-ary relation extraction.

The first contribution of this thesis is the extraction of the dataset, which includes Web scraping of Wikipedia's Portal Current Events (WCEP), which resulted in extracting 10,000 sentences from the website from 2016-2018. Secondly, the extracted event summary has been used for the extraction of noun phrases which do not constitute other NPs, in order to avoid overlapping in the annotation interface. This natural language processing task has been done through constituency parser of CoreNLP.

Furthermore, these NPs are used as arguments in an interface so that they can be selected as necessary or unnecessary. The second contribution is the implementation annotation interface which is used for the creation of dataset for training models. The annotation interface is used to annotate arguments and relations in a sentence, mostly done by experts and crowdsourced workers. As a result, two datasets come forth; one from experts and other from crowd workers.

Moreover, after the creation of these datasets, we have used the multi-task learning approach to generate n-ary relations. In the background of multi-task learning, an LSTM model is developed for the binary classification problem, which serves to classify the relations and arguments using two models. The first model identifies if a relation exists, after which it assigns a number "1" to it. If a relation does not exist, it assigns the number "0" to the class. Similarly, the second model performs the same role i-e if it finds an argument, it classifies it as "1", otherwise as "0". In this way, two models run simultaneously (multi-task learning), which learns n-ary relations among sentences.

For the evaluation of the model, evaluation metrics were applied on both models to check and monitor performance, such as accuracy, loss, precision, recall and f1-score. The results were optimal, deeming the models to have worked well. The improved performance is due to greater precision and recall, caused by the Relaxed Semantic Matching, which is enabled by computing similarities based on word embeddings. In the comparison of different test-train sizes, it was found that additional training data was required to get more results and relation extraction. We believe that the work presented in this thesis will serve to be a great contribution to make IE systems better and innovative, enhancing the complex relation extractions from richer textual data.

Future work on the lines of this research could focus upon the extraction of greater crowdsourcing data so that more training data can be made available for finer model training. Second, BERT model could be incorporated for n-ary relation extraction, or variant machine learning algorithms that could target the comparison of taking performances on different models.

Another addition could be relation classification, which could classify relationship type for each relation, such as Cause-Effect, Destination, Death-cause, Attack, Breaks-off and death or burial face. We have also implemented LSTM model for binary classification on two models, what can be done further is that more elaborate models can be introduced after creating different types of relationships; this would serve to add more classes to the model and find more relations within the text.

Another important area of future work could be to use the ‘GRU layer’ instead of LSTM and create an attention layer on top of it. The performance of the LSTM with attention is also impressive, but we can still improve the result by replacing LSTM with ‘Gated recurrent units’(GRU). GRU is simpler and thus easier to modify and does not need memory units; therefore, it is faster to train than LSTM.

Lastly, crowdsourced annotations can be taken as the future work of this research. Although there were some issues with the crowdsourcing site i-e failure to deploy an interface properly, the interface turned out to work better and was deployed adequately later on. Also, due to time constraint, we were not able to capture these annotations.

References

N. Bach and S. Badaskar. A review of relation extraction. *Language Technologies Institute, Canergie Mellon University*, 2017.

J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28:7–39, 1997.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. *In Proceedings of ICML*, pages 41–48, 2009.

R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

Y. S. Chan and D. Roth. Exploiting background knowledge for relation extraction. in proceedings of the 23rd international conference on computational linguistics. *Association for Computational Linguistics*, page 712–717, 2010.

G. Chen. A gentle tutorial of recurrent neural network with error backpropagation. *Retrieved from http://arxiv.org/abs/1610.02583*, page 1–9, 2016.

N. A. Chinchor. Overview of muc-7/met-2. proceedings of the seventh message understanding conference (muc-7). 1998.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. *In Proceedings of ICML*, pages 160–167, 2008.

L. Deng and D. Yu. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7:3–4, 2014.

Duong, L., Cohn, T., and Bird. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, page 845–850, 2015.

P. Ernst, A. Siu, and G. Weikum. Highlife: Higher-arity fact harvesting. *International World Wide Web Conference Committee*, 2018.

O. Etzioni, A. Fader, and M. Janara Christensen, Stephen Soderland. Open information extraction: the second generation. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume One.*, pages 3–10, 2011.

