METAPHOR DETECTION USING TRANSFER LEARNING

An Interim Project Report

Submitted by

SAI AKHIL YERRAMSETTY [cb.en.u4cse17050]
SASIDHAR REDDY B [cb.en.u4cse17013]
ABHAY P [cb.en.u4cse17001]
VIJAY MANI DEV KALURI [cb.en.u4cse17033]

Under the guidance of
Bindu K.R.
(Assistant Professor, Department of Computer Sciene & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENE & ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

श्रद्धावान् लभते ज्ञानम्

Amrita Nagar PO, Coimbatore - 641 112, Tamilnadu

November 2020

ABSTRACT

Metaphors are an important component of natural language as they make the language more creative. It allows people to apply their knowledge of the base, which is typically more concrete and familiar, to inform their understanding of the less-familiar target. While humans can easily understand multiple meanings behind a sentence, it is challenging for machines to comprehend the non-literal meaning behind the sentence.

Metaphors in nominal sentences are made up of two parts: a target, which is the topic of the statement, and a base, which provides information about the target. The use of poor choice of words and acronyms in online texts, posts and comments make it even harder to process natural language. This has triggered the industrial and research community in the last few years to come up with a solution for detection of metaphors. However, these steps are still in their infancy and need further development.

Thankfully, advances in hardware, cloud computing and natural language processing allow the development of NLP based Deep Learning approaches. We will take the datasets from different sources and propose a model that detects metaphors in a given context. Later, we try to improve it using other features to get an appreciable accuracy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

LSTM Long Short Term Memory

POS Parts Of Speech

CNN Convolution Neural Networks

BERT Bidirectional Encoder Representations from Transformers

KNN K Nearest Neighbor

NLP Natural Language Processing

AI Artificial Intelligence

BiRNN Bidirectional Recurrent Neural Networks

WE Word Embedding

NAACL North American Chapter of the Association for Computational Linguistics

CRF Conditional Random Field

SVM Support Vector Machine

GPT2 Generative Pretrained Transformer 2

ERNIE 2.0 Enhanced Representation through Knowledge Integration 2.0

TF-IDF Term Frequency - Inverse Document Frequency

EDA Exploratory Data Analysis

NN Neural Network

DNN Dense Neural Network

Bi-LSTM Bidirectional Long Short Term Memory

ML Machine Learning

Chapter 1

INTRODUCTION

Metaphor is the central problem of language and thinking, it is a commonly used figure of speech. There is one metaphor in every three sentences in daily life. In metaphorical expressions, seemingly unrelated features of one concept are associated with an other concept. The rise of internet media has made the need of a metaphor detection system a necessity since it is challenging for machines to comprehend the intended meaning behind the sentence. Combining this with poor use of language and acronyms in online conversations makes it harder to process the natural language.Metaphors are nowadays used in many areas ranging from regular day to day conversations to important news articles. Hence an efficient model for Metaphor detection is needed to be build. This project deals with building a model that is designed for metaphor detection.

1.1 Problem Domain

Before, all the information about the things happening in the world were gathered from news channels, newspapers or news articles. Now,because of the abundance of electronic texts that have arisen in the Internet era, metaphor identification has become a subject of natural language processing. As a result, academics are increasingly focusing on metaphor detection as the first step in the metaphor processing process. Furthermore, since verb metaphors are so popular in texts, they deserve more consideration. The impact of metaphor detection on natural language processing tasks such as machine translation, reading comprehension, and automated summaries is direct. As said above metaphors exist in almost each and every domain. It is important to find what sentences in the text are actually metaphorical in nature to stop people from understanding the text in a wrong way.

1.2 Problem Definition

Metaphor detection has been a challenging aspect of natural language processing.In the Internet age, lots of electronic texts have emerged, which has made metaphor detection the focus of natural language processing.The effect of metaphor detection directly affects related tasks of natural language processing,such as machine translation, reading comprehension, and automatic summaries.

The existing models will be analyzed to figure out what must be done and where to put in more work to increase the efficiency of the model.Different models related to Natural Language Processing and metaphor detection will be tested out to see which one suits the best. The model will then be trained using a metaphor corpus.The model's performance will then be compared to the existing models and the necessary tweaks will be made. The model will be finally deployed for further testing.The overall implementation's skeletal representation is depicted in Figure 1.1.

