CLASSIFICATION OF CONTRADICTORY OPINIONS IN TEXT USING DEEP LEARNING METHODS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
İskender Ülgen OĞUL

December 2020
İZMİR

I would like to dedicate my work to all souls who are in search of the truth and science. Science is a never ending journey only flows to future in time. This can be justified by saying *The present is theirs; the future, for which I really worked, is mine*. The pursue of the future keeps the flame of knowledge alive and strong.We may be limited to our times technology and have limited sight for future. Whenever we are consumed by the idea of limitation one should remember that *We can only see a short distance ahead, but we can see plenty there that needs to be done*. Our present once was a future for our ancestors yet they never stopped imagining and the search of the truth. This dedication can only mean that *Imagination is more important than knowledge*. We all must know that what we see around us became reality with the imagination of their creators. Everyday, many sees mediocrity but the ones in the search of light must see possibilities. *There is one truth we can not deny is that Where there is light there is life...*

ACKNOWLEDGMENTS

I would like to present my special thanks to my dear advisor Dr. Selma TEKİR who always showed me the right way during my research. Her ambition and determination are real definition of a science woman. Her patient personality allowed me to learn from my mistakes and move forward on my academic path with strong steps. I also would like to thank my beloved mom Selma OĞUL who always stood by my side during my research. She always encouraged me to the future as a mom and supported my work as a teacher.

This thesis is supported by TUBITAK under TUBITAK 2210 - C Domestic Priority Graduate Scholarship Program. I present my thanks and gratitude to TUBITAK (Türkiye Bilimsel ve Teknik Araştırma Kurumu) Scientific and Technological Research Council of Turkey for supporting my 1649B021901688 number thesis both financial and morally.

ABSTRACT

CLASSIFICATION OF CONTRADICTORY OPINIONS IN TEXT USING DEEP LEARNING METHODS

Natural language inference (*NLI*) problem aims to ensure consistency as well as accuracy of propositions while making sense of natural language. Natural language inference aims to classify the relationship between two given sentences as contradiction, entailment or neutrality. To accomplish the classification task, sentences or words must be translated into mathematical representations called vectors or embedding. Vectorization of a sentence is as important as the complexity of the classification model. In this study, both pre-trained (*Glove, Fasttext, Word2Vec*) and contextual word embedding methods (*BERT*) were used for comparison and acquire the best result.

One of the natural language processing tasks NLI, is highly complex and requires solutions. Conventional machine learning methods are insufficient to carry out natural language processing solutions. Therefore, more advanced solutions are required. This study used deep learning methods to perform the classification task. Unlike conventional machine learning approaches, deep learning approaches reduce errors while increasing accuracy by repeating the data many times.

Opinion sentences have complex grammatical structures that are difficult to classify. This study used Decomposable Attention and Enhanced LSTM for natural language inference to perform NLI classification task. Using the advanced LSTM deep learning method and Bert contextual vectors for natural language extraction on the SNLI dataset, an accuracy result 88.0% very close state of the art result 92.1% was obtained. In order to show the usability of the developed solution in different NLI tasks, an accuracy of 80.02% was obtained in the studies performed on the MNLI data set.

ÖZET

METİNLERDEKİ KARŞIT FİKİRLERİN DERİN ÖĞRENME YÖNTEMLERİ İLE SINIFLANDIRILMASI

Doğal dil çıkarımı (*NLI*) problemi, doğal dili anlamlandırırken önermelerin doğruluğunun yanında tutarlılığını da sağlamayı hedeflemektedir. Doğal dil çıkarımı, verilen iki cümlenin birbiri arasındaki ilişkinin karşıtlık, örtüşme – gerekseme veya taraf-sızlık olarak sınıflandırmasını hedefler. Sınıflandırma görevini gerçekleştirmek için cümleler ya da kelimeler vektör ya da gömme olarak adlandırılan matematiksel gösterimlere çevrilmiş olmalıdır. Bir cümlenin vektörizasyonu, sınıflandırma modelinin karmaşıklığı kadar önemlidir. Bu çalışmada, hem önceden eğitilmiş (*Glove, Fasttext, Word2Vec*) hem de bağlamsal kelime gömme yöntemleri (*BERT*) karşılaştırma ve en iyi sonucu elde etmek için kullanılmıştır.

Doğal dil işleme görevlerinden biri olan NLI oldukça karmaşıktır ve gelişmiş çözümler gerektirmektedir. Geleneksel makine öğrenmesi metodları doğal dil işleme çözümleri gerçekleştirmek için yetersizdir. Bu yüzden, daha gelişmiş çözümler gerekmiştir. Bu çalışma sınıflandırma görevi gerçekleştirmek için derin öğrenme yöntemlerinden faydalanmıştır. Geleneksel makine öğrenmesi yaklaşımlarından farklı olarak, derin öğrenme yaklaşımları veri üzerinde birçok kez tekrarlama gerçekleştirerek (Epoch), doğruluğu arttırırken hatayı düşürmektedir.

Düşüncesel cümleler sınıflandırması zor olan karmaşık gramer yapılarına sahiptir. Bu çalışma, ayrıştırılabilir ilgi ve doğal dil çıkarımı için gelişmiş LSTM derin öğrenme modellerini, NLI sınıflandırma görevini gerçekleştirmek için kullanmıştır. SNLI veri seti üzerinde doğal dil çıkarımı için gelişmiş LSTM derin öğrenme yöntemi ve Bert bağlamsal vektörleri kullanılarak, rapor edilmiş en iyi sonuca %92.1 çok yakın bir değer %88.0 elde edilmiştir. Geliştirilen çözümün farklı NLI görevlerinde kullanılabilirliğini gösterebilmek için MNLI veri seti üzerinde yapılan çalışmalarda %80.02 doğruluk elde edilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

NLI . . . . . Natural Language Inference
BERT . . . . Pre-training of Deep Bidirectional Transformers for Language Understanding
LSTM . . . . . Long Short-Term Memory
SNLI . . . . . Stanford Natural Language Inference
MNLI . . . . . Multi-Genre Natural Language Inference
RTE . . . . . Recognizing Textual Entailment
AI . . . . . Artificial Intelligence
ESIM . . . . . Enhanced LSTM for Natural Language Inference
SOTA . . . . . State of the Art
ANLI . . . . . Adversarial Natural Language Inference
NIST . . . . . National Institute of Standards and Technology
TREC . . . . . Text REtrieval Conference
DUC . . . . . Document Understanding Conferences
JTE . . . . . Joint Topic Expression
BiLSTM . . . . . Bidirectional Long Short-Term Memory
SVM . . . . . Support Vector Machine
MLM . . . . . Masked Language Model
ELMO . . . . . Embeddings from Language Models
LSA . . . . . Latent Semantic Analysis
GPT . . . . . Generative Pre-trained Transformer
ReLU . . . . . Rectified Linear Unit
RNN . . . . . Recurrent Neural Network
TF . . . . . Term Frequency
IDF . . . . . Inverse Document Frequency
NLP . . . . . Natural Language Processing
OOM . . . . . Out Of Memory
SRL . . . . . Semantic Role Labeling

CHAPTER 1

INTRODUCTION

In today's world, technology advances with a rapid speed. Contributions from both corporations and open source programmers to the technology offer even faster advancements. The more the technology is improved, faster the people can access it with a low cost. It is assumed that each day the amount of people that comes online increases rapidly and the data they created are increased exponentially. Considering the latest achievements on data storage systems and data warehouses, it is no longer a problem to store massive amounts of raw data. Without the space and connection limitations, data created by individuals on daily basis are fetched, indexed, and stored by following the rules of privacy preserving of an individual. This increase of the data offers valuable information for research areas, governments, and companies. It is mentioned that the amount of online people increases along with the data they create each day. Data created by individuals carry high value since they contain personal information. Their opinions are composed of their daily problems, reviews of something they experienced or their political ideas. Gathering these opinions and interpreting them using suitable methods can provide inspiring results and solutions for specific problems. This work aims to classify opinionated sentences into their corresponding inference classes. Classification task is carried out by implementing various types of feature extraction approaches using different language inference corpora and two proven deep learning inference classification models.

It is known that Natural Language Inference (NLI) is a popular yet complex language task. NLI aims to classify opinionated sentences into contradiction, entailment or neutral (having no semantic relation). Contradiction is an opinionated description where two sentences disagree with each other semantically. In entailment, two sentences have supportive claims while in a neutral description two sentences have no semantic relation between them. NLI task focuses on finding the context relations rather than finding a semantic similarity. Unlike conventional sentence similarity, NLI task focused on understanding the underlying context of the given text pairs and decide whether pairs are contradicts – entails or have no relation.