N. FitzGerald, O. Tackstrom, K. Ganchev, and D. Das. Semantic role labeling with neural network factors. in proceedings of the conference on empirical methods in natural language processing. in proceedings of the conference on empirical methods in natural language processing. *Lisbon, Portugal, September. Association for Computational Linguistics*, page 960–970, 2015.

M. Gerber and J. Y. Chai. Beyond nom-bank: A study of implicit arguments for nominal predicates.in proceedings of the 48th annual meeting of the association for computational linguistics, acl '10,. *Stroudsburg, PA, USA. Association for Computational Linguistics.*, page 1583–1592, 2010.

Z. GuoDong, Z. J. Su Jian, and Z. Min. Exploring various knowledge in relation extraction. in proceedings of the 43rd annual meeting on association for computational linguistics. *Association for Computational Linguistics.*, page 427–434, 2005.

Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *The MIT Press ISBN 9780262018258*, 2015.

P. Jackson and M. I. Natural language processing for online apllications: Text retrievel, extraction and categorization. *Amsterdam, NLD: John Benjamins Publishing Company. Retreived from http://ww.ebrary .com*, 2007.

N. Kambhatla. Combining ltactic,and semantic features with maximum entropy models for extracting relations. in proceedings of the 42nd annual meeting on association for computational linguistics,demonstration sessions. *Association for Computational Linguistics*, 2004.

R. McDonald, F. Pereira, S. W. Seth Kulick, Y. Jin, and P. White. Simple algorithms for complex relation extraction with applications to biomedical ie. *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics.*, page 491–498, 2005.

T. Mikolov, J. Kopecky, O. G. Lukas Burget, and J. Cernocky. Neural network based language models for highly inflective languages. *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'09 IEEE Computer Society*, 2009.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *In Proceedings of Workshop at International Conference on Learning Representations ICLR'13*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *In Advances in Neural Information Processing Systems, NIPS'13. Curran Associates, Inc.,*, pages 3111–3119, 2013b.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. in proceedings of the joint conference of the 47th annual meeting of the acl and the 4th international joint conference on natural language processing of the afnlp: Volume 2-volume 2. *Association for Computational Linguistics.*, page 1003–1011, 2009.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. Foundations of machine learning. *The MIT Press ISBN 9780262018258*, 2012.

T. H. Nguyen and R. Grishman. word representations and regularization for domain adaptation of relation extraction. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, page 68–74, 2014.

N. Pengl, H. Poon, C. Quirk, K. Toutanova, and W. tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics (TACL) 2017*, 5, 2017.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *In Proceedings of EMNLP*, page Jeffrey, 2014.

H. Poon, C. Quirk, C. DeZiel, and D. Heckerman. Literome: Pubmed-scale genomic knowledge base in the cloud. bioinformatics. 2014.

K. Reschke, M. Jankowiak, M. Surdeanu, C. D. Manning, , and D. Jurafsky. Event extraction using distant supervision. in proceedings of 8th edition of the language resources and evaluation conference. 2014.

M. Roth and M. Lapata. Neural semantic role labeling with dependency path embeddings. in proceedings of the 54th annual meeting of the association for computational linguistics. *Berlin, Germany, August. Association for Computational Linguistics*, (Volume 1: Long Papers):1192–1202, 2016.

L. Song, Y. Zhang, Z. Wang, and D. Gildea. N-ary relation extraction using graph state lstm. 2018.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15: 1929–1958, 2014.

F. M. Suchanek, G. Ifrim, and G. Weikum. Combining linguistic and statistical analysis to extract relations from web documents. *In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.ACM.*, page 712–717, 2006.

K. Swampillai and M. Stevenson. Extracting relations within and across sentences. in proceedings of the conference on recent advances in natural language processing. 2011.

G. B. Tran and M. Alrifai. Indexing and analyzing wikipedia's current events portal, the daily news summaries by the crowd. 2014.

J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. *In Proceedings of ICML*, pages 384–394, 2010a.

J. Turian, L.-A. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, page 384–394, 2010b.

Y. Xu, L. Mou, G. Li, H. P. Yunchuan Chen, , and Z. Jin. Classifying relations via long short term memory networks along shortest dependency paths. in proceedings of conference on empirical methods in natural language processing. 2015.

K. Yoshikawa, S. Riedel, T. Hirao, M. Asahara, and Y. Matsumoto. Coreference based event-argument relation extraction on biomedical text. *Journal of Biomedical Semantics, 2(5):1.*, 2006.