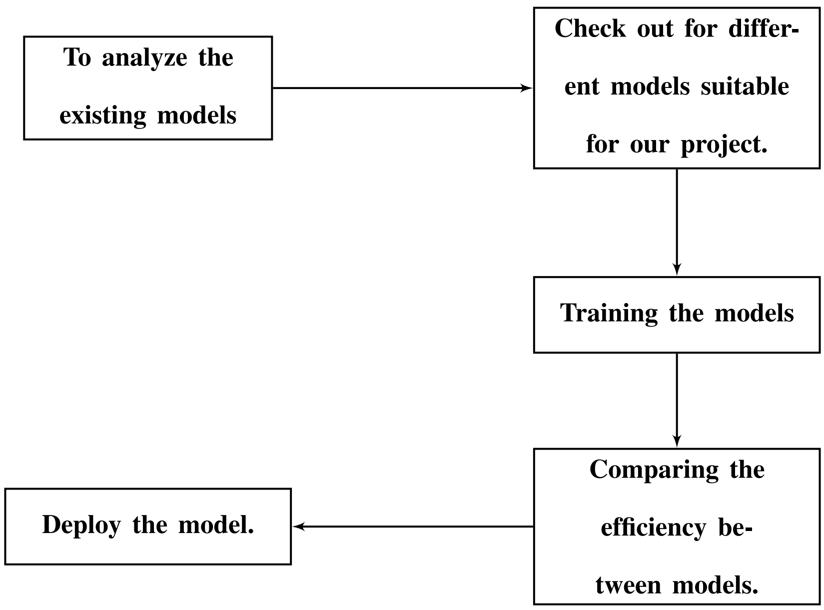

Figure 1.1: Implementation structure

1.3 Motivation

In recent years, people had been using metaphors extensively. Although most of us understand the metaphors in a sentence, there are still some people who cannot understand the non-literal meaning of a word in a sentence. This problem calls for some kind of solution. So we wanted to implement a model that detects metaphorical sentences from a corpora of literal and non-literal sentences.

1.4 Summary

In this report, you will find the experimental findings on different Neural classification algorithms for text classification, in our case metaphor detection. The Contextualised embeddings from pretrained language model are created and fed into a Neural Network Classifier that output a likelihood based on both local and global contextual knowledge. Performances of different classifiers were compared, and the best performing algorithm is eventually deployed.

Chapter 2

LITERATURE SURVEY

2.1 Combining the Attention Network and Semantic Representation for metaphor classification

Zhang et al. (2019) presented a Verb metaphor attention network based on subject-verb-object info which is executed by using LSTM, to encode sentences and calculate the weight of each word. The obstacles faced was that the usage of semantic resources is relatively simple, and there is a lack of deep semantic support. Therefore, the author proposes a word representation method suitable for metaphor classification tasks, which combines the traditional word vector with structural information from the Synonym Thesaurus, so that the word vector can contain the abstraction degree of the word in the metaphor. The they perform LSTM encoding and calculate the weight of each word. At the same time, it can facilitate the understanding of literalness and non-literalness. The experimental results show that the identification effect improves on the existing results, indicating that word representation combining semantic resources and attention network can improve the verb metaphor identification performance.

2.2 Metaphor Detection using verb and noun clustering

Shutova et al. (2010) present a novel approach to automatic metaphor identification in unrestricted text. They start from a small seed set of manually annotated metaphorical expressions, the system is capable of harvesting a large number of metaphors of similar syntactic structure from a corpus. What makes it special is that it does not employ any hand-crafted knowledge, other than the initial seed set, but, in contrast, captures metaphoricity by means of verb and noun clustering. The essence of metaphorical logic based on correlation is the driving force behind the use of clustering techniques for metaphor recognition tasks. The scheme was put to the test with a collection of metaphorical phrases that represented verb-subject and verb-object constructions in which the verb was metaphorically used. Being the first to employ unsupervised methods for metaphor identification, the system operates with the precision of 0.79.

2.3 ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding(2019)

Yu Sun (2019) Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, Haifeng Wang. Proposed a continual pre-training framework named ERNIE 2.0 which incrementally builds pre-training tasks and then learn pre-trained models on these constructed tasks via continual multi-task learning. Based on this framework, several tasks were made and trained the ERNIE 2.0 model to capture lexical, syntactic and semantic aspects of information in the training data were conducted. Three kinds of unsupervised language processing tasks were conducted to verify the effectiveness of the proposed framework. Experimental results demonstrate that ERNIE 2.0 model outperforms BERT and XLNet on 16 tasks including English tasks on GLUE benchmarks and several similar tasks in Chinese.

2.4 XLNet: Generalized Autoregressive Pre-training for Language Understanding(2019)

Zhilin Yang (2019)Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell,Ruslan Salakhutdinov.,propose XLNet, a generalized autoregressive pretraining method that (1) enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and (2) overcomes the limitations of BERT thanks to its autoregressive formulation. It also integrates ideas from Transformer-XL, the state-of-the-art autoregressive model, into pretraining. Empirically, under comparable experiment settings, XLNet outperforms BERT on 20 tasks, often by a large margin, including question answering, natural language inference, sentiment analysis, and document ranking.