Earlier research shows that solutions for NLI were based on lexical information (Harabagiu et al., 2006) This lexical information is captured using WordNet (Miller, 1995) and later SentiWordNet (Esuli and Sebastiani, 2006) Inference research area was restricted to small train corpus such as RTE (Voorhees, 2008). With the introduction of pre-trained word vectors such as Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014) and Fasttex (Mikolov et al., 2018), most of the research approaches started using new word vectors for their solutions. Major drawback of the word vector-based language inference solutions was the limitations of the training corpora. In 2015, Stanford Natural Language Processing Group released a new corpus for natural language inference research area. The most important side of the new corpus, Stanford Natural Language Inference SNLI (Bowman et al., 2015) is that it is completely hand-crafted and human annotated. With the advancements of word vectors and a new corpus, most of the NLI research used data-driven solutions and introduced new and strong algorithms. Not long after another drawback appeared on the horizon. The new problem was that SNLI corpus had a restricted context. Therefore, proposed solutions consumed the SNLI benchmark quickly. A solution to this problem was to create a new corpus named MultiNLI (Williams et al., 2018). MultiNLI addresses these problems and structures a new corpus. Proposed approach was to craft a corpus from multiple genres using written and spoken English. It is assumed that this solution would generate a new benchmark and challenge the existing algorithms. This solution worked for a while but not long enough. In 2018, Google AI team introduces their new transformer language model BERT (Devlin et al., 2019). BERT learns from the context and produces contextualized word embeddings. Moreover, BERT implements a fine-tuning approach which enables its use in downstream tasks. In downstream tasks, using BERT with labeled training data provides much better results for a wide range of natural language processing tasks. With this newly introduced transformer, SNLI and MultiNLI became obsolete as benchmarks. This problem drew attention and was addressed by Facebook AI team. Facebook AI team proposed a new approach named human and machine in the loop process. Human and machine in the loop approach states that in order to create a challenging NLI corpus, human generated premise and hypothesis pairs must be tested on the best inference model and take the wrongly predicted results. These wrong predictions later verified by human annotators and if premise – hypothesis pairs meets the required standards; they are included in the new training corpus. The new training corpus later was used to create a new model and the same process is repeated for three runs. The new approach introduced a new NLI corpus called Adversarial NLI (Nie

et al., 2020). Unlike previous inference corpora, Adversarial NLI provides a much more challenging inference corpus. Also, this new approach introduces the never-ending corpora creation. The more powerful models become new corpus will be harder to analyze using human model in the loop approach.

Our solution is based on contextualized word embeddings obtained from BERT. Later, these contextualized word embeddings are used with two best natural inference algorithms, Decomposable Attention (Parikh et al., 2016) and ESIM (Chen et al., 2017). Each algorithm set the state-of-the-art results in their time of release. Through time, natural language inference research is introduced with multiple corpora. Each corpus addresses a specific problem and attempts to solve it. Considering all the improvements, this work builds a solution which used both pre-trained and contextualized word embeddings with two strong algorithms. In order to give comparable results and benchmark, our models are trained on all the corpora. Each trained model is tested on all test sets each corpus can provide. It is believed that this would provide better comparison and address the drawbacks of each corpus. It is long discussed that there is a relation between training data size and the model accuracy. Later, to justify the data magnitude relation with model accuracy, all corpora are concatenated to train new model. First concatenation is performed on SNLI and MultiNLI, second is composed of all the corpora. Concatenated corpus models are only trained using BERT to give better insights about contextualized word embeddings.

Our solution provides 88% accuracy on SNLI test set while recently reported state of the art result is 92.1%. It must be noted that MultiNLI has two different test sets, matched and mismatched. Matched test set is derived from the train corpus genres distribution while mismatched test set is composed of the genre examples that are not seen in training set. This work achieved 79.40% accuracy on matched and 80.02% on mismatched set. Similar works that used ESIM model reported 72.4% matched and 71.9% mismatched accuracy while Bert fine tuning achieved 86%. New solution achieves SOTA results by combining contextualized word embeddings with word knowledge. Result that are obtained using ANLI is not as promising as other NLI corpora results. ANLI is constructed with the idea of fooling the best trained model thus, ANLI solutions requires background information and more complex approaches.

CHAPTER 2

RELATED WORK

In this work, literature search is conducted by following the chronological order. Hence it is assumed that following a chronological order would give a better understanding of the work that is done on this domain and show the improvements over time.

Sanda Harabagiu et. al. proposed a framework to detect negations, contrast, and contradiction in text using lexical information with machine learning algorithms. Their framework relies on negation, antonym, and semantic – pragmatic information to recognize contradiction on text inputs. In other words, contrast discourse relation of the text inputs. To generalize the problem and provide a solution their study used 2006 PASCAL RTE Recognizing Textual Entailment dataset. 2006 RTE entailment dataset consist of 1600 pairs of sentences which are annotated by human experts in the same domain. Their framework can combine preprocessing and removal of negations. Later the antonymy is derived via contrast relations. In their research, WordNet (Miller, 1995) was used to successfully detect antonymy. As a final step, they created a classifier by casting contrast relation into a classification problem. After experimenting several machine learning techniques, they proposed that decision tree is the most suitable one for their solution. Their main research interest area was contradictory sentences, for that they used three different linguistic features as follows, Only Negation, Only Paraphrase, and a combination of negation and paraphrase features. They achieved 75.38% accuracy for textual entailment with pascal RTE dataset (Harabagiu et al., 2006).

It is mentioned that WordNet is used to extract linguistic features and detect antonymy. WordNet is a large and constantly updating lexical database of English. Database contains verbs, adjectives, adverbs and provides synonym – antonym relations among words as a synonym – antonym set (Miller, 1995).

Andrea Esuli and Fabrizio Sebastion introduced SentiWordNet which is widely used in researches that focused on opinionated sentences. SentiWordNet provides a new type of WordNet approach where each WordNet (version 2.0) synset is associated with three numerical scores. These numerical scores describe objective Obj(s), positive Pos(s) and negative Neg(s) terms in the synset. SentiWordNet helps to extract opinions from

opinionated sentences by providing PN-polarity of subjective terms. WordNet also aids the researchers with SO-polarity which helps to decide if given text has factual nature or expressed as an opinion. Another task SentiWordNet provides is helping to determine the strength of PN-polarities such as weak, strong, or mildly. Considering three tasks that SentiWordNet provides, it is clear that it provides a powerful tool for opinion mining (Esuli and Sebastiani, 2006).

Not long after SentiWordNet's release, Stefano Baccianella et. al. released SentiWordNet 3.0. Apart from the minor changes, major improvement conducted in the release of SentiWordNet 3.0 is that new approach constructed using WordNet 3.0 database. Another difference is that SentiWordNet 3.0 uses different approach to annotate WordNet synset. In version 3.0, semi-supervised learning approaches was used the same as version 1.0, but they differ on random-walk step approach. Random-walk step approach used to create SentiWordNet 3.0 approach by viewing WordNet 3.0 dataset as a graph and run an iterative random-walk process on each type of annotation, Pos(s), Neg(s), and Obj(s). New random-walk step approach resulted in improvements on the performance of SentiWordNet by 20% compared to SentiWordNet 1.0 (Baccianella et al., 2010).

It is mentioned that opinion classification tasks use RTE entailment dataset. Third RTE PASCAL challenge brought up a new approach as an extension task to the main task. Ellen M. Voorher, proposed that instead of a two-way entailment task as 'entailment' and 'contradiction', it is required to create a three-way entailment task. Unlike the two-way entailment task, three-way entailment task would have three labels as follows 'contradiction', 'entailment', 'neither'. Three-way entailment task need emerged due to having only 10% contradiction examples in the RTE set and this caused difficulty for systems to detect the correct label for given text pairs. RTE uses ordinary understating principle which assumes if human reading concludes hypothesis were true, then the hypothesis considered to entail with the text. Extended RTE task also uses the same ordinary understanding principle. Three-way task proposed a way to convert the two-way RTE task into the three-way task with the contributions of six NIST human annotators who worked as TREC and DUC assessors. This solution provided an opportunity for detection systems to improve their abilities to detect contradiction (Voorhees, 2008).

