Relational Extraction in the Business domain

Name: **Narayanan Balasubramanian**

Registration Number**: 2003231**

A thesis submitted for the degree of Master of Science in

Artificial Intelligence

Supervisors: **Long, Yunfei**

School of Computer Science and Electronics Engineering

University of Essex

August 2021

Acknowledgements

I would like to thank the following people, without whom I would not have been able to complete this research and without whom I would not have made it through my master's degree.

I would like to acknowledge and give my warmest thanks to my supervisor who made this work possible. It is immense pleasure to express my sincere gratitude to my advisor and supervisor lecturer Long Yunfei for the continuous support in Theis of my Master's degree, and for his patience, motivation, enthusiasm and immense knowledge in natural language processing. His guidance carried me through all stages and allowed my studies to go an extra mile.

I would like to thank University of Essex for having a very good online environment and library structures where I can access the needed resources. The library also provided access to every document, most important articles and journals like IEEE, ACM, databases.

I also thank lecturer Papanastastiou Girogos for his guidance in the group project. His suggestions and motivation helped the group to perform well by achieving high accuracy in supervised learning and detection of coronary heart disease.

Another thanks goes to all of my classmates who helped me virtually in this pandemic situation and made me feel comfortable with the studies.

Finally, I would like to thank my family for always supporting me in every situation. My parents Balasubramanian and Deivanai Balasubramanian for believing in me and for all the love and support, for giving me privilege of good education in such a country.

Abstract

Relational extraction is a popular task in the field of Natural language processing. It is a task of extracting the semantic relationship from the text. This task generally requires two main steps detection and classification. Relation extraction can be regarded as one subtask of information extraction. This task is mainly focused on extracting the related information from an unstructured text data aiming to facilitate the use of data by the applications. In order to extract the information from the text it is essential to pre-process the data and organize it into useful data structure. It can be organized based on the entities and the relation between then.

The relational classification is another subtask which relays on the entities and relationship between them. The relationship occurs between two or more entities of certain types. It may be between a person, organisation, location or anything that could relate to the text. All these related entities fall into definite number semantic categories which leads us to a next step of classification. In this dissertation we are going to follow the method of supervised learning and compare the different types of machine learning models such as naïve baes classifier, support vector machine and decision tree and pretrained models to extract the entities and classify their semantic relation. This approach is mainly based on the feature learning with an existing dataset and common pre-processing tools which are used for natural language processing tasks. The results show that the supervised learning and feature-based solutions can perform better than the unsupervised and heuristic solutions. The Bert model is trained and validated on Semeval 2010 task 8 datasets. The Bert model is tuned on two different strategies like freezing the layers of the Bert and hyper parameters tuning. Both the techniques work well when combined. The hyper parameters which can perform well are identified by experimenting with different parameter settings. The evaluation of this model is validated based on the f1_score and accuracy. The model development, experimentation and evaluation methods are elaborated in this thesis.

Keywords: Relational extraction, Text classification, Natural Language processing, Entity recognition, BERT

Table of Contents

List of Figures

List of Tables

Abbreviation and Acronyms:

NN – Neural network

NLP – Natural language processing

BERT – Bidirectional encoder representation of transformer

LSTM – Long-short-term memory

NE – Named entities

CRF – conditional random fields

TF – Transfer learning

MLP – multi layer perceptron

SVM – Support vector machine

RNN – Recurrent neural network

Bi -LSTM – Bidirectional Long-short term memory

TF -IDF – Term frequency – Inverse Document Frequency

BOW – Bag of Words

ULMFiT – Universal Language model Fine-Tuning

AI – Artificial intelligence

NER – Named entity recognition

ANN – Artificial neural networks

CHAPTER 1 Introduction

Introduction

Relation extraction major task in Natural language processing. It is a combination of two subtasks identification of entities and identification of relations between the entities. It can also be described as the task which automatically extracts the structured data from an unstructured text. The structured data combines two entities and their relation. For example, these entities may be a person, animal, flower, place, location, or a thing, and the attributes which relate to these entities are known as relations. The statistical revolution in information extraction and Natural language processing has led to the process of an infinite number of data and machine learning to extract the relation between the entities. [1] Although there are many data and models, they are addressed through a traditional method like the rule-based method, a different area of studies has produced different solutions and alternative methods.

Relational extraction is a sub-task of information extraction. Relational extraction focuses on the semantic relationship between two named entities. Machine learning and neural networks have been applied to relational classification tasks. In the traditional method, entities were manually marked. They were used to train a deep learning model we need a huge amount of data to bridge this gap researchers developed various techniques for training normally used language or general-purpose language models on millions of unannotated examples. For instance, *Elon musk is the owner of tesla and he lives in America.* The main objective of the relational extraction is to understand the relationship between the elements Elon musk, America, and Tesla.

1.1. Problem statement

In a world of technology, we are facing an enormous amount of data and an organization needs to analyze the data for the growth of the business. This project intends to build a supervised relation extraction model on the business domain dataset. The classifiers take features about the text as input, thus requiring the text to be annotated by other NLP modules such as entity extraction first. The project is addresses challenges like A detailed visualizing chapter of recognized entities/relations will be necessary to communicate obtained information.

1.2. Information extraction

To make a structured dataset there are many steps evolved. Initially, the unstructured text has to be processed and these texts have to be analyzed to form the element and the relation. The process of the next step is to validate the candidates for the entity extraction. Finally, the relationship extraction step evolves to find the relationship between the entities.

Many methods are existing to perform the relational extraction task. Unlike the traditional method, the machine learning models analyze the pattern of the data [2]. Mainly there are two methods feature-based learning and kernel-based learning. Feature-based learning makes the direct extraction of characteristics they are also known as features. The ultimate goal of the kernel-based learning is to develop a system that can automatically generate good kernels for a given circumstance, avoiding the subjective choice of a kernel function by the user.

1.3. Research methodology

The design science methodology provides a framework that perfectly suits to address the proposed research. In a design science project, the researchers iterate over the design cycle which is composed of three phases problem investigation, solution design, and validation.

1.3.1. Problem investigation

The task of relational extraction has been addressed in the form of competitions that aim in evaluating the different methods and approaches to accomplish a specific task. This thesis attempts to overcome some issues found in the previous competitions found in relational extraction. This thesis is inspired by the work of previous papers that has produced a benchmark for relational extraction. Semeval dataset 2010 task 8 has developed their dataset and their validation metrics as well as their training corpora. With this benchmark, this thesis aims to compare several models such as naïve Bayes, support vector machine, and decision tree models to compete with the pre-trained model such as bert.

1.2.2. Solution design

Google AI Research department open-sourced a pretrained model named the BERT model (Bidirectional Encoders representation from Transformers) for Natural language processing. It had a huge impact in the field of Natural language processing because of its performance.

Unlike the other techniques, Bert is a neural network-based technique for pre-training natural language processing which interprets the sentence from both the direction left to right and right to left in an unlabelled data in all layers. With this property, the negative scenarios can also be captured. As a result, the Hyperparameters of the Bert model can be fine-tuned to get the expected result. It conceptually simple model and empirically powerful. The human language Words and phrases can be ambiguous, sentences are often ungrammatical and spelling mistakes are natural. Human language is always inherently noisy and unstructured. Considering all these facts we are going to discuss the possibility to predict the correct relationship between the two entities.

1.3.3 Validation

The last phase is validation, which aims to fulfil the purpose of this thesis. The Semeval dataset has got its own set of validation metrics. Mainly it focuses on the F1 score and accuracy. The goal is to accomplish the task with high metrics, a high f1 score, and accuracy. The validation score of the pre-

trained model is compared with machine learning models to finalize the best. The high performed model can be deployed in the application to get the user experience.

1.4 Contributions

The main contributions of this MSc thesis are:

- Implementation of pretrained model Bert to perform the task of Relational Classification. Evaluating methods extracted relations based on their features mentioned. Results show that feature-based extraction leads to higher precision.
- Fine-tuning models for relational classification methods with SemEval task8 dataset and use of a common set of NLP tools and techniques
- Development and implementation of feature-based learning which can be used for other models
- Evaluating and comparing different methods against the pre-trained model with pre-processing and classification approach which aggregates relational feature. Deployment of the tuned model in an application may result in an extensive set of experiments based on relation extraction.

To summarise, the contribution of this thesis is to change the traditional way of relational classification and study the methods of relational classification using pre-trained models and classifiers of machine learning models using a business domain dataset.

1.5 Thesis structure

The remainder of this thesis is structured as:

- Chapter 2 describes the task of relational extraction and the related topics and techniques. It mainly focuses on the existing work on relational extraction and its limitations. This thesis is motivated based on this in-depth analysis of the state of research.
- Chapter 3 describes the aim and methodology of this thesis conducted and it also explains the contributions fit with one another. It also describes the validation on the dataset of the different machine learning models
- Chapter 4 describes the Results of qualitative study of the proposed model and analyses the reason behind the limitations of the existing methods.
- Chapter 5 summarises the thesis work and contribution and suggests the future work direction of the thesis

CHAPTER 2

Background studies on Relational extraction

This chapter contains the descriptive background and related work for relational extraction. In section 2.1 we go through the related works in the relation extraction and pre-processing analysis operations and then we discuss the relation classification methods.