2.5 Metaphor Detection Using Contextual Word Embeddings From Transformers(2020)

Jerry Liu (2020)Jerry Liu, Nathan O'Hara, Alexander Rubin, Rachel Draelos, Cynthia Rudin. Propose using both BERT and XLNet language models to create contextualized embeddings and a bidirectional LSTM to identify whether a given word is a metaphor. Their best model achieved F1-scores of 0.68 on VUA AllPOS, 0.73 on VUA Verbs, 0.669 on TOEFL AllPOS, and 0.697 on TOEFL Verbs, placing 7th, 6th,5th, and 5th respectively . Its shown that a KNN classifier provides a similar F1-score on a validation set as the LSTM and yields different information on metaphors.

2.6 Using Language Learner Data for Metaphor Detection

Stemle and Onysko (2018) showed that The combination of WEs with a BiRNN is capable of recognizing metaphorical usage of words better than many other already tested approaches. More importantly, their design does not rely on WordNet or VerbNet information, and does not need concreteness or abstractness information like many successful architectures from previous annual workshops at NAACL. Besides VUA, their system only needs running text.

2.7 Metaphor Detection Using Context and Concreteness

Hall Maudslay et al. (2020) model was designed to try and exploit knowledge of lexical concreteness and contextual meaning to identify metaphors. Their results improved over the previous best performing system by an average of 8.0but trailed behind the leader of the task by 5.0

2.8 Neural Metaphor Detecting with CNN-LSTM Model

Wu et al. (2018) introduced a CNN-LSTM model with CRF or softmax layer for the metaphor shared task to detect metaphors in texts.We combine CNN and LSTM to capture both local and long-distance contextual information to represent the input sentences which are lemmatized as part of preprocessing.They compared the performance of CRF and softmax classifier with weighted loss.

2.9 Di-LSTM Contrast : A Deep Neural Network for Metaphor Detection

Swarnkar and Singh (2018) described a deep neural architecture Di-LSTM Contrast Network for metaphor detection.The input to the model is given as pre-trained word embeddings. An encoder uses these word embeddings to encode the context of the sentence with respect to the target word using forward and backward LSTMs. Their system achieves an overall F1 score of 0.570 on All POS category and 0.605 on the Verbs category.

2.10 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Swarnkar and Singh (2018) proposes a new pre-training goal, the "masked language paradigm," to overcome the previously stated unidirectional constraints (MLM). The masked language model hides certain tokens from input at random. The aim is to use the meaning to predict the vocabulary id of a masked expression. The MLM model goal enables the representation to fuse the left and right sense, which aided in the pre-training of a deep bidirectional transformer.

2.11 Summary

Based on understanding from the literature survey, it can be summarised that certain important aspects of the previous papers could be combined to build a more efficient model that can detect metaphors using Natural Language Processing. Implementation of attention network, BERT, open AI's gpt2 transformer over the metaphor data set for detection of metaphors within a sentence.

2.12 Data Set

For the first dataset, the Birke and Sarkar (2006) TroFi Dataset. This dataset consists of sentences with 50 different verbs. There are multiple literal and metaphoric sentences for each verb. There are a total of 3737 samples in this dataset.

	verb	sentence	verb_idx	label
0	absorb	An Energy Department spokesman says the sulfur...	22	0
1	absorb	The yellow beta carotene pigment absorbs blue ...	5	0
2	absorb	This time , the ground absorbed the shock wave...	5	0
3	absorb	" Vitamins could be passed right out of the b...	12	0
4	absorb	As Eliot wrote : " In a warm haze , the sultr...	14	0

Figure 2.1: Trofi Dataset.

For the second dataset,the VUA dataset. This dataset has sentences from different verbs. There are a total of 17380 samples made with 2063 unique verbs.

	verb	sentence	verb_idx	label
0	fail	Ca n't fail to be entertaining .	2	0
1	go	How much was he going to tell her ?	4	0
2	win	Up until that news hit the Committee , Don had...	10	0
3	go	Could go on to the rugby and go with them coul...	1	0
4	go	Finally , we went to the office and they gave ...	3	0

Figure 2.2: Trofi Dataset.

The above two datasets are similar in nature with same type sof columns and they are different in nature in terms of difference in number of unique verbs in the data. TroFi will help in understanding the model's performance in terms of less variety and VUA will help us understanding the model's performance in terms of large variety of verbs.