Marie-Catherine et. al. proposed that finding contradiction in a text is harder than detecting entailment pairs. Researchers of this work claim that contradictions occur when there are differences in the sentence context such as antonym, negation, numeric mismatches, and word-knowledge. All these features are lexical components of the text

that compose the meaning of the given hypothesis. Researchers of this project claim that it is not enough to assign the correct final decision for non-entailment pairs. Hence, they also include event co-reference into their systems which considers whether they occur in the same event or not. According to their contradiction definition, there are two primary categories for contradiction. The first category is antonym, negation, and date-time mismatch information set which are easy to detect. The second is fictive, modal words, lexical contrast, and word knowledge which they refer as hard to detect features. These features require precise models of sentence meaning for contradiction detection. Even though two given sentences may express the same meaning by having similar words in their grammar, structural differences can indicate the contradiction. Their work uses the RTE datasets to build a solution to the contradiction problem. Their system considers the linguistic analysis by converting hypothesis and text pairs into a dependency graph using the well known word database WordNet (Miller, 1995). After graph conversion, text and hypothesis graphs are aligned to score graph nodes for similarity. When graph scoring is finished, non-co-referential events are filtered, hence words or sentences which do not indicate the same event can be removed. In the final step, extracted features are used to build a logistic regression classifier to determine if the given pair is contradictory or not (de Marneffe et al., 2008).

Alan Ritter et.al carried out the contradiction task by considering lexical information. Apart from its predecessors, this research points out the importance of background knowledge and tries to offer domain independent solution using simple logical function. Research proposes that apparent contradictory sentences are consistent statements composed of meronyms, synonyms, and hypernyms. It also offers a corpus, crawled from the web that is composed of the sentences which are seemingly contradictory. They also claim that their contradiction detection system can discard seemingly contradictory sentences and detect the real contradictions using a logical function. Logical function states that, if two statements contain antonyms, negation and other lexical elements, this indicates that both sentences are contradictory. To detect real contradiction, they include the background knowledge to their environment and form the logical function. Their function, also named AuContraie works as tuples. It decomposes sentences into tuples and builds a function to detect contradictions. The real challenge in this work is to define which of the tuples are relevant ones for the contradiction process. Hence, URNS model (Downey et al., 2005) is used to estimate the probabilities on binomial distribution. Later, factual assertion is used which works with the TextRunner (Banko et al., 2008) method.

Handling the seemingly contradiction statements are conducted by describing synonyms and meronyms using the well-known word vocabulary WordNet. To avoid false positive results, they implemented an argument typing method which uses a named-entity tagger with large dictionaries. Lastly, all the information extracted with previous functions are combined to build logistic regression classifier to solve the contradiction detection problem (Ritter et al., 2008).

Some other research also focuses on opinionated sentences but not directly in the contradiction domain. Arjun Mukherjee and Bing Liu proposed that, their research takes two individual opinionated sentences and classifies them as contentions or agreement. Their work proposes a model named JTE , Joint Topic Expression. JTE is a statistical model that jointly analyses topics and contention-agreement features. In the research, JTE model is later extended to JTE-R and JTE-P which correspond to reply to relations and author-pair structures, respectively. Although they aim to analyze opinions, their project domain is topic mining. JTE is a member of generative models for text and words or phrases. JTE considers the words as n-gram where they are viewed as random variables and documents are considered as bag of words. Their JTE implementation, considers the terms that appeared least thirty times in the document. Their work also encodes the pos-tagging and lexical features which appeared previous, current and the next of the terms. Since it is not possible for JTE to learn exact inference from data, Gibbs sampling (Griffiths and Steyvers, 2004) was used to resort approximate inference. Gibbs sampling is a member of Markov Chain which constructed to have a particular stationary distribution. To evaluate the model, researchers obtained political, religious, scientific text data to discover contention and agreement expression by considering the topic features. According to observations, JTE-R and JTE-P is much better than JTE and JTE-P produces the best result. JTE-P has paired sentences in its corpus and acts as a pair specific model thus gives the best performance (Mukherjee and Liu, 2012).

Kasper Van Veen focused on detecting the contradiction on a specific feature level using antonym and negation features. The question proposed in this research is that what kind of contradictions can be detected between news articles using antonymy and synonymity. Research was later experimented on RTE dataset. Their methodology is based on dependency parser and graph. The aim of this process is to find syntactic structure for each candidate sentence. Instead of using Stanford parser, researcher claimed that Spacy parser is much more efficient than Stanford parser. After obtaining dependency information, each graph is aligned with each other. Alignment phase provides a possibility score

whether each sentence contradicts with each other. Alignment scores are obtained based on antonym and negations. Later, the co-referent sentences are filtered out. Co-referent sentences refer to a sentence pair which do not share the same event. As a last step, logistic regression classifier is built to detect contradictions (Veen, 2016).

Google engineers Ankur P. Parikh et. al. and his colleague's solution to the contradiction problem is different from the rest. Instead of building complex solutions based on lexical information, proposed method uses attention approach to decompose the problem into sub problems. Their objective is to separate complex tasks into sub problems, solve the problems separately and later parallelize the subtasks. Another benefit of this work is that decomposing approach achieves state of the art result with fewer parameters. It, enables solving complex inference task in a lightweight way. Achieving scalable decomposability relies on alignment approach. Decomposition consists of two stages. First stage creates alignment matrix. Alignment matrix is created by attention method using word embeddings. Second stage uses soft alignment approach to solve given tasks separately. After alignment stage, in the final step, subtasks are merged to form a classification layer. Decomposable attention approach uses attentions completely based on word embeddings; hence, it provides language inference solution regardless of the word order. Word embedding based attention shows that, pairwise word information comparison is more effective than sentence level comparison (Parikh et al., 2016).

Conflict analysis is another subject of sentiment analysis that focuses on detecting disputes around a topic. Unlike contradiction detection, conflict analysis takes the subject as a sentiment analysis problem and classifies given text inputs into positive and negative rather than entailment or contradiction. According to Adem Karahoca et. al. the more time debate continues, it would become harder to analyze and solve the issue, thus they propose a solution to detect and classify the disputes. Their work takes the structure and word knowledge of the SentiWordNet and translate its structure to Turkish language to propose a new dictionary. Their main research objective is to provide a solution on Turkish language disputes, yet their solution can be scaled to other languages easily by converting SentiWordNet to other languages (Karahoca et al., 2017).

Since word embedding breakthrough, majority of the researches are based on embedding methods. Word embedding states that words can be mapped into three-dimensional space according to their appearance and relations with other words in a large corpus. Thus, enables researchers to turn words into better mathematical expression to build a robust solutions Luyang Li et. al. claims that using word embeddings is not

enough to build robust solution. According to their research, contrasting words such as ‘overfull’ and ‘empty’ are mostly mapped into same vector space. Their work proposes to build a hand tailored neural network that can learn contradiction-specific word embeddings. Results show that classifiers that are built with tailored contradiction-specific word embeddings outperforms traditional word embeddings (Li et al., 2017).

Siti Nuradillah Azman et. al. focuses on detecting the contradictions on online reviews, especially hotel reviews. Rather than proposing a new solution method, aim of this work is to detect contradictions on a real-life scenario. Theory is, mining online reviews to detect contradictions could improve the customer satisfaction. In order to successfully detect contradictions, proposed solution is based on mining lexical information from sentences such as numerical mismatched. Proposed method tokenizes the given sentences and pos tags each extracted token. Later, named entity recognition is used and dependency graph is created. In feature selection stage, syntactic analysis and aspect detection is applied to the given dependency graph. Proposed method shows the importance of the numerical mismatch-based contradiction detection on online reviews (Azman et al., 2017).

With the introduction of SNLI (Bowman et al., 2015), it is more feasible to train language inference classification solutions with low neural net complexity. This research exposes the advantage of the large, human annotated SNLI inference corpus. It is claimed that sequential chain LSTM models can improve overall performance. Thus, their solution is based on using chain LSTM neural network architecture with large SNLI inference data. Language Inference solution in this work uses BiLSTM memory units. BiLSTM architecture takes an input and processes the information forward and backward in time. Proposed neural network solution includes input encoding, local inference, and inference composition. In addition to given solution, syntactic parse information tree based LSTM is also presented. Results show that using chain LSTM based solution with large SNLI corpus is much more powerful than its predecessors (Chen et al., 2017).

Ismail Badache’s work suggests that, finding the polarity of the contradiction may help to better understand the origin of the dispute instead of just classifying. This research aims to find contradiction intensity around specific topic or aspect, using the sentiment analysis methods. Their solution extracts aspect characteristics of the reviews and uses sentiment analysis to capture opposing opinions of aspects. Later, applying a dispersion based measure on user reviews allows detection system to extract contradiction intensities and polarities. Dataset which is collected from Coursera is used for evaluation of the

proposed method. Their work can be explained in two ways. A way to estimate contradiction intensities and whether there is an impact when jointly considering polarity and the intensity of the contradiction (Badache et al., 2018).