2.1 Task of Relational Extraction

2.1.1 Formal Definition

Relational extraction is defined as the task of extracting semantic relationships between the elements (Bach and Badaskar,2007). These elements can be either general concepts such as “a person” (PER), a company (COM), a place (PLA), or an instance of such concepts (e.g., “Elon Musk”,” Tesla”) which are known as proper Named Entities (NE). An example of a semantic relationship would be (PER-COM) person founder of the company. These semantic relationships are also called as Relation of the entities.

In this thesis, the elements are named with numbers. Every sentence has got two elements E1 and E2 whereas the elements are made in the tuples along with the relations. Formally the concepts are also called concepts be defined as C. Thus, every class has got their instances E1 and E2.

2.1.2 Named entity recognition

Named entity recognition or named entity identification is a subtask of relational extraction that locates and classify the named entities from an unstructured dataset to a defined set of categories. The phrase Named entity restricts to a word or a phrase. Named entity recognition is the first step in the relational classification process. It deals with the identification of named entities. It can also be defined as the process of identify the specific group of common words which share the semantic characteristics. This system can reach extremely high scores like human performance. There are many techniques such as CRF, rule-based techniques which can bridge the gap between the human efficiency

2.2 Transfer learning

Throughout the history of natural language processing there are many advancements made. The few advancements are like supervised learning methods are advanced in recent years. Some special considerations need to be taken into consideration concerning feature engineering when it comes to NLP. Representing words and sentences as numerical vectors in a Neural Network has certain problems. Commonly, a specific number of words from the training set are chosen called a dictionary that is represented within the model. The dictionary is constructed by all, or a subset of, words in the training

dataset. Any words not contained within the dictionary cannot be used for prediction. The encoding of those words plays a crucial part in the NLP process. Previously, popular techniques for encoding the words included simple representations like Bag of Words (BOW) [3] and Term Frequency–Inverse Document Frequency [4] (Tf-idf). Where Bag of words only sums the number of occurrences of a word within the text, Tf-idf normalizes words over the number of occurrences within the corpora, the concept of being that is property of less common words carry more relative importance. These techniques are simple and work well for some problems. More recent developments in Neural Networks have created the most efficient techniques, such as Embeddings.

The embeddings look at words and try to determine how to convert them into numeric vectors based on the context. The embedding layer represents the word as a vector in a space so that words that occur together are placed close together and unrelated words far apart. Different pairs of words are usually distanced using the Euclidean or Cosine distance. Embeddings are the fundamental tool for the NLP problem but they are still in the development phase. The GLoVE [5] and the Word@Vec [6] are the two examples that were used in the early phase of transfer learning. They were developed by Penington et al and Mikolov et al on a large amount of text and are supposed to work as the encoder of text for general purposes. These embeddings are generally referred to as feature engineering. When the neural network strategies have been discussed the technique of crafting the features should be mentioned. Hand-made features are common in the conditional random field classifiers and other classifiers but recent neural networks have demonstrated the capacity of feature extractors and encoders. There are very less applications that are benefited from the handcrafting features.

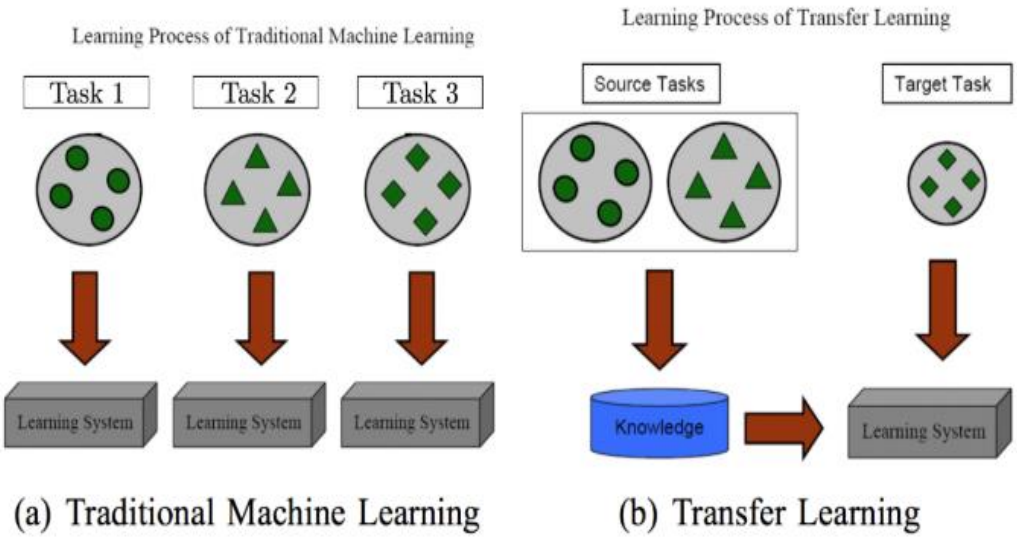


Figure 1 Difference between Traditional method and transfer learning

There were many improvements in Natural language processing in 2018. There were many benchmarks throughout the year [7] [8] [9]. Transfer learning had a very big success with pre-trained models such as Imagenet for computer vision [10]. These models attempt to create a general understanding of visual objects sometimes with various classes. These models are fine-tuned to generalize to specific applications. With unique pre-training, the strategy is that we could fine-tune any model based on the small dataset of domain-specific data and get a very good result. This pretraining is also known as extensive dictionary learning [11]. The technique is that we decode the signal as a linear combination of the other simpler functions. We can also try this method such as learning the basis for a problem and modify the result accordingly.

Elmo model was the first wave for transfer learning which was developed by peters et al. the general idea of Elmo was to pretraining using the large corpus and to create a model using the text. This pretraining results in the efficient use of unstructured data with semi-supervised learning techniques but not having the annotated datasets. The task was carried away by the general structures in the language by the next word prediction by the given observed sequence. Elmo works as a feature extractor bypassing the sequence through a bidirectional LSTM. This feature captures the patterns generally similar to the lower-level convolution layer capturing the lower-level image feature. In Elmo, each embedding is conditioned on its context. That is the same token or word can have different embeddings. The ELMo is not fine-tuned to a specific task. It is used as a technique for feature extraction. [12]

Another development was made with the results of Elmo named ULMFiT. This was developed by Howard et al. [11] It is a deep lstm based model used to improve feature extraction. It uses several techniques that converge during fine-tuning. AWD-LSTMs were proposed by Merity et al [13]. In this lstm triangular learning rates and learning rate schedules are used to avoid catastrophic forgetting. It does not just apply dropout to recurrent state h but allows for each gate as well. These tactics push the transfer learning to the next level. Previously, the attempts to fine-tune were tricky and proven to be having poor convergence and catastrophic forgetting. These problems have been solved by these methods. ULMFiT is hard to train and it is sensitive to hyperparameter tuning.

2.3 Transformers

Transformers were proposed as an alternative to lstm and other RNN methods by Vaswani et al [14]. It was published in the paper Attention is all you need. It is the first model to relay upon attention mechanism to compute the representation of its input and output without using the sequence of aligned Recurrent neural network or a convolution. The core technique in transformers is “self-attention” mechanism combined with feed forward neural network. Because of its efficiency it has become a tool for several tasks in NLP and time series prediction problems. Unlike RNN transformers are not recurrent, it uses token pair wise scoring. The algorithm contains the ability to determine or prioritize the words in a sequence. For example, if a sentence is like “Sundar is the CEO of Google and he live in

America". For people to understand he refers to Sundar in the sentence, but the machine learning models had a problem in understanding this and even though LSTMs parsers the sequence left to right it understands partially and they are influenced by the nearby word. Self-attention technique does the other way around by sequentially updating the hidden state at each time instead of looking the entire sentence.

Figure 2 shows a representation of the model. The architecture is based on the idea of Autoencoders, a type of unsupervised learning technique developed by Ballard et al. [15]. Encoder14 decoder stack is used by Autoencoder. In the autoencoder encoder creates an improved representation of original inputs where decoder reconstructs it. Transformers use the similar kind of technique with additional features like sharing the information between the encoder and decoder stacks.

The Self-Attention matrix calculation has very few components to be calculated such as Query, key and value matrices. We can calculate by embedding the matrix X and multiplying them with the weight matrices.