2.13 Software/Tools Requirements

A) Jupyter Notebook/Google Colab - An open-source web application that allows you to create and share documents that contain live code, equations. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization and machine learning.

B) Pandas, Numpy (Python libraries) – used for processing and extracting features from the data

C) Keras, Pytorch – open source library used for deep learning applications

D) Scikit Learn - for dataset modifications and model evaluation

E) Natural Language ToolKit(NLTk) - It is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

Chapter 3

PROPOSED SYSTEM

3.1 System Analysis

3.1.1 System Requirement Analysis

Purpose

Through our model, we aim to separate the metaphorical and non metaphorical sentences from a given corpora. Our aim is to build a Deep Learning model that works towards detecting metaphorical sentences by analyzing the text of the sentence using NLP.

Functional Requirements

- Our model should be able to verify the metaphorical nature of sentences and provide the classification of sentence in terms of 'literal' or 'non-literal'.
- Any python 3 installed device should be able to run the model

Non-Functional Requirements

- To run the algorithm a user must need at least 8GB of ram or any cloud computing or online service such as google colaboratory.
- A good GPU is recommended to make the model perform calculations faster or to train the model faster

3.1.2 Modules

Obtaining Contextualized Word Embeddings

Pre-processing

- Removing null values from the dataset is one of the important steps in data wrangling. These null values adversely affect the performance and accuracy of any machine learning algorithm. So, it is very important to remove null values from the dataset before applying any machine learning algorithm to the dataset.

- Dropping duplicates from the data. Duplicates are an extreme case of nonrandom sampling, and they bias the fitted model. So removing the redundant data is necessary in order to avoid overfitting.
- Converting all sentences to lowercase. Converting all your data to lowercase helps in the process of preprocessing and in later stages in the NLP application, when parsing is done.
- Tokenizing the sentences. In order to get the computer to understand any text, we need to break that word down in a way that our machine can understand. This is why tokenization is essential. This process breaks the sentences into tokens which becomes convenient for the model to process.
- Lemmatizing the tokens. Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.

Visualisation

- Count plot : For checking the balance of datasets, number of sentences for a list of verbs, number of literal and non-literal sentences based on sentence polarity.
- Histogram : For text length vs number of texts
- Wordcloud : For frequent words.
- Horizontal Bar plot : For bigrams and trigrams analysis.
- KDE plot: For finding density of each text length.
- Strip plot: For distribution of different polarity sentences against text lengths.

Contextualised Word Embeddings

- Breaking down of sentence into a list of words.
- We add [CLS] and [SEP]. For the classification task, a vector representing the whole input sentence must be fed to a classifier. In BERT model, the hidden state of the first token is taken to represent the whole sentence. To achieve this, an additional token has to be added manually to the input sentence, the token [CLS] is chosen for this purpose.
- Padding the sentence with [PAD] tokens so that the total length equals to the maximum length specified.
- Converting each token into their corresponding IDs in the model
- The above tokens are sent into BERT model to get a pooled output

Classification

This module deals with the classification. The word-embeddings obtained as an output from the previous step are divided into a test-set and a training set. The training set is used to train the classifer. We will start by studying traditional classifiers like Naive Bayes, SVM and logistic regression. Then proceed with neural networks like LSTM and BiLSTM as classfiers with sigmoid activation function. Sigmoid is used here because the value should be between 0 and 1 inorder to predict the metaphoricity of the sentence

Using bidirectional LSTM, it will run your inputs in two ways, one from past to future and one from future to past. This will help us in preserving information of both past and future

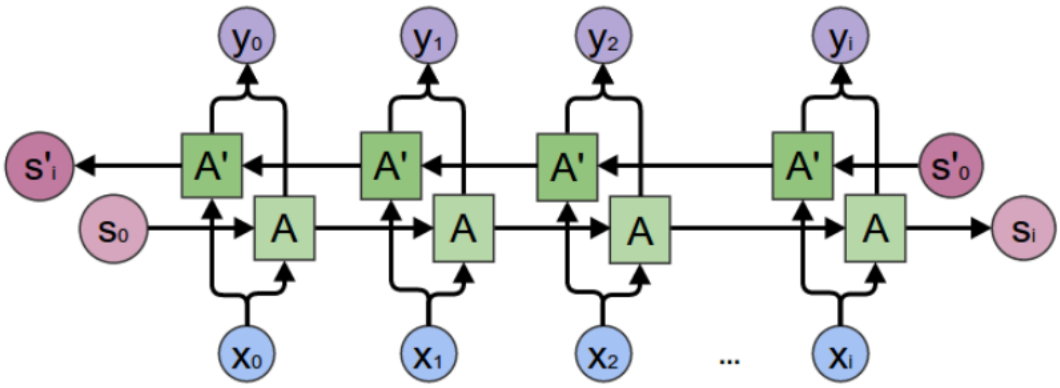

Figure 3.1: Bi LSTM Architecture obtained from Swarnkar and Singh (2018)