Vijay Lingam et. al.'s work uses Bidirectional LSTM neural network method with Glove word embeddings. Proposed method focuses on finding the contradiction types such as negation, antonyms, numeric mismatch. First, proposed method extracts textual features such as negation and antonym. Later, neural network solution is applied using the extracted features with glove word embeddings. Their method uses three different features to build classification model. First method extracts and uses manual features to build a classifier. Second method only uses LSTM based features. As a last solution both manually extracted, and LSTM based features are combined to build a robust classifier. Proposed method tested on Stanford, SemEval and Pheme datasets. Except the combination of manual and LSTM feature classifier, other two solutions provide promising results (Lingam et al., 2018).

Shoreh Haddadan, Elena Cabrio and Serena Villata approach to contradiction detection problem in a different way. Their work's aim is to conclude an argument mining on political data. In their claim, no other research is worked on large corpus of political data. 39 political debate over 50 years of time span is crawled for this research. Crawled corpus is human annotated as premise and claim. Problem is identifying argument components in debate datasets and classifying them as claim – premise. Instead of contradiction – entailment, research approaches to problem as attack and support relation between claim and premise. Proposed solution handles the input texts on a sentence level in order to accomplish argumentative sentence detection and component identification. Two classifier methods, support vector machines and LSTM are used to build classifier for political argument mining task. SVM constructed with linear kernel and stochastic gradient descent and LSTM used as bidirectional to preserve forward and backward information. Their work utilized Fasttext word embeddings for word representations. Results show that, SVM with linear kernel outperforms the other methods by nine percent. This research is novel in a way that they proposed large human annotated political debate data for inference problem (Haddadan et al., 2019).

Recent state of the art solution came from Google AI team. Bidirectional Encoder Representations from Transformers shortly BERT. Bert is the new AI breakthrough that can handle majority of the natural language processing tasks alone. Research states that current pretrained transformers are lack of using bidirectionality. This restricts the power

of bidirectional information. BERT overcomes unidirectional restriction by using MLM short of Masked Language Model. Masked language model enables transformers to use left and right context. Bert transformer logic works as follows. In all layers of transformer, features are jointly conditioned on the left and right directions, hence it is pretrained using bidirectional information on unlabeled data. This approach lets BERT to be the first transformer that can represent both sentence and token level representation. To our knowledge Bert has achieved state of the art results on eleven natural language processing tasks. Its structure enables fine tuning with labeled data. Fine tuning trained model can achieve state of the art results with few epochs and fewer parameters (Devlin et al., 2019).

The technological advancement through years enabled people to have access to online social media platforms. With the easy access to social media, people around the world started sharing their opinions and their daily life problems on the internet. Real life opinionated data can contain reviews, emotional expression of an event, or simple thoughts. These types of data contain high importance and when it is mined with appropriate methods, results can shed light on solutions of common problems. Language inference is a tool to mine opinionated data. In related research, it is shown that language inference problem is widely studied in the natural language processing problem domain. With the release of human annotated RTE dataset, language inference research area gained so much importance. Variety of solutions are proposed to classify whether given two texts or sentences entail or contradict with each other.

Early-stage solutions approached to the problem as classifying entailment or contradiction. These approaches handle the inference problems as binary classification task. As the research value on language inference increased, it is emerged that binary classification is not sufficient. Later, it is proposed that instead of binary label, three-way classification should be used. First three-way entailment task is introduced as an extension of RTE challenge. Three-way classification task includes entailment, contradiction, and neutral classes. Three-way approach states that when a pair of two texts do not contradict nor entail, then the relation must be neutral which states that there is no context relation between given texts. To conduct an analysis between two opinionated sentences, appropriate solutions must be used to convert texts into meaningful representations. Conversion process is named as feature extraction. Extracted features are used to build a classifier so that inference data can be classified. In the early stages of language inference research area, solutions are usually based on language structure and text features such as negations, antonym, synonym, meronym, and date-time mismatches.

As the research area is expanded through years, in natural language processing dense embeddings are constructed for words. Newly introduced representation technique handles the feature extraction problem in a way where structure of the text can be preserved and represented with numerical features. It is shown that, dense vector representations of text inputs are much superior to its predecessors. After dense vector representation gaining so much importance and attention, majority of the language inference solutions are conducted based on dense weights approach. Major downside of the text embeddings is that it cannot preserve the meaning of the same or similar word in different context. In different contexts, the same word can carry two different meanings. Conventional vector representation methods are based on a lookup table structure which assigns the same vector to a word regardless of the context.

2018 was the year of transformer (Vaswani et al., 2017) networks. BERT is a language model based on transformers. Newest approach transformer networks are able to represent text inputs as vector representations preserving the context and the meaning of words in a sentence. It is known that a word can carry two different meanings in a sentence depending on the position and context in the sentence. Transformer networks learn from the context so that they can represent the word information without distorting the meaning of the word. Other transformer network such as ELMO, independently uses LSTM networks and concatenates the outputs. Elmo processes the left-to-right and right-to-left information separately and concatenates the LSTM outputs to achieve bidirectional LSTM approach. This approach creates some performance issues and lacks complete bidirectional network base. Unlike Elmo, BERT achieves complete bidirectional information processing approach using masked language model, MLM. Complete bidirectional base is achieved by jointly conditioning the representation on both left-to-right and right-to-left in all layers. Thus, provides the best results for feature representation. To our knowledge, BERT holds the state-of-the-art results in eleven language processing tasks.

Related works can be divided into three subdivision based on their feature extraction methods. Prior models were using feature extraction such as antonymy, synonymy, meronymy, and date time mismatch. Those feature extraction methods were highly dependent on WordNet dictionary and SentiWordNet representations. Later, dense vector representations are introduced. Their downside is that they are not able to preserve meaning when representing the words. The latest solution is the Transformer encoder networks. Transformer networks are pretrained networks that represent features preserving the context and the meaning.

CHAPTER 3

RESEARCH CORPORA, FEATURE REPRESENTERS AND ALGORITHMS

It is hypothesized that using transformer encoder contextualized word representations with deep learning classifiers would prove the best and robust solution. It is also discussed that variety of the natural language inference corpora are exist and free to use for research purposes. This chapter widely discusses natural inference corpora, feature extractors and proposed deep learning method architectures.

3.1. Natural Language Inference Corpus

Sentence understanding is a valuable task for natural language inference. Having an understanding in language inference enables to interpret sentence relation in a matter of contradiction and entailment. However, scaling this task to machine level is challenging to apply. Current machine learning and deep learning algorithms rely on density of the train data. Lack of ground truth data makes this task is less possible to apply. Hence, natural inference corpora are constructed to solve this problem

3.1.1. Stanford Natural Language Inference Corpus

Table 3.1. SNLI corpus sample (Source: Bowman et al., 2015)

A black race car starts up in front of a crowd of people.	**contradiction**	A man is driving down a lonely road.
An older and younger man smiling.	**neutral**	Two men are smiling and laughing at the cats playing on the floor.
A soccer game with multiple males playing.	**neutral**	A happy woman in a fairy costume holds an umbrella.

Stanford natural language processing group states that, existing natural language inference corpora are not sufficient to provide reliable solution. Existing corpora are small and do not expose the true power of data driven neural net based methods. There are some inference corpora that can reach large amounts, but they are algorithmically computer generated and create ambiguity. To provide a solution for the emerged problem, Stanford natural language group introduces SNLI corpus (Bowman et al., 2015). Stanford natural inference corpus is a collection of sentence pairs and labels created for natural inference research purposes. In contrast to other corpora, SNLI is completely human written and labels are human annotated thus provides ground truth. New corpus provides 570k sentence pairs, with that it achieves the twice the magnitude of data compared to priors. SNLI corpus helps to outperform the conventional lexical based natural language inference solutions and it opens a way to expose true power of neural network architectures. This work also creates the first benchmark of language inference problem domain. SNLI corpus is also eligible to be used as evaluation corpus for rule based, simple linear classification and neural network-based approaches.