Equation of Query $Q = X \times W_Q$

Equation of Key $K = X \times W_K$

Equation of Value Matrix $V = X \times W_V$

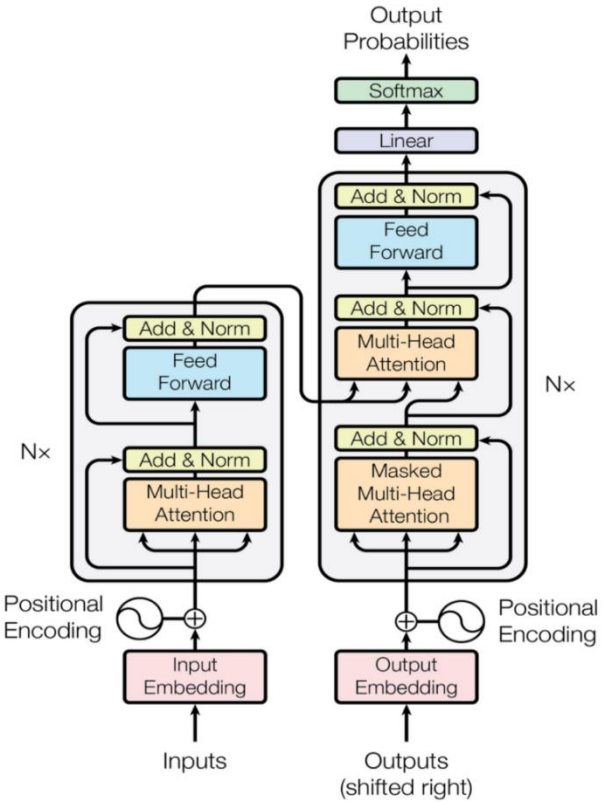


Figure 2 Transformer Architecture

2.4 BERT

Bidirectional encoder representation from transformers (Bert) is a most popular transformer-based machine learning for language modelling published by google which was introduced by Jacob Devlin and his colleagues from google in the year 2018 which resulted in better search. The English language has two models namely Bert Base and Bert Large. The difference between these two models are only the encoders and self-attention heads. The both models are trained with unstructured data from book corpus which contains of more than 800 million words and they were also trained with English Wikipedia which comprises of 2500 million words.[19] The Analysis behind the Bert states that the understanding tasks are not well understood. But current research and analysis results in the statement that the output is mainly based on the chosen inputs.

The Bert Base model comprises of 12 stack Encoders with 12 self-attention layers whereas the Bert large encoders comprise of 24 stack encoders with 16 self-attention layers. Both the model is designed with bidirectional self-attention heads. These models read the entire sentence sequentially either from left to right or right to left. The input sequence is a sequence of tokens which are embedded into vectors as the transformers do and the processed in the neural network. Each of the vector will correspond to the input token with the particular index. Model and size also matter in the Bert model that is the Bert large architecture has 345 million parameters and Bert base architecture has 110 million parameters [16]. With enough training and tuning we can achieve high accuracy with this pretrained model. Bert can be used for various kinds of tasks such as Classification task, Question answering task and named entity recognition task. The most important task in the Bert was Masked language modelling. In this task the whole sentence is given to predict the possible word to fit in the sentence. To ensure that the model does not predict the wrong word there are few noisy sentences in the data so that the Bert can analyse the correct prediction. Another important task is Next sentence prediction. The Bert learns from the pairs of sentences as the input and then the prediction is made for the following sentence in the test dataset.

There is special kind of input that is required for this architecture. To distinguish between the two sentences the input has to be processed in certain pattern starting with [CLS] token and [SEP] token has to be inserted at the end of sentences. This pattern helps the Bert to understand the start and the end of the sequences. Positional embeddings are also added with the sequence for more precision of the sequence embeddings.

2.5 Naïve bayes classifier

The Naïve bayes classifier works on the principle of bayes theorem which can be extremely fast when is comes to classification problem. It predicts the classes of the unknown dataset based on the Bayes theorem. In this thesis we have brought the Naïve bayes classifier in action. The assumption of the naïve bayes classifier is independent among the predictors. For instance, if the fruit may be classified as an

orange if the features are orange, round in shape and 3 inches diameter. Even these features may depend upon other features, all these features may contribute to the probability for the classification. So, this is the reason it is known as ‘naïve’. This theorem works on the posterior probability of class and target.

$$P\left(\frac{c}{x}\right) = (P\left(\frac{x}{c}\right) * P(c))/P(X)$$

P(c) is the probability of class

P(x/c) is the probability of predictor given class

P(x) is probability of predictor

This classification is easy to perform and it has high performance when there is less amount of data. It performs well in case of categorical input than numerical inputs. It is also known as bad estimator. In real time scenarios we don’t get the same data independent with their features. These are the limitations of naïve bayes.

2.6 XlNet

There are many pretrained models for text classification. Few of them are XLNet, ERNIE, Text-to-Text Transfer Transformer, Binary Partitioning Transformer. [17] XLNet, Google's most recent model, achieved State-of-the-Art (SOTA) performance on major NLP tasks like Text Classification, Sentiment Analysis, Question Answering, and Natural Language Inference, as well as the crucial GLUE benchmark for English. It outperformed BERT and has already established itself as the model to beat for complex NLP tasks as well as text categorization. The following are the main concepts that is behind XLNet are Language Understanding Using Generalized Autoregressive Pretraining and Transformer-XL. Though BERT's autoencoder took care of this, it had additional drawbacks, such as believing there was no association between the masked words. During the pre-training phase, XLNet suggests an approach called Permutation Language Modelling to counteract this. Permutations are used in this technique to create information in both the forward and backward directions at the same time. Transformer xl is used in XLNet. Transformers, as we know, were an alternative to recurrent neural networks (RNN) in that they permitted non-adjacent tokens to be processed simultaneously. This resulted in a better comprehension of textual long-distance relationships. Transformer-XL is a modified version of the transformer used in BERT that includes two additional components. First component is a recurrence at certain segments between two sequences that provides context. Second component is a relative positional embedding that gives information on how similar two tokens are. [17]

2.7 Related Works

Wenrui Xie from Beijing university has published an IEEE journal in 2021 [18]on the topic A entity attention-based model for entity relational classification for Chinese literature text. This paper is mainly based on the Chinese language but the methodology followed in this journal is quiet amazing. This paper has been the motivation for this thesis It helps the model to classify the text in most reasonable manner. In this paper the results exhibit the entity attention-based model outperforms the state of art methods in Chinese literature. EA Bert model is used in this paper for feature extraction and Bert model for the classification. analyses the characteristics of Chinese literary works and the deficiencies of the existing standard models. It examines the peculiarities of Chinese literary works as well as the shortcomings of current conventional paradigms. EA-BERT is proposed on this premise, and then the overall structure and various modules of EA-BERT are described in depth, followed by comparison tests with mainstream models and the removal test, which demonstrated the effectiveness and progress of EA-BERT. [18]

Relational classification with the application to terrorist profiling [19]was published by the Jian Xu and his colleagues from united states of America. In this work they have implemented the Random Forest classifier and statistical learning methods. This Random Forest model outperforms the fuzzy clustering's and ordinary decision tree models. This works demonstrates that the number of attributes is big and the amount of data available is limited, random forest is known to perform well. They have compared the performance of the random forest technique to that of ordinary decision trees and fuzzy clustering on a synthetic terrorist dataset.

There is another excellent work published by Ya zhang on the topic of Improving the Relational classification with multi graph. The relation entity graph is framed to learn the structural feature through a GCN model. The Bi-Lstm model is used to obtain the information from the text and the information is added to the graph. This paper was most recently published in the IEE portal. They have conducted experiments on SemEval 2010 task 8 datasets. And the performance is compared to the latest model of m=MGGCN without adding any additional information. The Semantic features which are based on the Lstm architecture is combined with the feature based two different GCN structures. The study demonstrates that Length of the sentence is directly proportional to the parsing of the sentence. We may need to add more information or use a more complex model structure to acquire more useful information from the parse tree. We can get more useful information from the relation-entity graph by looking at the correlation between entity pairs and the relationship between other related entities and entity pairs.

CHAPTER 3

Methodology

3.1 Relational classification

As there are many models for transfer learning the Bert model has to be fetched from google pretrained model sources. The list of models which were released at the time of writing this thesis are given below

1. Bert base model - Uncased
2. Bert large model-Uncased
3. Bert Base model - cased
4. Bert large model-cased

The lower case is considered as the best practice for the Bert model. In this thesis we are going to discuss based on the Uncased Bert base model. When the Bert was released, it was huge and large with many investigations the open AI and Microsoft improvements the Bert is optimized. This improvement requires less expensive configuration of computer hardware for fine tuning.

3.2 Data

Semeval 2010 task 8 dataset is a multiway classification dataset comprises of semantic relations between nominals. The task was developed to compare different approaches to the task of semantic relational classification. This dataset is pre-processed and developed by Natural language processing techniques with the help for the natural language tool kit libraries and textblob libraries. The semeval dataset is represented using the word vector model and it is published as a numpy array. This dataset contains different det of frozen numpy array.

Few of the sentences are derived from semeval -1 task 4 (classification of semantic relations between the nominals) and remaining were obtained from web especially for this task. This dataset contains two separated dataset of original English language. It is separated as train dataset and test dataset. Train dataset comprises of 8000 sentences marked with pairs of elements and classes.