Evaluation and Comparison

After the classifier is trained, the next step is to predict the metaphoricity of the sentences using the trained classifiers. The score for each of the classifier is calculated and compared with other classifiers. Based on the metrics like precison, recall and F1 score, the performance of the classifier is evaluated for over-fitting or under-fitting and the hyper-parameters are tuned. Other changes if necessary are also made.This is an iterative process. Finally, we hope to obtain a classification model which performs well enough to be deployed for applications in the real world.

3.2 Architecture Diagram

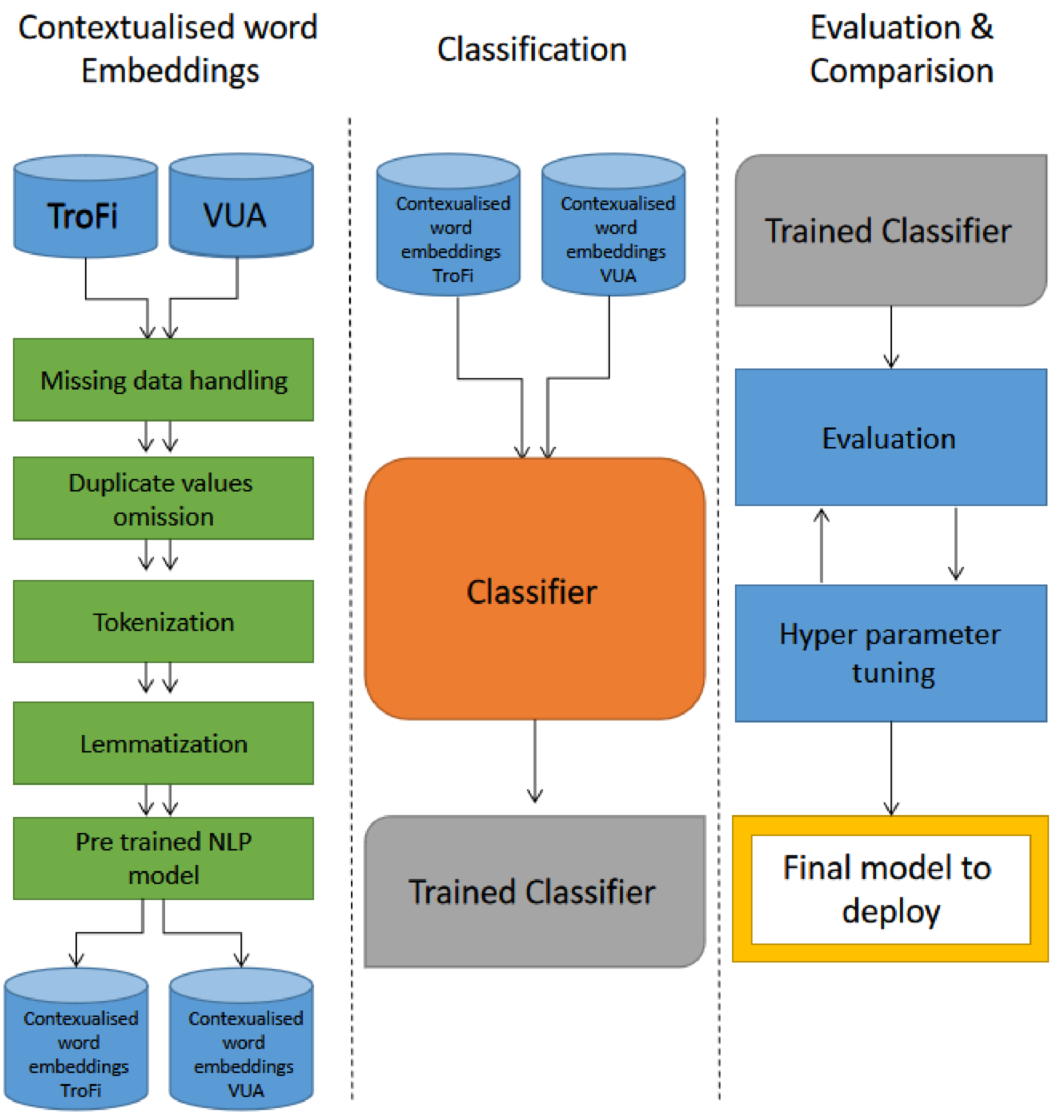

Figure 3.2: Architecture Diagram

As shown in Figure 3.2 We use pre trained state-of-the-art NLP model (Devlin et al. (2018) BERT / Sun et al. (2020) ERNIE 2.0) to get the contextualised word embeddings and then pass the same to a classifier like the Naive Bayes, Support Vector Machines or a two-layer bi-directional long short-term memory (Shen et al. (2019) Bi-LSTM) neural network architecture.Later, the classifiers are evaluated and the hyperparameters are tunes and the best performing model is finally deployed.