To this day primary resource for language inference problems was RTE corpus. Same as SNLI, RTE dataset is also hand crafted and human annotated. Major deficiency of RTE corpus is that it is too small compared to SNLI. This deficiency limits the power of learning-based approaches. The other corpus is SICK Sentence Involving Compositional Knowledge relatively larger than RTE but still not efficient enough compared to SNLI. Although there are other corpora that is superior to SNLI in terms of size. Denotation Graph Entailment Set is relatively larger than SNLI, but this corpus contains noisy examples, and it is labeled with automatic systems. Hence, it is suitable to be used as supplementary corpus.

Forming SNLI corpus requires a special framework that uses Mechanical Turk infrastructure. Common works suffer from indeterminacy due to having single label. To prevent indeterminacy, Stanford language group created crowd-sourcing framework. Framework states that examples must be grounded to specific scenario and premises – hypothesis pairs must be derived from specified scenario. This approach helps to control event and entity co-reference. Secondly framework lets participants to create their own sentences in accordance with the rules. Creating novel sentence contributes to corpus by providing wider example space. Framework gives premises in batches of five and asks participants to create three hypotheses for each premise. hypotheses must include one true answer, and false answer and one answer that can be both true and false, this corresponds

to entailment, contradiction, and neutral labels. Later these examples are sent to human annotators to achieve ground truth labels. Labeling phase uses same Mechanical Turk infrastructure as used for sentence creation. Framework gives premise and hypothesis to annotators in batches and asks annotators to label the pairs. This process repeated for four more annotators and provides five labels in total. Later framework assigns the gold label based on consensus such as having an agreement three out of five labels. Labeling phase conducted having three out of five consensuses for 98% of total data and five out of five consensuses for 58% of the data. Overall agreement states that SNLI is highly sufficient and ground truth language inference corpus that can expose the true power of data driven learning-based approaches (Bowman et al., 2015).

Table 3.2. SNLI corpus key statistics

Data Set Size		**Sentence Length**	
Training Pairs	550,152	Premises mean token count	14.1
Development Pairs	10,000	Hypotheses mean token count	8.3
Test Pairs	10,000		

3.1.2. Multi Natural Language Inference Corpus

Table 3.3. MultiNLI corpus sample (Source: Williams et al., 2018)

Fiction: The Old One always comforted Ca'daan, except today.	**neutral**	Ca'daan knew the Old One very well.
Telephone Speech: Yes, now you know if if everybody like in August when everybody's on vacation or something, we can dress a little more casual or...	**contradiction**	August is a black out month for vacations in the company.
9 / 11 Report: At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	**entailment**	People formed a line at the end of Pennsylvania Avenue.

MNLI is one of the largest natural language inference corpora contains 433.000 sentence pairs available for free use. Unlike SNLI which used specific genre, MNLI is

consist of ten different genres. Each genre is composed of both written and verbal English sentences. Composing different genres into one corpus approach enables to challenge learning-based classifiers. Even though SNLI corpus already introduced to language inference research domain, its sentences are derived from flicrk30 dataset. Sentence pairs are based on image captioning and visual scene interpretations. This derivation creates relatively shorter premise sentences. SNLI slightly damages the tense, belief, and modality therefore existing corpus cannot present challenging evaluation benchmark. MultiNLI addresses these problems and introduces new corpus. The challenge MultiNLI focuses on is to remove these limitations by composing different genres and provides better benchmark for ambitious language inference researchers (Williams et al., 2018). Data collection of MultiNLI uses the same footsteps that SNLI used. MultiNLI selects premises from pre-existing text sources and uses human participants to compose novel hypothesis for each premise. Premise text sources come from ten different genres and nine of these genres comes from Open American National Corpus. These genres are defined as follows.

Face-to-face: Compositions of two-sided real human conversation.

Telephone: Compositions of two-sided telephone calls held in 1990-1991. Examples are composed of regardless of the gender and region.

9/11: Compositions of report calls collected the attack of 9/11.

Travel: Compositions of travel reviews and discussions released by Berlitz Publishing.

Letters: Compositions of letters from fundraising by of non-profit organizations. Released by Indiana Center of Intercultural Communication of Philanthropic Fundraising Discourse.

OUP: Compositions of non-fiction works by Oxford University.

Slate: Compositions of articles derived from Slate Magazine.

Verbatim: Compositions of articles from magazine containing short sentences.

Government: Compositions of government domain sentences and reports.

Fiction: Compositions of fiction writings from 1912 to 2010.

Premise sentences are constructed around genres that are stated above. In order to provide better corpus, relatively short sentences are pruned, and non-narrative pairs are removed. It is also stated that none of the examples in SNLI are allowed in MultiNLI corpus. Hypothesis sentence creation process works same as SNLI corpus. Both corpora use Mechanical Turk infrastructure to compose hypothesis using real human annotators. To preserve the balance, each annotator asked to create a novel hypothesis for each label contradiction, entailment and neutral. MultiNLI also provides better evaluation technique

by creating of two different test sets, matched and mismatched test set. Matched test set contains examples from train set genres. Unlike matched, mismatched set contains only examples that are not seen in train set.

Validation process inherits the same way SICK and SNLI corpora used. Each annotator is presented with sentence pairs and asked to assign single label. Each pair presented to four different annotators thus creates five labels in total. Using consensus logic, based on most of the votes, each pair assigned with gold label. MultiNLI addresses the weak sides of SNLI and creates new, more challenging corpus composed of different genres (Williams et al., 2018).

Table 3.4. MultiNLI corpus key statistics

Data Set Size		**Sentence Length**	
Training Pairs	433,000	Premises mean token count	19.9
Development Pairs	10,000	Hypotheses mean token count	10.1
Test Pairs	10,000		

3.1.3. Adversarial NLI: A New Benchmark for Natural Language Understanding Corpus

Table 3.5. ANLI corpus sample (Source: Nie et al., 2020)

Will Vodery (October 8, 1885 - November 18, 1951) was an African-American composer, conductor, orchestrator, and arranger, and one of the few black Americans of his time to make a name for himself as a composer on Broadway, working largely for Florenz Ziegfeld	**neutral**	Will Vodery wrote Ziegfeld's first song
Abesim is a town in Sunyani Municipal District in the Brong-Ahafo Region of Ghana. Abesim is very close to the regional capital town of the Brong-Ahafo Region, Sunyani. Abesim is known for the St. James Seminary and Secondary School. It is also known for the Olistar Senior High School. The school is a second cycle institution	**entailment**	Abesim is known to be in Sunyani

According to Facebook AI team, current trend in natural language understanding tasks is shifted to combined benchmarks. Combined benchmarks can evaluate the learning-based model performance on multiple tasks. Combined benchmarks can provide unified platform for analysis. In 15 years, AI researches achieved near human performance. Even after combining benchmarks, the rapid advancements in AI makes current benchmarks obsolete. Rapid advancements emerge new approaches to create more robust and challenging benchmarks. This work raises the question that, is it possible to create new benchmark can last longer. Moreover, rapid consumption of benchmarks raises another question. Are the current models as good as their performance say or are they just exploiting the patterns inside the current benchmarks. This can be interpreted as, whether current models just exploring the structural patterns inside the benchmarks or do they really learn the underlying meaning of the given sentences.

This work proposes a new approach, adversarial *human and model in the loop solution*. Human in the loop process works as testing an annotator created example on current best NLI model. After prediction, misclassified examples are collected and verified by another human annotator to be sure if the given example is good enough to be considered as language inference example. If annotators are agreeing on verification, these challenging examples are added to new training corpus. New corpus is then trained to create new model. Same procedure continues for three rounds thus creates A1, A2 and A3 sets. Each set is more challenging and contains more example than previous one.

Annotation phase is conducted using same Mechanical Turk infrastructure. In addition, ParlAI is utilized to collect the hypotheses. Annotators are presented with target label and a context which corresponds to premise in previous NLI corpora. Later, annotators are asked to write hypothesis. These pairs are used on a best NLI model and resulting labels are presented to annotator. If labels are incorrect, process continues if label is correct, annotator is asked to provide new hypothesis. Adversarial NLI project uses $\text{BERT}_{\text{Large}}$ that trained on combination of SNLI and MNLI as their base model. Later in each round new model is trained to create more challenging examples. Thus, creates a much more strong, robust, and last longing benchmark.

It is the first research that introduces human – model in the loop corpus. Corpus is created using three rounds and each round is published separately as A1, A2, A3. Combination of these three rounds forms the Adversarial NLI dataset which reaches up to 162.000 examples. ANLI contains less dev and test examples. Major difference is that, unlike previous NLI sets ANLI has much longer premise (context) sentences. It is

also stated that models trained on ANLI dataset can achieve good results on other NLI corpora. New dataset can also shed light to current corpora and model's weaknesses (Nie et al., 2020).