Datasets	Total
Training sentences	8000
Test sentences	2717

Table 1 Semeval 2010 task 8 datasets

In total there are 9 relations for semeval -2 task 8. The data format is given below with the examples

Relations	Description
Cause-effect	An object leads to an effect. Example: corona was caused by covid virus
Instrument-Agency	A technician uses an instrument. Example: telephone operator
Product-producer	A creator causes a product to exist. Example: a factory manufactures cars.
Content-container	They saw that the <e1>tool</e1> was put inside <e2>cupboard</e2>, which looked tidier."
Entity - origin	A child is coming or is derived from a parent (e.g., position or material). Example: parcel from foreign countries
Entity-Destination	An object is moving towards a destination. Example: the kid went to school
Component-Whole	An entity is a component of a larger member. Example: my house has a large bedroom
Member -collection	Individual member forms a non-functional part of a collection. Example: there are many plants in the garden
Message -Topic	A written or spoken message about a topic. Example: the lecture was about history

Table 2 Example for relations in semeval 2010 task8 dataset

3.2.1 Data visualization

Data has to be visualized before any process is processed. We have to confirm whether the data is balanced or imbalanced. Balanced dataset is the foremost cost that has to made ready for a classifier to perform well. The imbalance dataset may cause the confusion for the dataset. For instance, if the majority of the class dominates the minority dataset, then the classifier may opt for the majority class and still get the high accuracy and prediction may go wrong. To avoid this the classes in the data has to be balanced. That is when the train data and test data are split the classes and sentences based on the classes has to be stratified. Otherwise, there are many other techniques which can balance the data. In semeval data set there is two different splits for training and testing. When compared the train and the test has got the equal percentage of each class. There is no need for data balancing but to ensure the classifier does not classify the wrong prediction the data has to be visualized

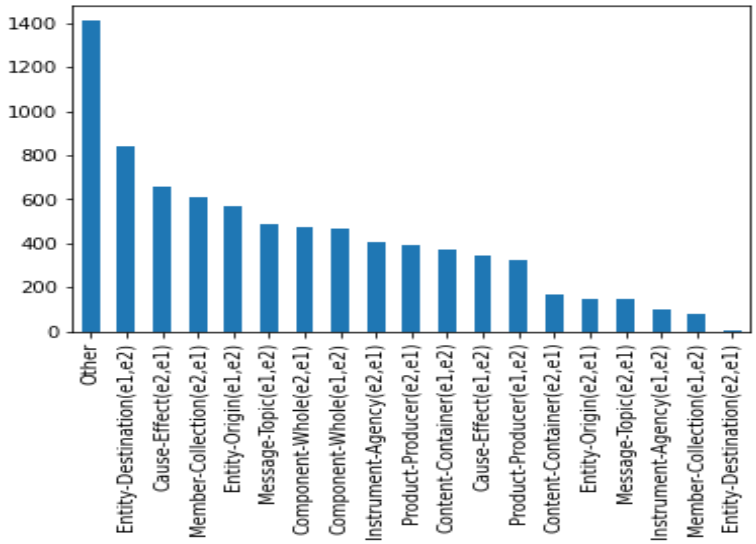


Figure 3 Semeval 2010 task 8 train dataset visualization

Then based on the relation and the position of the entities the number of sentences from the test and train dataset are combined together count of the sentences are measured. This step is visualised because the impact of the improper and imbalanced data can be always negative in the results. The pie chart demonstrates the percentage of every label from the semeval 2010 task 8 datasets. The e1 and e2 are differentiated into different class based on their position in the sentence. If the e1 is after the e2 even though it has same pair of relation because of the position of the entities the class is sub divided as e2-e1

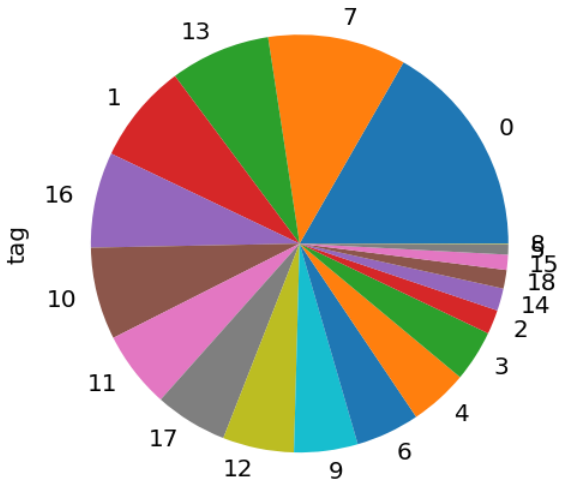


Figure 4 One hot encoded labels

Relations	Total	(E1, E2)	(E2, E1)
Cause-effect	1331	478	853
Instrument-Agency	660	119	541
Product-producer	948	431	517
Content-container	732	527	205
Entity - origin	974	779	195
Entity-Destination	1137	1135	2
Component-Whole	1253	632	621
Member -collection	923	110	813
Message -Topic	895	700	195
other	1864		

Table 3 Data visualization of task 8 dataset

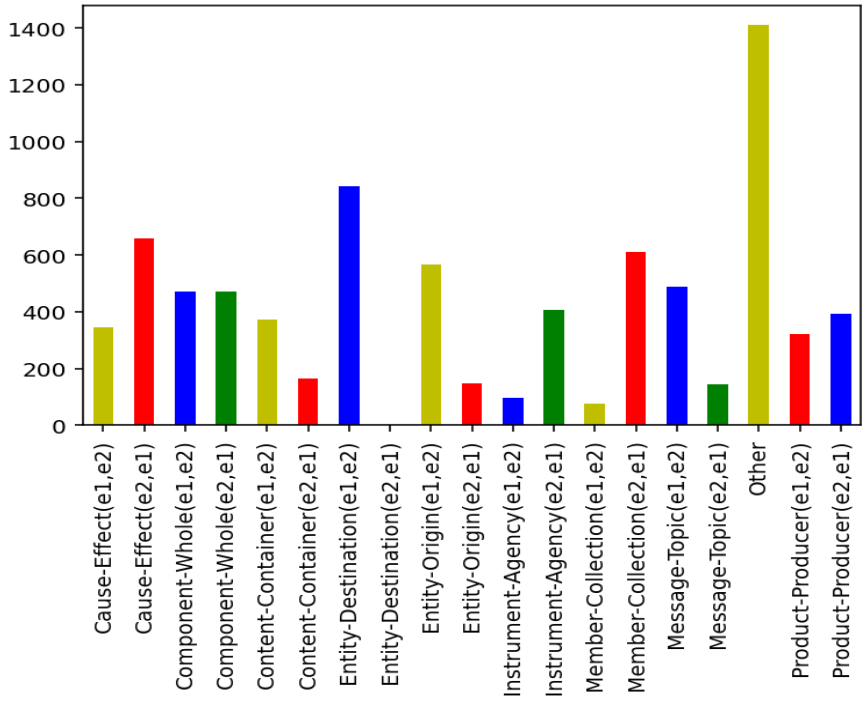


Figure 5 Semeval 2010 task 8 classes visualization

3.3 Pre-processing

Getting started with pre-processing, this step depends on the task and the volume of the dataset. In this thesis we are going to discuss about the pre-processed step that was accomplished to attain high accuracy. Pre-processing is an essential step required for every dataset to acquire high accuracy and precision. This phase enables the valid data and removes the chunk from the dataset. Especially the natural language has different versions like case sensitive scenarios, punctuation scenarios and many other unnoticeable errors. Every pretrained model has their own set rules. Mainly Bert has own set of

rules that has to be considered to identify the sentences start and end position. The following steps are followed as a part of pre-processing.

- Text Normalization
- Lower case the text
- Punctuation removal
- Blank space removal
- Stop words removal
- Lemmatization
- Tokenization

3.3.1 Text Normalization

Text normalization is the process of cleaning the text into a readable form by removing the noise. Text normalization includes converting every number from the data set to words or removing the numbers completely from the dataset. Normalization also includes the process of converting the text into lower or upper case. By lowering the case we can maintain simplicity and it will help us to maintain the flow during the Natural language processing tasks. It makes the process quite forward. All the punctuations, expressions and unwanted symbols will be removed in this step. All the white spaces will be removed including the stop word and typos. Stop words are nothing but the commonly used words in the language. In our case we are going to use English. For example, a, the, is, are these are the few stop words. They are undeniable but they do not produce any meaning to the sentence thus all these words are removed. If these are not removed, they will not bring much information to the model. Thus, the stop words are removed

3.3.2 Lemmatization

The stemming is similar to lemmatization. Stemming is typically easy but it is proceeded without any knowledge of the word. Unlike stemming lemmatization intends to identify the part of speech and extracts the exact meaning of the sentence. It is also a step-in text pre-processing. This step involves in the process of grouping the words together in their inflected forms. By this approach each word can be addressed in their inflected forms. These words are known as lemma or the words in the dictionary form. To illustrate the example of lemmatization drinking, drink, drunk conveys the action of drink all these words will be lemmatized to base word drink. The association of the base word is called as lexeme. For example, good is the lemma for the word better. This link is not achieved by the stemming process. For another example, meeting can be addressed as a verb meet or action of meeting with people. This may completely depend on the context. All these words can be addressed by lemmatization. Simple way to proceed with this step is to look up dictionary. Sometimes rule based system is required to perform this action.