Chapter 4

IMPLEMENTATION AND TESTING

Our method for metaphor detection begins with generating contextualized word embeddings for each word in a sentence using the hidden states of pretrained language model.

4.1 Preprocessing

So the required modules are imported. We installed the Python transformer library, which includes a PyTorch implementation of BERT and several pretrained BERT models. For the implementation, we have used the TroFi and VU Amsterdam Datasets.The duplicates sentences were removed.Labels were manually added for simplifying the preprocessing and the Visualizations. Necessary visualizations were made .The stop words are removed and Lemmatization was made on the sentences.

4.2 Pretrained Models

A pre-trained model is one that has already been trained to solve a similar problem. Instead of starting from scratch to solve a similar problem, you start with a model that has already been trained on another problem. The following are some few famous pre-trained models that have been tested out.

4.2.1 BERT

The main tehnial innvtin of BERT is the application of Trnsfrmer's bidiretinl trining, a common ttention mdel, to lnguge mdelling. This is in contrast to previous efforts, which looked at a text sequence from left to right or a combination of left-to-right and right-to-left trining.The findings show that a bidirectionally trined lnguge mdel may have a better sense of lnguge background and flow than single-direction lnguge mdels.

The researchers describe a new technique called Msked LM (MLM) that allows bidirectional training in previously impossible models. BERT employs Trnsfrmer, an attention mechanism that learns contextual relationships between words (or sub-words) in a document. Trnsfrmer inludes tw separte mechnisms in its vnill frm — an encder tht reds the text inut nd decder tht rdus reditin fr the tsk.Since BERT's goal is to create a language model, only the encder mechanism is required. Ggle describes the detailed wrks of Transsfrmer in a per. Unlike directinal mdels, who read the text in a sequential order (left-to-right or right-to-left), the Trnsfrmer encder reads the whole sequence of words all at once. As a result, it is considered bidirectional, but it would be more accurate to say it is non-directional.This charcteristic allows the mdel to learn the context of a word based on all of its surroundings (to the left and right of the wrd). The diagram below is a high-level description of the Transsfrmer encder. The output consists of a series of tokens that are first embedded in vectors and then processed in the neural network. The output is a sequence of H-size vectors, each of which corresponds to an input with the same index. BERT may be used for a broad range of lnguge tasks by simply adding a smll lyer to the core model: clssifitin tsks such as emotion nlysis are done in a similar way to Next SentencE classifitin, by dding a classifitin lyer t the Trnsfrmer utut fr the [CLS] token. Most hyper-parameters remain the same in fine-tuning training as in BERT training, and the paper provides specific guidance on the hyper-parameters that demand tuning. This technique has been used by the BERT team to achieve state-of-the-art success on a wide range of challenging natural language tasks.

4.2.2 XLNet

The most recent and best model to emerge from the burgeoning field of Natural Language Processing(NLP) is XLNet Zhilin Yang (2019). XLNet is an auto-regressive language model that uses a transformer architecture with recurrence to produce the joint likelihood of a sequence of tokens. Its training goal is to determine the likelihood of a word token based on all permutations of word tokens in a sentence, rather than only those to the left or right of the target token.XLNET incorporates ideas from Transsfrmer-XL, a state-of-the-art autoregressive model, into pretrining. Google's Trnsfrmer is a mdel used for lnguge trnsltin purposes. It bsiclly revolves around the word "attention." It's an encoder-decoder method in which you map one sequence to another

— English to French. To trnslte sentence from English to Frenh, the decder must look at the whole sentence and selectively extract information from it at some point in time (because the order of tokens in English does not have to be the same in Frenh). As a result, the decder also has access to all of the encder's secret states.Two crucial features of Trnsfrmer-XL have been integrated into XLNET. Positional Ending — keep track of the position of each action taken in a sequence. Segment recurrence — in each layer, find the secret state of the first segment in memory and update attention accordingly. It allows memory to be reused for each line. XLNET is a generlized autoregressive model in which the next token is determined by all previous tokens. XLNET is "generlized" since it captures bi-directional context using a mechanism known as "permutation lnguge mdeling." It combines the ideas of auto-regressive models and bi-directional context modelling thus overcoming BERT's disadvantages. It out-performs BERT on 20 tasks, often with a large margin in tasks like question answering, natural language inference, emotion analysis, and document ranking.