Table 3.6. ANLI corpus key statistics

Data Set Size		**Sentence Length**	
Training Pairs	162.865	Premises mean token count	54.11
Development Pairs	3200	Hypotheses mean token count	9.70
Test Pairs	3200		

3.2. Textual Feature Representations for Learning Based Algorithms

In order to carry out learning based natural language processing tasks, textual features must be represented as numerical features. Through years many representation techniques are introduced to natural language processing research area. This work uses the most preferred textual feature extraction techniques such as Word2Vec, Glove and Fasttext. In addition to classical dense word vector representations, this work mainly focuses on pretrained transformer encoder model BERT.

In 2013 Tomas Mikolov et. al. introduced the word2vec approach that represents words with numerical values using skip-gram. Later, Glove pretrained word embeddings are introduced addressing the limitation of Mikolov's approach and proposed better solution. Their solution achieved better results based Mikolov's analogical evaluation method. Not long after, Mikolov introduced the Fasttext that focuses on performance improvements on pretrained word weights and achieved better results than its predecessors. For a long time, pretrained word weights dominated the natural language inference research area and provided good scores. In 2018 Google AI research group introduced BERT. Bert exposes the true power of bidirectionality and implements this approach to transformer encoder. Thus, opens new horizons for feature representation. Unlike previous methods Bert learns from the context and provides contextualized word embeddings. This approach removes the limitation of pretrained word weights that usually damages the context. Bert also introduces better fine-tuning solutions and lets researchers to use its architecture for their specific problem domains.

It is known that there are other feature representator such as Doc2Vec which rep-

resents sequences or paragraphs with fixed size vector. Doc2Vec uses the word vectors and takes their weighted mean to provide fixed size embeddings. In most cases this approach provides good results but for this work's problem domain this solution damages the information in sentences in a dire way. Another approach on transformer encoders is ELMO . ELMO provides contextualized word embeddings using character-based embeddings. In most cases this approach provides near state-of-the-art results but unlike Bert, Elmo suffers from shallowly bidirectional architecture. This research aims to provide best results therefore Elmo embeddings are not attended. Also, ELMO it is widely used for many language processing tasks and it is not reasonable to re-produce same results again.

3.2.1. Word2Vec: Distributed Representations of Words and Phrases

It is hypothesized that distributed representations of words would help learning based algorithms to achieve better performance. In 2013, Mikolov et. al. introduced a skip-gram approach that learns vector representations from unlabeled and unstructured text data. The advantage of skip-gram approach, this method does not involve dense matrix multiplications. Thus, enables great amount of time complexity improvements. It is stated in the research that optimized single machine can train up to 100 billion of tokens in a day. Their research also extends skip-gram approach and implements sub-sampling of frequent words. Sub-sample implementation during training increases the speed of learning process by the magnitude of 2x – 10x. Sub-sampling also improves the accuracy of non-frequent words representation. Task of the skip gram model is to determine word representations which are useful to predict surrounding words. In a formal way, given training words w_1 w_2 w_3 ... w_t task is to maximize average log probability.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{\mathrm{t+j}}|W_{\mathrm{t}}) \tag{3.1}$$

Equation 3.1 states that, objective of the Skip-gram model is to maximize the average log probability

In the equation c represents the training context size. It can be hypothesized that larger c can provide better training results. Another limitation is that, representing the word phrases is not feasible and limited due to the architecture of skip-gram. Thus, makes skip-gram much more computationally expensive to represent word phrases compared to

single word vectors. Proposed extension to represent word phrases is rather simple. Word phrases are identified using data-driven approach, then each phrase is considered as single token during training.

Simple evaluation method such as measuring accuracy using test and developer sets is not suitable for skip-gram based word representation approaches. Therefore, new evaluation metric was necessary. To solve this problem analogical reasoning task method is used to create new test dataset and evaluate the trained model. Example below is derived from the original paper to describe analogical test set. It is assumed test set has following tokens, *"Montreal"*, *"Montreal Canadiens"*,*"Toronto"*, *"Toronto Maple Leaf"*. By taking their nearest vector representations, following process is applied.
Vec(*"Montreal Canadiens"*) − Vec(*"Montreal"*) + Vec(*"Toronto"*) must be equal to Vec(*"Toronto Maple Leaf"*).
If resulting answer meets the expectation, it indicates that trained model works correctly. This also shows another improvement on skip-gram based vector representations. Inspecting the example, it can be said that simple mathematical operations can be done using word representations. This leads to better vector coverage for longer sequences linear.

To train skip gram model, large corpus of news articles is used. Word that are seen less than five considered as not important and discarded from the train set. Thus, leads to 692.000 vocabulary size. Later, to learn vector representations, words that are frequently seen together are identified. To reduce complexity instead of complex n-grams, this work uses uni-gram and bi-gram approach. As a last step, trained data iterated through two to four times. Evaluation is conducted based on analogical method as described above. This work shows that learning vector and phrase representations using skip-gram model provides better representations of tokens. Training step repeated with several magnitudes of data and it is proved that with larger training data it is possible to obtain better result. Sub-sampling of frequent words improves the training time and the representations of non-frequent words. Word2Vec method achieved 72% performance with 33 billion training word corpus. To justify the relation between training data and performance, another model trained on 6-billion-word corpus resulting 66% performance. This proves the linear relation between word corpus and performance (Mikolov et al., 2013).

3.2.2. Glove: Global Vectors for Word Representations

As it is discussed in Mikolov et. al.'s research, representing a word with a real number is main task of semantic vector space models. Usually, word representation heavily dependent on token distance or angle between token pairs. This representation of words is widely used in natural language processing area. Also, Mikolov's word analogy evaluation approach enabled to benchmark vector space models. This new benchmark uses analogical reasoning instead of measuring scalar distance between words, thus provides better measurement.

There are two main approaches in vector space model family, first is global matrix factorization such as LSA and second is skip-gram model which Mikolov's model is based on. With the development of analogical evaluation method, it is known that algorithms like LSA performs well on statistical information but performs poorly on word analogical tasks. On the other hand, skip-gram based methods work well with analogical tasks, but they cannot utilize the statistics of the given corpus. Glove proposes specific weighted least squares model. Proposed model efficiently uses statistics when training global words to word co-occurrences.

Their work suggests that, starting point of an unsupervised word vector learning approach should be ratios of co-occurrences rather than the possibilities of the words. This is justified in their work by comparing raw probabilities of a word with its ratio. It is stated that ratio is better starting point. Glove states that using new weighted least squared regression model can remove the drawback that occurs due to weighting all co-occurrences evenly. Evenly distributed co-occurrences are noisy and have a possibility to damage the overall evaluation. This work states that noisy data carry less information and it occurs 75% - 95% of the data considering zero entries. This ratio may change according to vocabulary size. Their new weighting function formulated as follow.

$$J = \sum_{i,j=1}^{v} f(X_{\mathrm{ij}})(w_i^T \widetilde{w}_{\mathrm{j}} + b_{\mathrm{i}} + \widetilde{b}_{\mathrm{j}} - \log X_{\mathrm{ij}})^2 \tag{3.2}$$

Equation 3.2 states that, introducing a weighting function $f(X_{\mathrm{ij}})$ into the cost function gives the model where V is the vocabulary.

V represents the vocabulary size. Without delving deep with the mathematical details to preserve simplicity. This new weighted function needs to meet to following properties.

1. $F(0) = 0$. If f is considered as continuous function. It is expected to neutralize $x \longrightarrow 0$ statements.

2. To prevent over-weighting of rare co-occurrences, $f(x)$ should be non-decreasing.

3. To prevent over-weighting of the frequent co-occurrences for large values $f(x)$ should be small.

Glove is trained on several word corpora, such as *1B – 1.6B – 4.3B – 6B* and the largest one is composed of 42 billion tokens. Glove model can be easily trained on 42 billion tokens with some performance tweaking. Research states that corpus size and model quality is not guaranteed to be linear, for some cases corpus size may affect quality negatively. Evaluation of Glove word embeddings takes place in several benchmark methods. Glove is evaluated with 100 and 300 dimensional variations on same analogical method that Mikolov et. al. introduced. Also, Glove evaluated with word similarity and named entity recognition methods. For most of tasks, glove outperformed the previous methods and determined the new state-of-the-art (Pennington et al., 2014).