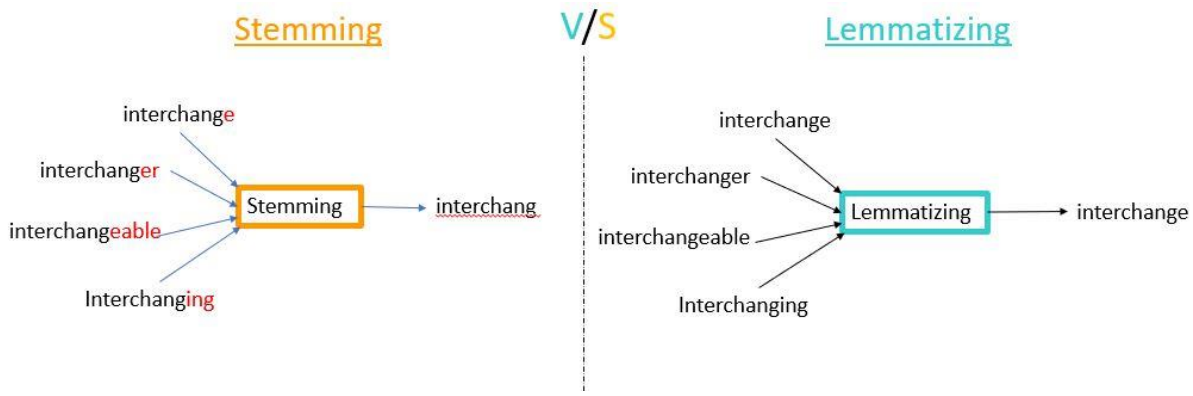


Figure 1 Difference between stemming and lemmatizing

3.3.3 Punctuation and blank space removal

Punctuation is the utmost factor when it comes to writing. Punctuation is used to express the pause, emotion and the stop of every sentence which are conveyed to the reader, when it comes to analyzation these are just symbols that does not bring much of information to the model. When the sentence is loaded with enough of punctuation and loaded to a model. The model considers every symbol as the word and during the classification the information gain percentage will be much less. Thus, is it necessary to remove the punctuation. In this thesis we have considered this as a main part of pre-processing step and all punctuations are converted to blank spaces. All the blank spaces are removed from the context. The reason behind this step is that when the sentence is tokenized as a word all the empty and blank spaces will be considered. This will also make a huge impact in our modelling and precision will be fluctuated. So, it is better to remove the blank and empty spaces

3.3.4 Tokenization

Tokenization is a most important part of Natural language processing techniques such as normalization and cleaning process. Tokenization is a process of splitting or breaking the sentence from a paragraph or a word from the sentence. The main aim of this process is to split the sequence. This process is also known as tokenizing and the individual parts which are separated by tokenizing is known as tokens. There are two types of tokenizing separating every individual sentence from a paragraph is known as sentence tokenization and splitting each and every word from a sentence to individual tokens is known as word tokenization.

Sentence	This is an amazing place to stay and warm up.
Tokens	“This”,”is”,”an”,”amazing”,”place”,”to”,”stay”,”and”,”warm”,”up”

Table 4 Tokenization technique

3.3.5 Padding

Padding is not a part of cleaning process but it is useful when it comes to processing the text. The reason behind this step is there are many data and text available on internet and sources. But text is varied

according to the context. The length of the text may vary for every instances. Sometimes our classifier may also consider the length of the sentence as a parameter. To make the unstructured data into a unique set of patterns we have to frame a set of instruction for our classifier. Padding helps us to align the text to the same length. The longer text may have the information at the end of sentence and some text may shorter length. Classifier will not get the exact number of tokens. To solve this issue the padding sequence will create a dummy padding which may inform the classifier that it doesn't contain the information that is required for consideration

3.4 Algorithm

3.4.1 How Bert works?

It is very essential to know about the working architecture of Bert before using it for classification. As mention before there are many types of Bert exists, we are going to discuss the Bert uncased base model for classification. The Bert base model has 12 layers and 12 attention heads which includes 110 million parameters. All these parameters can be used for fine Tuning. To compare with the OpenAI GPT model Bert base also have the same size. All transformer layers in the Bert are Encoder blocks. The model design has three main parts: pre-processing, pre training and classifier. However, google did not publish the exact method for pre-processing and how well the code is adapted for the classification some assumptions are made in order to complete the task.

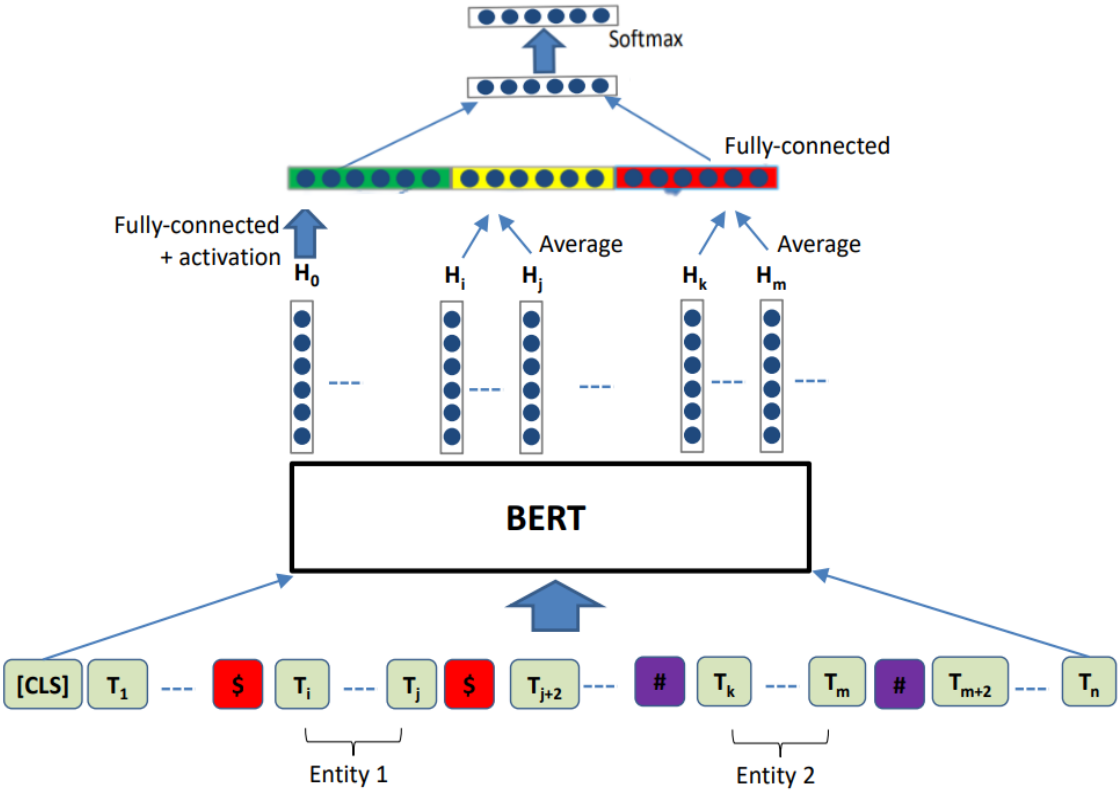


Figure 2 Bert Architecture

3.4.2 Designed Text Pre-processing for Bert

To initiate a text Bert model, have certain rules and regulations for the input text. These creative rules have made the model for great use. To start with the embeddings for Bert there are three embeddings.

Position Embeddings: Bert has the capability to learn the position of the words and use the positional embeddings from the sentence. These are the additional feature to the transformers. It also captures the sequence and the order of the information

Segment Embeddings: In segment embeddings the Bert model takes the sentence pair as the input for the tasks. It also helps the model to distinguish between the first and second sentence

Token Embeddings: Token embeddings are nothing but the embeddings which are learned from a specific token from the token vocabulary.

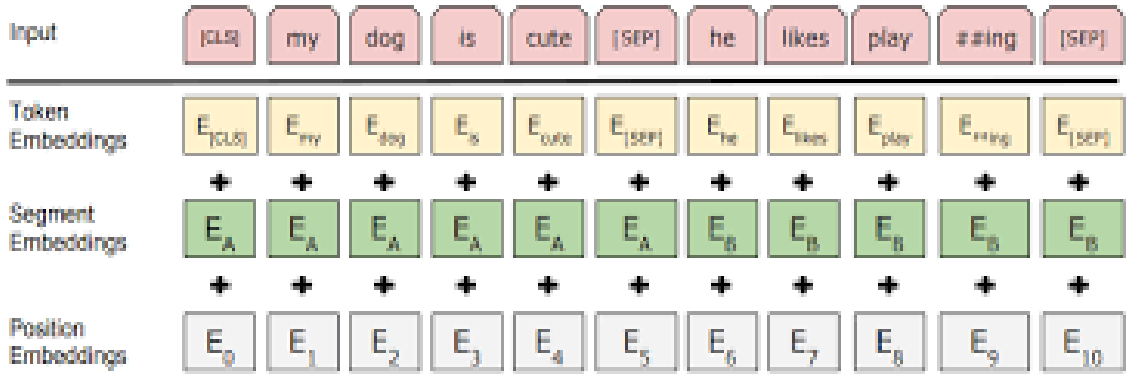


Figure 3 Input for Bert

3.4.3 Pre-training

To start with the input structure, we have to analyse the task. In this task of relational classification with semeval dataset there are sentences marked with entities and the targeted class. Considering the two relational entities within the sentence and the target we can mathematically express the following statement as follows:

$$\tilde{r}^i = \left(\tilde{x}^i, e_1^i, e_2^i\right)$$

Table 5 Targeted input

In this equation the r is the targeted variable, x is the tokenized sentence and e1 and e2 are the entities within the sequence. The i in the equation is iterative for each and every sentence. If the r in the sentences represent the entity pair, then the entity e1 and e1 must be having the relation as r. To train the algorithm this type of input is given along with the targeted variable.