4.3 Word Embeddings

The sentences are broken down into a list of words. An additional token is added manually to the input sentences, the token [CLS] is chosen for this purpose so that the NLP model can recognise the whole sentence.The sentences are then padded with [PAD] tokens and assigned with corresponding IDs and later passed through the NLP model (Devlin et al. (2018) BERT / Sun et al. (2020) ERNIE 2.0) which returns the contextualized word embeddings.

4.4 Classifier

In machine learning, a classifier is an algorithm that automatically sorts or categorises data into one or more classes.Its primary use will be to classify if the sentence is metaphoric or not. The following are some of the classifiers that have been used.

4.4.1 Dense Neural Network

A dense network is one in which each node's number of connections approaches the maximum number of nodes. Each node is connected to nearly every other node. A fully wired network is one on which every node is connected to every other node in the same way. The dense layer is an ordinary deep-connected neural network layer. The dense layer performs the following operations on the input and returns the result. Syntax: output = activation (point (input, core) + offset), where input represents input, kernel represents weighing data, point represents the numerical product of all inputs and their corresponding weights, and offset represents deviation. As a value used to optimize the model in machine learning, activation represents an activation function. The names of dense layers indicate that these layers are fully connected by neurons in the network layer. Each neuron in one layer receives input from all neurons present in the previous layer, so they are tightly connected to each other. In other words, the dense layer is a fully connected layer, that is, all neurons in a layer. These layers are connected to neurons in the next layer. The tightly coupled layer provides training attributes for all attribute combinations from the previous layer, while the convolutional layer relies on consistent attributes. The repeated fields are small. The dense layer of the neural network maps each neuron in one layer to each neuron in the next layer. This allows the potential function to be as close as possible within a given slice width. This also means that many parameters must be set, so the training of very wide and very dense networks requires a lot of calculations.

4.4.2 LSTM

Long short-term memory is an artificial recurrent neural network architecture. LSTM has feedback relations, unlike normal feed-forward neural networks. It can handle not just single data points, but also whole data sequences. Long-term and short-term memory networks (LSTM) are a type of repetitive neural network capable of learning the dependency of order in sequence prediction problems. It is used in complex problem areas such as machine translation, speech recognition, and more. It can process entire data streams such as voice or video. A common LSTM unit consists of a cell, an entrance door, an exit door and a forgetting door. The cell (the memory part of the

LSTM unit) stores values at any time interval and on all three gates. Regulate the flow of information in and out of the cell. Long delays with certain problems are overcome with LSTM, where noise, distributed representations and continuous values are also processed. With LSTM, there is no need to maintain a finite number of states in the cell. Advances as required in the Hidden Markov Model. LSTMs offer us a wide range of parameters such as learning rates and entry and exit biases.

4.4.3 Bi-LSTM

Bidirectional LSTMs(Bi-LSTM) are a type of LSTM that can be used to increase model performance in sequence classification problems. Bidirectional LSTMs train two instead of one LSTM on the input sequence in problems where all timesteps of the input sequence are visible . A bi-directional LSTM or biLSTM is a sequence processing model that consists of two LSTMs: one moves the input forward and the other backward. BiLSTM effectively increases the amount of information available to the network, improves the context available to the algorithm by replicating the first recurring layer on the network and then providing the input sequence as it is input into the first layer and one The reverse copy of the deployed is the input sequence for the replicated layer. RNN. The bidirectional recurring neural network (BRNN) can be trained using any input information available in the past and future of a particular time step. Mix mode that describes how the outputs should be mixed back and forth before moving to the next level. The options are: - 'Sum': The results are added up. - "Mul": the results are multiplied.- "concat" (default): The results are concatenated, which means that the output on the next level is twice as high. - "Bird": The average of the results is determined.

Chapter 5

RESULTS AND DISCUSSION

The datasets were pre-processed and passed throughthe pretrained BERT model and the the contextualizes word embeddings were successfully generated. These contextualized word embeddings is then passed through a DLL, LSTM and Bi-LSTM neural network classifier respectively and the metaphoricity is calculated, evaluated and compared with the former models.The following are the results obtained.