3.2.3. Fasttext: Advances in Pre-Training Distributed Word Representation

Each research addresses to its predecessors by identifying its limits and proposes a solution. Each proposed solution leads a new word weights corpus that outperforms the previous one. Majority of the Natural language processing researcher prefers to use pretrained word weights instead of training from scratch. By doing so, researchers demand the best word representer they can find. Over the years general approach was to train model either based on continuous bag of words or skip-gram model combined with log-bilinear. In this research Mikolov et. al. states that current word weights can be improved by simple pre-processing methods that are not discovered well enough. This work mainly focuses on implementing word sub sampling, phrase representations and sub-word information.

Standard continuous bag of word approach designed to learn word weights by predicting the word according to its context. Context can be defined as a window containing surrounding words. More formally, with a given sentence T that composed of words w_1 w_2 w_3 ... w_t task is to maximize log-likelihood.

$$\sum_{t=1}^{T} \sum_{c \in C_{\mathrm{t}}} \log p(w_{\mathrm{t}} | C_{\mathrm{t}}) \tag{3.3}$$

Equation 3.3 states that aim of the equation is to maximize log likelihood of sequence of words w_t in the context C_t

In the formula, C_t represents the context of the $t - th$ word. In the research it is assumed that scoring function between word is accessible and denoted as $s(w, C)$. Considering the knowledge that is provided, conditional probability is the softmax function applied to context and the word in vocabulary. It is stated that this approach is not practical for large corpora. To address that problem, proposed solution is replacing the binary classification over words so that correct word can be learned with sampled negative candidates.

$$\sum_{t=1}^{T} [\log(1 + e^{-s(w_t, C_t)}) + \sum_{n \in N_{C_t}} \log(1 + e^{s(n, C_t)})] \tag{3.4}$$

Equation 3.4 states that, negative log likelihood can be achieved using binary logistic loss for a context position c

Proposed solution simply changes the word probability and N_c represents the sampled negative examples. Replacing the log probability provides maximized objective function.

Other improvements that Mikolov et. al. attended is that word sub-sampling. Considering all occurrences of tokens evenly would cause an overfitting problem for most frequent words and underfit for less frequents ones. ZipF distribution states that majority of the words belong to small subset of given corpus. Mikolov et. al. introduced this strategy in his research (Mikolov et al., 2013). Phrase representation denotes to an expression carries a meaning with more than one word. It is proposed that using n-gram approach can solve the representation problem. Major disadvantage is using n-gram is that it can increase training complexity. Mikolov et. al. proposes to select n-grams iteratively, later these n-gram are merged in to one word in pre-process phase. As for sub word information, it is stated that current word vector models ignore the internal structure of a word. In most cases internal structure carries high value information in a form of misspelled or rare word. This situation usually happens in morphological rich languages such as Turkish or Finnish. Proposed solution is to enriching word weights using character n-grams. This is achieved by decomposing each word into its character n-grams and representing each n-gram with vector. The word vector is simply the sum of total n-grams. This assumes that

out of vocabulary words can be represented using the n-gram co-occurrence matrix based on character n-grams. This research uses variety of the training data such as wiki dumps, news data and common crawl. The largest corpus Fasttext trained on is common crawl corpus containing 630 billion words. Fasttext is evaluated on word analogy, rare words, and squad dataset. Results shows that new Fasttext pretrained word weights outperforms the previous Glove word vectors (Mikolov et al., 2018).

3.2.4. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding

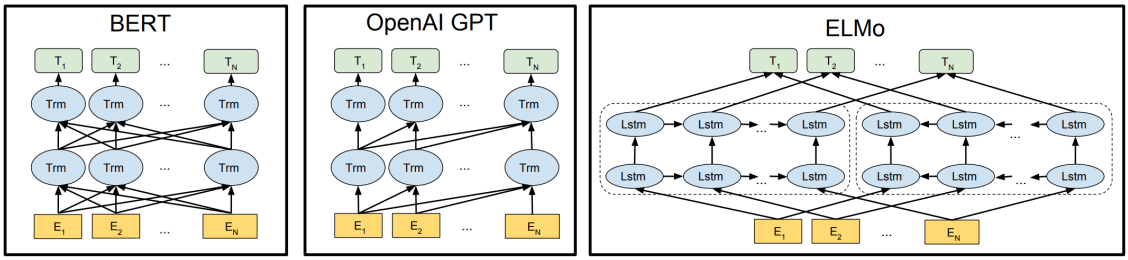

Figure 3.1. Structural Comparison between BERT and other transformers

The newest breakthrough in the natural language processing research area is BERT. BERT, single handily can handle eleven language processing tasks. This section focuses on BERT infrastructure and its capabilities over wide angle of language tasks. Bert infrastructure is built on pre-trained deep bidirectional representation using unlabeled text corpus. In addition, Bert uses joint condition to learn from left-to-right and right-to-left representations in all layers. There are two main approaches for pre-trained language representations, *feature based and fine tuning*. A great example to feature based approach is another transformer encoder mode ELMO. Elmo is another pre-trained representer network that uses task-specific approach which includes pre-trained representations as additional feature. On the other hand, fine tuning can be exampled with GPT from OpenAI. GPT uses minimal task-specific parameter and trained with fine tuning all parameters. These two tasks may seem different, yet they share common objective function during pre-training. They both suffer from unidirectional pre-training approach. Unidirectional approaches are considered to limit the power of transformer networks.

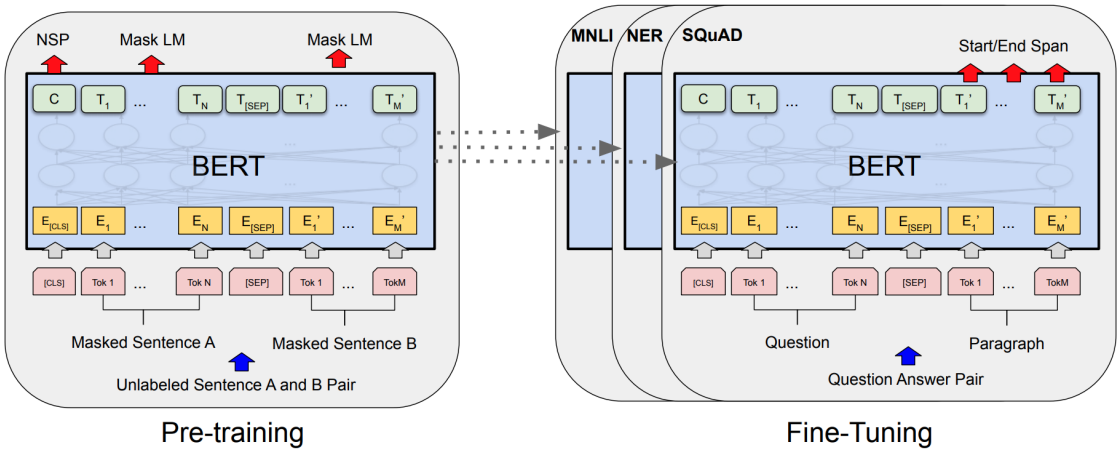

Figure 3.2. Bert pre-training and fine-tuning representation

Newly introduced BERT overcomes this limitation with Masked Language Model MLM. Using MLM enables Bert to use the left-to-right and right-to-left information while training and expose true power of bidirectionality. It is mentioned that there are two approaches in transformer networks, pre-training, and fine-tuning. Fine Tuning can generalize majority of the down-stream task using labeled training set. For fine tuning approach, Bert initializes all pretrained parameters, thus, enables to train desired solutions with small amount of work.

Bert's model architecture is based on multi-layer perspective. All layers are adjusted to benefit from bidirectionality using MLM. As a terminology Bert uses L to denote layer size, H for hidden state and A for attention heads. Model comes with two different variants as out of box, $\text{BERT}_{\text{Base}}$ and $\text{BERT}_{\text{Large}}$. Base model is consisting of 12-layer, 768 hidden state and 12 attention heads and large model comes with 24 layers, 1024 hidden state and 16 attention heads. Bert base is constructed to achieve comparable result with GPT. Major difference between $\text{BERT}_{\text{Base}}$ and GPT is that while GPT using constrained self-attention, Bert uses jointly conditioned attention.

Unlike conventional approaches, Bert implements different input and output representations to handle variety of downstream tasks. This work also introduces new tokenization process named WordPiece tokenizer. Unlike previous works which simply tokenize using white-spaces or requires large vocabularies. Bert uses 30,522 vocabulary to tokenize given sentences. Word piece tokenizer comes with variety of options, but best approach is Full WordPiece tokenizer which decomposes words to its prefix and postfix and tokenize using this information. This approach achieves better result using only small

amount of information. Each sequence input (whether single or sentence pairs) stars with '[CLS]' token. This token represents the start of the sequence and can be used as a sentence representation. If given input is sentence pairs, each pair would be separated with '[SEP]' token. Later each sentence sequence is represented with token ids, segment ids and position embedding as shown in the figure.