3.5 Experiment

The training is focused on the Bert model proposed in this paper. It can be divided into two stages. Firstly, the pretrained model from google is adapted. Secondly the parameters tuning and adding layers to the Bert model The exact parameters used in Bert base model are not specified by devlin et al but it can be taken from the results. The experiment is followed full comparison between the different setting of the Bert model. They are validated with both test and validation dataset. By understanding the different settings and then the behaviour of the model the hyper parameters are fine tuned. There is list of parameters chosen for the tuning. Total there are many combinations and few of them are listed below with the variation of the learning rates and the batch size.

In terms of optimization there are many tricks and techniques with weights. The best performance of the model is typically achieved by using the representation not on the top of the layer. Pretrained representation can be used as features. Secondly the added layers can be trained. When it comes to pretraining the weights we can adapt the pretrained model during the adaptation phase. That be proceeded by training each layer by layer to adapt to new task. Lowering the learning rates is performed to avoid the overwriting the information. In general more the parameter is fine tuned the model will slower your training.

Parameters	**Value**
Batch size	64, 32, 16, 8
Learning rate	5 e^-5, 2e^-4, 1e^-4
Epochs	5,4,3,2
Dropout	0.3,0.2,0.1

Table 6 Fine Tuning parameters

These are the few parameters which were recommended for fine tuning. There are many other parameters in the architecture. The batch size is experimented with the multiples of 8 because the TPU recommends the batch size of 8 or its multiples. Especially the larger the batch size it is easy for the model to converge. In addition to that the experimentation resulted in the smaller learning rate. This is because the larger the batch size the small learning rate can make sure that no problems are faced during the convergence.

In addition to these fine-tuning parameters one step ahead was performed. The layers were freeze in the different stages. The layers of the pre trained model can be frozen completely or partially. The experimentation was done on three levels. The unfrozen and the completely frozen layers did not result in producing high precision. On the other observation in the 12 layers of Bert exact half of the layers that is 6 layers were frozen on the pretrained model and remaining 6 layers were trained with this dataset.

3.6 Validation

The validation is continued by the Validation dataset which is unseen dataset for the model. Validation dataset is derived dataset from the semeval task 8 datasets. The semeval 2010 task 8 datasets consist of unique techniques of validation methods. Mainly it is focused on the F1_score for the tagged entities which are classified on the 18 classes and left the other class. One of each label are calculated in the standard procedure and then the macro average is considered. The other will be counted as the true negative or the false positive from the batch. This thesis will validate each sentence with the classified target. These individual scores are then averaged according to the maximum strategy stated by argmax. Maximum averaging treats the entity of all types equally by calculating the f1 score individually and the averaging them to the maximum. There is another type of validation method known as minimum averaging in this method it is contrast to the macro averaging techniques. It can be sensitive to the few classes. It treats all the entities equally it computes the scores and at last it averages them. The f1 score can be mathematically represented as follows:

$$f1\ score = 2\ .\frac{precision.recall}{precision + recall}$$

$$precision = \frac{True\ positives}{True\ Positives + False\ Positives}$$

$$recall = \frac{True\ positives}{True\ positives + False\ Negatives}$$

CHAPTER 4

Results and Discussion

4.1 Results

Results of Accuracy, F1_score, precision, recall is all macro averaged from the random set from the semeval task8 dataset. We can see that the variation in each epoch the model f1_score is increasing as it is trained on the dataset. The result takes the account of true and the false positives. It is generally considered as the balanced measure. The Bert model is trained and validated with f1 score and to be precise it works more efficiently than the normal machine learning classifiers. The Naïve bayes classifier and the Support vector machine classifier is tuned and experimented with this dataset. These two classifiers are not up to the expectation level. The machine learning classifiers are validated based on the accuracy of the prediction. That is the predicted labels are compared with the original test labels and then the results are calculated. The naïve bayes classifier produced the 60 percent accuracy after fine tuning the dataset. The support vector machine is a good classifier but when compared to the pre trained model it is not much efficient. Additionally, the best results on both test and validation datasets are generated by the pre trained models.

Comparing the same hyperparameter setting on validation dataset shows that Bert model outperforms the same parameters as machine learning models. Table 8 explains the different setting of the Bert model and their parameters. The batch size is increased in the multiple of 8 and the learning rate maintained as low as possible to get the expected results. To reduce the loss the pre trained model can be added with layers and the dataset can be trained on each of the layer. But this technique may reduce the performance speed of the model. It is compared with naïve bayes and svm model.

Model	F1_score
Bert	79 %
Naïve bayes	52 %
Support vector machine	54 %

Table 7 Comparison between the models on semeval dataset

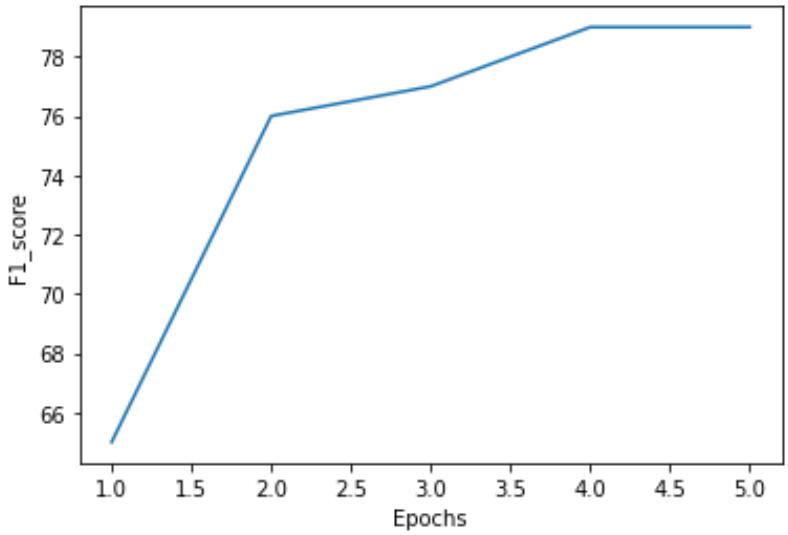


Figure 4 Bert model F1_score

4.2 General discussion

As seen the pre trained model Bert outperforms the machine learning classifiers such as support vector machine and the naïve bayes classifier. This is what is expected to be when the unknown dataset f1_score is higher when the model is trained. The overall conclusions are that the proposed model configurations are working perfectly when compare to the other classifiers. One of the challenging processes in this method is applying the text dataset pre-processing for the unstructured data and categorizing the information. The Relational extraction process seems to be typical but it is most interesting part of natural language processing. The data pre-processing has to be concentrated more because the text will be vectorized for the classifier. To make the exact prediction of the classes the steps for the normalization has to be followed. Sometimes more clean data may also cause the classifier to be less efficient. This thesis may not produce the result established by Devlin et al. Since the splitting and finetuning, pre-processing was not mention for the pretrained model. There are also may other factors even though we have tried to fine tune the Bert Base uncased model to maximum limit and could be extended in the future.

Batch size -	Learning rate	dropout	epoch	F1_score
6	1e^-4	0.1	2	62
8	1e^-4	0.2	3	64
16	1e^-4	0.1	4	67
32	1e^-4	0.1	5	68
32	1e^-4	0.1	5	69
64	2e^-4	0.3	4	70
64	2e^-4	0.1	5	79

Table 8 Experimental setup and Results

The naïve bayes and other machine learning classifier are good when it comes to classification problems but it is not as precise as the other models for textual inputs. Because the vectorization and other process has to be trained on a very large knowledge base which is quite costlier and time taking process. In terms of text classification, the Train and validation losses are taken into account. More than concentrating in the accuracy the loss can be minimized to get the good results. With the available resources the models are trained as good as possible to get maximum results.

The Figure 11 represents the curve of the train loss of Bert model. Initially the Bert has a huge train loss after different experimental setup and parameter tuning the train loss is reduced as much as possible. It shows that the curve is travelling towards the minimum point. At the end of the fifth epoch the train loss is minimized less than 15 percentage and the f1_score is raising to 79%.