VUA Dataset	
Model	**Accuracy**
BERT	0.71
XLNet	0.73

TroFi Dataset	
Model	**Accuracy**
BERT	0.71
XLNet	0.75

VUA Dataset			
Algorithm	**Precision**	**Recall**	**F1 Score**
DNN	0.75	0.88	0.81
LSTM	0.78	0.80	0.79
Bi-LSTM	0.82	0.85	0.83

TroFi Dataset			
Algorithm	**Precision**	**Recall**	**F1 Score**
DNN	0.58	0.62	0.60
LSTM	0.64	0.63	0.64
Bi-LSTM	0.74	0.73	0.74

Chapter 6

CONCLUSION

Model was created successfully that could predict the metaphoricity of a sentence. Contextualised embeddings were created using the BERT pretrained language model and fed into a Neural Network Classifier with a sigmoid layer that output a likelihood based on both local and global contextual knowledge. Performances of different classifiers were compared namely the DNN, LSTM and Bi-LSTM when paired up with the BERT pretrained model.It's observed that the Bi-LSTM classifier is able capture long-range relationships between words in the same sentence which may reveal the presence of metaphors and has outperformed the former classifiers.

Chapter 7

FUTURE ENHANCEMENT

The model is now fine tuned with respect to our use-case that is metaphor detection.The performance of the model can further be improved if any new state of the art pre-trained models can be implemented. Also, since classifiers play an important role in the performance, the accuracy can be pushed further by implementing any new state of the art classifiers. Further innovation can be brought when this model is fine tuned with a different language metaphoric datasets. Already previous researchers have implemented the model to a Chinese dataset. The idea of implementation of native language dataset over the model, native language could be Hindi, Telugu, Tamil, Malayalam and many others. Also further innovation is possible by translating the English language to an intermediate language which will later be translated to native language while keeping the metaphoric meaning intact.There are many good translators in the past like google translate but the key challenge would be to keep the metaphoric meaning of the sentence intact.

REFERENCES

1. Birke, J. and Sarkar, A. (2006). "A clustering approach for nearly unsupervised recognition of nonliteral language." *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

2. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding. cite arxiv:1810.04805Comment: 13 pages.

3. Hall Maudslay, R., Pimentel, T., Cotterell, R., and Teufel, S. (2020). "Metaphor detection using context and concreteness." *Proceedings of the Second Workshop on Figurative Language Processing*, Online. Association for Computational Linguistics, 221–226.

4. Jerry Liu, Nathan O'Hara, A. R. R. D. C. R. (2020). "Metaphor detection using contextual word embeddings fromtransformers(2020)." *Association for Computational Linguistics*, Duke University. Association for Computational Linguistics, 250–255.

5. Shen, M., Zhang, X., and Li, X. (2019). "Attentional bi-directional lstm for semantic attribute prediction.." *ICVIP*. ACM, 217–221.

6. Shutova, E., Sun, L., and Korhonen, A. (2010). "Metaphor identification using verb and noun clustering." *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China. Coling 2010 Organizing Committee, 1002–1010.

7. Stemle, E. and Onysko, A. (2018). "Using language learner data for metaphor detection." *Proceedings of the Workshop on Figurative Language Processing*, New Orleans, Louisiana. Association for Computational Linguistics, 133–138.

8. Sun, Y., Wang, S., Li, Y.-K., Feng, S., Tian, H., Wu, H., and Wang, H. (2020). "Ernie 2.0: A continual pre-training framework for language understanding.." *AAAI*. AAAI Press, 8968–8975.

9. Swarnkar, K. and Singh, A. K. (2018). "Di-LSTM contrast : A deep neural network for metaphor detection." *Proceedings of the Workshop on Figurative Language Processing*, New Orleans, Louisiana. Association for Computational Linguistics, 115–120.

10. Wu, C., Wu, F., Chen, Y., Wu, S., Yuan, Z., and Huang, Y. (2018). "Neural metaphor detecting with CNN-LSTM model." *Proceedings of the Workshop on Figurative Language Processing*, New Orleans, Louisiana. Association for Computational Linguistics, 110–114.

11. Yu Sun, Shuohuan Wang, Y. L. S. F. H. T. H. W. H. W. (2019). "Ernie 2.0: A continual pre-training framework for language understanding(2019)." *Computation and Language (cs.CL)*. Computation and Language (cs.CL), 3–11.

12. Zhang, D., Lin, H., Liu, X., Zhang, H., and Zhang, S. (2019). “Combining the attention network and semantic representation for chinese verb metaphor identification.” *IEEE Access*, 7, 137103–137110.

13. Zhilin Yang, Zihang Dai, Y. Y. J. C. R. S. (2019). “Xlnet: Generalized autoregressive pre-training for language understanding(2019). Association for Computational Linguistics, 1–9.