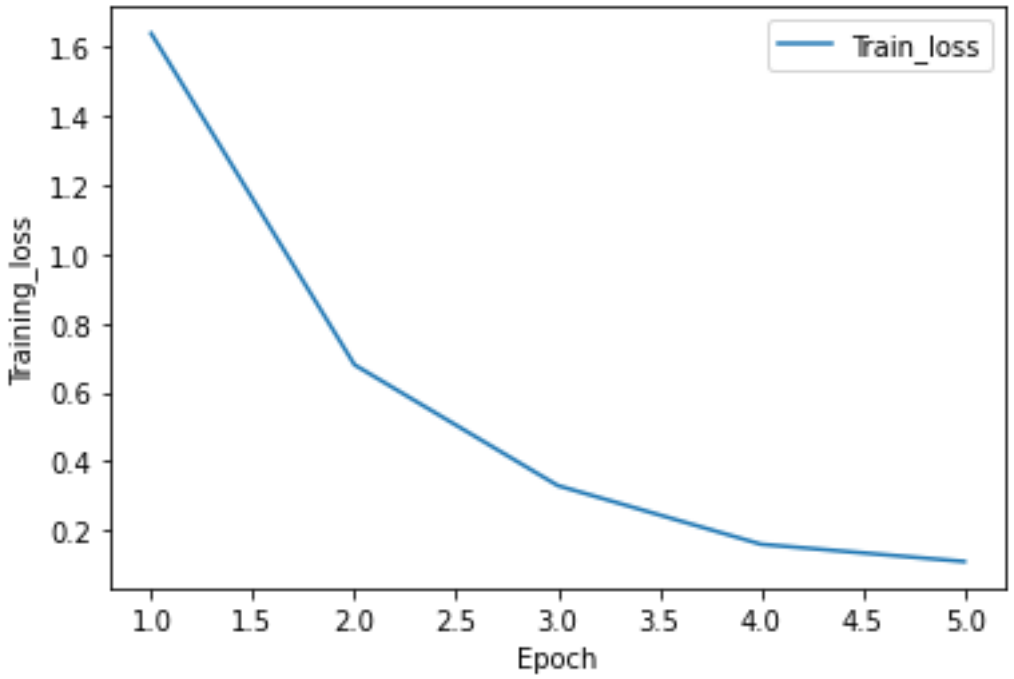


Figure 5 Bert base uncased Training loss vs epoch

CHAPTER 5 Conclusion and Future Works

Conclusion

5.1 Conclusion

The overall conclusion is that the proposed model works well in its configurations. With the available resource and intensive research, the Bert model is trained to classify the extracted entities and relations. The Relational extraction and classification task can be performed in many other ways using many techniques. This thesis demonstrates one of the techniques for relational classification which can work with high precision. In addition to that the results in this thesis discuss about the comparison of the pretrained models and the machine learning models for the task of relational classification. As discussed above there are several other reasons, evaluation methods, designs and tunings that could variate the results. These may cause the fluctuations and could not reproduce the original results. It concludes the use of bidirectional encoders representation of transformers are more efficient than any other machine learning classifiers.

5.2 Future works

A large number of corporate information is existing in the form of Textual data information which are mostly unstructured and the key challenges are to find their entities and relations. This research can be further developed for the information retrieval in a document of unstructured data in a large collection of documents. Every single entity and relations can be captured for the user and recommender can suggest the recommendations according to their needs. Content based recommender system may have a greater benefit using this model. A user profile can be learned using the relational method and all the entities will have their own relation with the other entities. This thesis can be approached in different methods towards identifying different opinions and subject in the text. With the rapid technology development this method can be developed in many ways particularly in text format and text classification. The terms such as parts of speech, opinion words and phrases can be considered as features for the classification techniques. Application of relational learning are too developing in business information analysis, content and Web mining, and random other areas, such as the investigation of melodic exhibitions. Not only in the business domain the bi medical fields will also have a huge response for the relational classification models.

References

[1] [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing.

[2] [Online]. Available: https://link.springer.com/article/10.1007/s42979-021-00592-x.

[3] Z. C. a. D. G. B. Michael McTear, “The Conversational Interface: Talking to smart devices. Vol. 1. Springer International Publishing, 2016.”.

[4] K. S. Jones, ““A statistical interpretation of term specificity and its application in retrieval”. In: Journal of Documentation 28 (1972), pp. 11–21.”.

[5] R. S. a. C. M. Jeffrey Pennington, ““Glove:Global vectors for word representation”. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).2014, pp. 1532–1543.”.

[6] T. M. e. al., ““Efficient estimation of word representations in vector space”. In: Proceedings of Workshop at ICLR (Jan. 2013).”.

[7] J. H. a. S. Ruder, ““Universal Language Model Finetuning for Text Classification”. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Jan. 2018, pp. 328–339. arXiv: 1801.06146. url: https: //arxiv.org/abs/1801”.

[8] M. P. e. al, ““Deep contextualized word representations”. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018, pp. 2227–2237. url: http://allenn”.

[9] J. D. e. al, ““BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: (2018). issn: 0140-525X. doi: arXiv:1811.03600v2. arXiv: 1810.04805.”.

[10] T. P. a. M. E. Ron Rubinstein, ““Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model”. In: IEEE Transactions on Signal Processing 61.3 (2013), pp. 661–677.”.

[11] J. a. D. a. S. R. a. L. L.-J. a. L. K. a. F.-F. L. Deng, “ “ImageNet: A large-scale hierarchical image database”. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE. 2009, pp. 248–255. isbn: 978-1-4244-3992-8. doi: 10 .1109/CVPRW.2009.5206848.”.

[12] J. L. e. al, ““A Survey on Deep Learning for Named Entity Recognition”. In: CoRR XX (2018), p. 1. arXiv: 1812.09449v1”.

[13] N. S. a. R. S. Stephen Merity, ““Regularizing and Optimizing LSTM Language Models”. In: (Aug. 2017). arXiv: 1708.02182. url: http://arxiv.org/abs/1708.02182.”.

[14] A. V. e. al, ““Attention Is All You Need”. In: Advances in neural information processing systems. 2017, pp. 5998–6008. isbn: 1469- 8714. doi: 10.1017/S0952523813000308. arXiv: 1706.03762.”.

[15] [Online]. Available: https://en.wikipedia.org/wiki/BERT_(language_model).

[16] [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

[17] [Online]. Available: https://www.analyticsvidhya.com/blog/2020/03/6-pretrained-models-text-classification/.

[18] W. Xie, *A Entity Attention-based model for Entity Relation Classification for Chinese Literature Text,* 2021.

[19] J. C. L. Jian Xu, "Random Forest for Relational Classification with Application to Terrorist Profiling".

[20] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.In: Psychological Review 65.6 (1958), p. 386. issn: 0033295X. doi: 10 . 1037 / h0042519. arXiv: arXiv:1112.6209".

[21] S. Dreyfus, ""The numerical solution of variational problems". In: Journal of Mathematical Analysis and Applications 5.1 (1962), pp. 30– 45. issn: 0022-247X. doi: https://doi.org/10.1016/0022- 247X(62) 90004 - 5. url: http : / / www . sciencedirect".

[22] G. E. H. a. R. J. W. David E. Rumelhart, ""Learning representations by back-propagating errors". In: Cognitive modeling 5.3 (1988), p. 1. issn: 00280836. doi: 10.1038/323533a0. arXiv: arXiv:1011.1669v3".

[23] [Online]. Available: https://images.app.goo.gl/yw8PJts11UPmvA1f7.

[24] S. H. a. J. Schmidhuber, ""Long Short-Term Memory". In: Neural Computation 9.8 (1997), pp. 1735–1780. issn: 08997667. doi: 10.1162/neco.1997.9.8.1735. arXiv: 1206.2944".

[25] J. S. a. F. C. Felix A. Gers, ""Learning to forget: Continual prediction with LSTM". In: Proc ICANN'99 Int. Conf. on Artifical Neural Networks 2 (1999), pp. 850–855. issn: 08997667. doi: 10.1162/089976600300015015. arXiv: arXiv:1011. 1669v3.".

[26] K. G. e. al, ""LSTM: A Search Space Odyssey". In: IEEE Transactions on Neural Networks and Learning Systems (2017). issn: 21622388. doi: 10.1109/TNNLS.2016.2582924. arXiv: 1503.04069.".

[27] [Online]. Available: https://www.pluralsight.com/guides/importance-of-text-pre-processing.

[28] [Online]. Available: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235.

[29] [Online]. Available: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/.

[30] [Online]. Available: https://towardsdatascience.com/transfer-learning-in-nlp-fecc59f546e4..

[31] [Online]. Available: https://towardsdatascience.com/transformers-89034557de14.

[32] S. N. K. Z. K. P. N. D. Ó. S. S. P. M. P. L. R. S. S. Iris Hendrickx, “SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals”.

[33] E. Rosvall, “comparison of sequence classification techniques with bert and named entity recognition,” sweden, 2019.

[34] [Online]. Available: https://ruder.io/state-of-transfer-learning-in-nlp/.

[35] [Online]. Available: https://en.wikipedia.org/wiki/Multilayer_perceptron.

[36] [Online]. Available: https://blog.google/products/search/search-language-understanding-bert/.

[37] [Online]. Available: https://jalammar.github.io/illustrated-bert/.

[38] S. Modi, “Relational Classification using Multiple View Approach,” Ahemedabad.

[39] J. P. A. Ioannidis., ““Why Most Published Research Findings Are False”.In: PLOS Medicine 2.8 (Aug. 2005). doi: 10 . 1371 / journalpmed.0020124. url: https://doi.org/10.1371/journal.pmed.0020124.”.

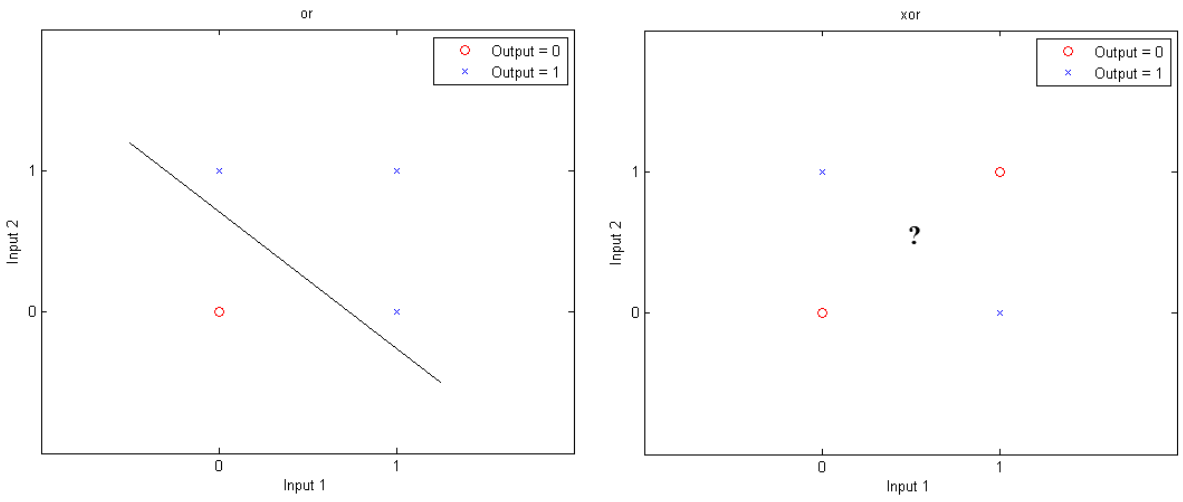
[40] P. B. a. P. A. Babatunde K. Olorisade, ““Reproducibility in Machine Learning-Based Studies: An Example of Text Mining”. In: ICML 2017 RML Submission (2017).”.

Appendix

A. Multi-layered perceptron

Multi-layer perceptron is a feedforward neural network also known as ANN and it refers to neural network composed of multiple perceptron layers. The evolutions of present days tools and algorithms of deep learning started with the research on model neurons combined with mathematics during the fifties. The researchers on perceptrons, introduced by Rosenblatt [20] could only model simple decision boundaries. The single-layer perceptron limited many possible outcomes for the problem. Later to solve the complex problems the multilayer perceptron was introduced. This development enabled the non-linear decision boundaries to be modelled. The perceptron in the multi-layered variant was trained in such a way to create a revolution. A new activation function was introduced to train the multi-layer perceptron. The activation function is normally non-linear which was applied to a linear function and it is denoted by σ (·).

The main reason to get to this approach is to update the weight iteratively. They are updated according to the gradient in the algorithm popularly known as backpropagation which was developed by [21] and later rediscovered by Rumelhart et al [22]. Initially, there was two famous activation function tanh and sigmoid, recently Rectified Linear unit has been replaced. These non-linear functions play a vital role in adjusting the ability of the machine learning models to learn. Without activation, it would be like linear mapping and the algorithm will be incapable to generalize. [23]



Multilayer Perceptron

`

$$\hat{y} = \sigma(W \cdot x + b)$$

In equation 2.2.1 we see that the function is simple, with the input vector as x and the weights denoted as W and b. b represents the bias and is a special form of weight vector to help the (W · x)-vector not stuck around the origin

Activation function	Equation
Sigmoid	$\frac{1}{1+e^{-z}}$
Tanh	$\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$
ReLU	max (0, z)

Activation Function

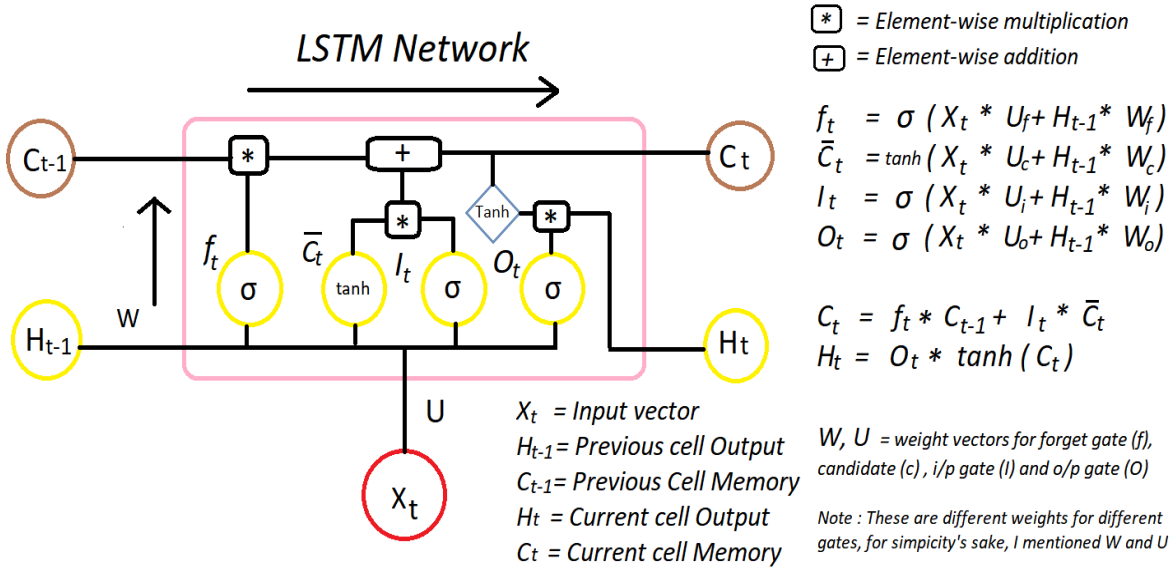
B. Recurrent Neural Network

Recurrent Neural networks (RNN) are also part of Artificial neural network where nodal connections are in the form of directed graph they are the further development of multi-layered perceptron that can solve complex problems. It exhibits the dynamic behaviour. In natural language processing, there are particular problems faced by MLPs while handling of data sequences. It doesn't have the state of storing the previous state in a sequence. Text is rarely normalized in length and original information is neglected in many cases. The rediscovery of Rumelhart et al [3] and Hopfield satisfied this limitation by feeding the previous hidden state in the neural network to the current state. It solves two problems: Firstly, it allows us to look at dependences back in time as the model is influenced by all of the inputs it has seen before. Secondly, it allows us to handle different input lengths because we can run each new data point into the model repeatedly. However, RNNs, in particular, has some downsides. For example, training them via backpropagation can be difficult since their gradient will either become too small or too large, preventing convergence. Gradients that diminish or "explode" make RNNs more difficult to train. Additionally, these models tend to become large, especially in natural language processing. Since RNNs require considerable computation resources to train, some advances have been made that help to improve convergence. One example is the use of L1 or L2 regularizations, which are sometimes used in general MLPs. Although we focus on NLP in this study, it is worth mentioning that RNNs can be used for many kinds of sequential data general time series.

C.Long-Short Term Memory

In response to these factors, Hochreiter and Schmidhuber [24] developed the long short-term memory (LSTM) which used gates to update the hidden states better. LSTM uses input, forget, cell, and output gates. Gates enable a network to learn when to update the hidden state, thereby adding a de-noising layer to the hidden state. This makes the hidden state less likely to contain unimportant updates. Later the model has been developed by Gers et al. [25] and Greff et al. [26] with additional gates from the original ones. They are intended to improve the vanishing and exploding gradient problems associated with regular RNNs, but they don't solve them entirely. And while they make training the model easier

since parameters are less sensitive, it's far from trivial, particularly with deep or stacked networks. This issue will be discussed in detail.



LSTM Architecture

Equation: Different gates of lstm ◦ denotes element-wise operations ((A ◦ B) ij = (A)ij (B)ij) t is a subscript for current time step. W and U is the matrices that transform the input x. The hidden state h. The LSTM combines the current time step t with the hidden state from the previous time step.

One of the complexities of NLP is that textual content might also have future dependencies as well as previous ones. Thus takes the form of the meaning and structure of words being determined by context not yet expressed in a sentence. These issues are significant in NLP, as people may study further into a sentence before comprehending the context of a word. As LSTMs cannot read ahead this problem is solved by bidirectionality [24]. We create two independent LSTMs and let them look left to right and right to left parallelly, then combine outputs at inference, usually by the simple concatenation method. This is referred to as a Bidirectional LSTM or Bi-LSTM. This solves part of the problem; they are no longer conditioning the hidden state both forwards and backward in time but do both in parallel. LSTMs are capable of modelling infinite dependence relationships in theory, but in practice, they do not prove to be accurate space in which it encodes useful information is significantly shorter than that This limit in the range is a considerable problem in NLP, as a reference can be established several sentences or paragraphs.