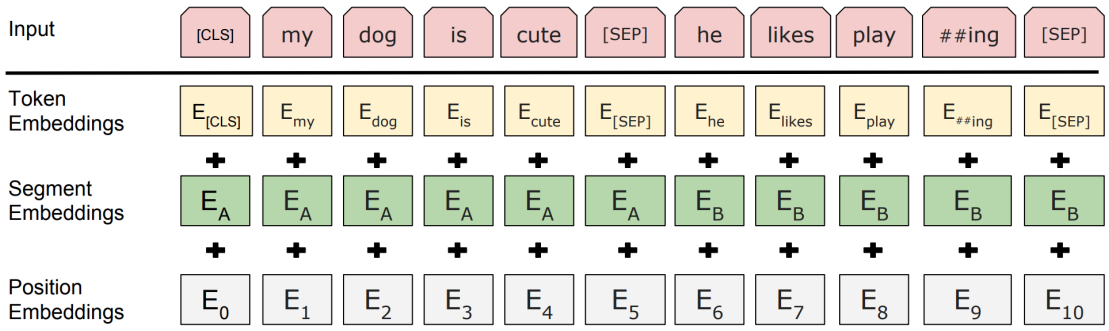

Figure 3.3. Input representation of BERT

It is mentioned that Bert benefits the bidirectionality in its structures by using masked language model. Masked language model simply masks some of the input tokens randomly and tries to predict these tokens. Bert architecture randomly masks the 15% of the word piece tokens for masked language model. Later, final hidden vectors that represents masked tokens are fed into softmax output over the vocabulary. This process is similar for majority of the language models. To prevent noise problem in sequences, Bert only predicts the masked word instead of reconstructing whole sequence.

Pre-training phase conducted using Book Corpus which has 800 million words and English Wikipedia that contains 2.500 million words. As for Wikipedia corpus, only text passages are extracted, and rest of the information is discarded. Bert states that it is crucial to use document level corpus to achieve better result for pre-trained based encoders. Fine tuning phase is simple and straight forward thus enables researchers to use Bert for many downstream tasks. Fine tuning compatible to use with both sentence pairs and single sentences. Bert achieves new state of the art results on eleven natural language tasks such as general language understanding, question answering, and natural language inference (Devlin et al., 2019).

3.3. Language Inference Algorithms

This research aims to expose the power of contextualized word embeddings while providing solution to language inference task. This research implemented two best algorithms on natural language inference task. These algorithms once set new state of the art results on SNLI benchmark. Both algorithms use different approaches. Thus, enables us to compare standard feed forward neural network structure with the bidirectional chain LSTM architecture.

3.3.1. A Decomposable Attention Model for Natural Language Inference

In previous sections it is mentioned that Decomposable attention model uses alignment between inputs and uses feed forward network architecture. Alignment can also be named as attention between given inputs. In 2016, Decomposable attention model set the new state of the art results for SNLI task. In this section this work focuses on underlying architecture of the model. This research defines given two sentence inputs as $a = (a_1, a_2, a_3...a_{la})$ and $b = (b_1, b_2, b_3...b_{lb})$ where l_a and l_b define the sentence lengths respectively. It is assumed that each a_i and b_i is vector representation of tokens named word embeddings with a dimension d. . Using this definition, training data can be represented as $\{a^{(n)}, b^{(n)}, y^{(n)}{}_{n=1}^{n}\}$ and $y_{(n)} = (y_1^{(n)}, y_2^{(n)}, ..., y_c^{(n)})$ denotes the vector encoding of labels where C is the output classes. . Aim is to correctly predict the label for given (a, b) sentence pairs.

Attend: First objective is to soft align the elements $\overline{a}$ and $\overline{b}$ using neural attention. To do so non-normalized attention weight are obtained using function $F^{'}$. function $F^{'}$ denotes a feed forward network with user determined hidden size parameter and uses ReLU activation function. This function can be decomposed as.

$$e_{ij} = F^{'}(\overline{a}_i, \overline{b}_j) := F(\overline{a}_i)^T F(\overline{b}_j) \tag{3.5}$$

Equation 3.5 formalizes dot product of input weights to obtain attention scores

Separately applying $F^{'}$ to sentences $l_a \times l_b$ times would create quadratic complexity. Decomposition method avoids the possible complexity and reduces it to $l_a + l_b$ times.

Later these attention weights are normalized as follows. β_i represents the sub-phrases in $\overline{b}$ which is softly aligned with α_i.

$$\alpha_j := \sum_{i=1}^{l_a} \frac{exp(e_{ij})}{\sum_{k=1}^{l_a} exp(e_{kj})} \overline{a}_i \qquad (3.6) \qquad \beta_i := \sum_{j=1}^{l_b} \frac{exp(e_{ij})}{\sum_{k=1}^{l_b} exp(e_{ik})} \overline{b}_i \qquad (3.7)$$

Equation 3.6 and 3.7 normalizes the attention weights to soft align normalized vectors

Compare: In this step, aligned phrases are compared with each other using another feed forward network G. Feed forward properties are same as$F^{'}$. In the formulation, brackets represent the concatenation process. Having only linear numbers in this stage removes the need of decomposition.

$$v_{1,i} := G([\overline{a}_i, \beta_i]) \qquad \forall i \in [1, ..., l_a] \qquad (3.8)$$

$$v_{2,j} := G([\overline{b}_i, \alpha_i]) \qquad \forall j \in [1, ..., l_b] \qquad (3.9)$$

Equation 3.8 and 3.9 uses feed forward network to compare attended weights

Aggregate: Compare layer provides two sets of comparison can be denoted as $\{v_{1,i}\}_{i=1}^{l_a}$ and $\{v_{2,j}\}_{i=1}^{l_b}$. Aggregation process is conducted by applying summation to each set.

$$v_1 = \sum_{i=1}^{l_a} v_{1,i} \qquad (3.10) \qquad v_2 = \sum_{j=1}^{l_b} v_{2,j} \qquad (3.11)$$

Equation 3.10 and 3.11 conducts aggregation over each set by summation

Obtained summations later fed into feed forward layer named H. $\hat{y} = H([v_1, v_2])$ in here $\hat{y} \in \mathbb{R}$ corresponds to non-normalized scored of predictions. Using argmax we achieve the predicted class, $\hat{y} = argmax_i \hat{y}_i$. Decomposable Attention model implements multi-class cross entropy loss for training phase.

In the Algorithm-1 architecture of Decomposable Attention is demonstrated. a_i and b_i represent the initial word embeddings obtained after embedding layer. Obtained embeddings are then passed trough feed forward layer F. Processed weights $\overline{a}_i$ and $\overline{b}_i$ are dot product-ed to obtain attention weights and normalized. Normalized weights and initial embeddings are dot product-ed to obtain self attention. Self attended and initial weights are concatenated and fed into time distribution layer G. Later sums of each sequence are calculated and concatenated to be fed in to final layer H. In the figure 3.4 initial weights are represented with x_1 and x_2 in the second dot product due to α and a are printed similar.

Algorithm 1 Decomposable Attention

procedure DECOMPOSABLE ATTENTION(a_1,a_2,...,a_n and b_1,b_2,...,b_n)
    m = embedding array of length 64
    for i =1 to m **do**
        $\bar{a}_i$, $\bar{b}_i$ = $F_{\text{Feed\_Forward}}([a_i][b_i])$
        $attention$ = Dot($[\bar{a}_i, \bar{b}_i]$)
        $norm_a$,$norm_b$ = Normalize($[attention_{axis1}][attention_{axis2}]$)
        $alpha$, $beta$ = Dot($[norm_a, a_i]$), Dot($[norm_b, b_i]$)
        $\_x1$ = $\text{TimeDistributed}_{\text{G\_Feed\_Forw}}$(Concatenate($[a_i, beta]$))
        $\_x2$ = $\text{TimeDistributed}_{\text{G\_Feed\_Forw}}$(Concatenate($[b_i, alpha]$))
        $v1\_sum$, $v2\_sum$ = Sum($\_x1$), Sum($\_x2$)
        y = Softmax(Dense($H_{\text{Feed\_Forward}}$(Concatenate($[v1\_sum, v2\_sum]$))))
    end for
    return predicted label
end procedure

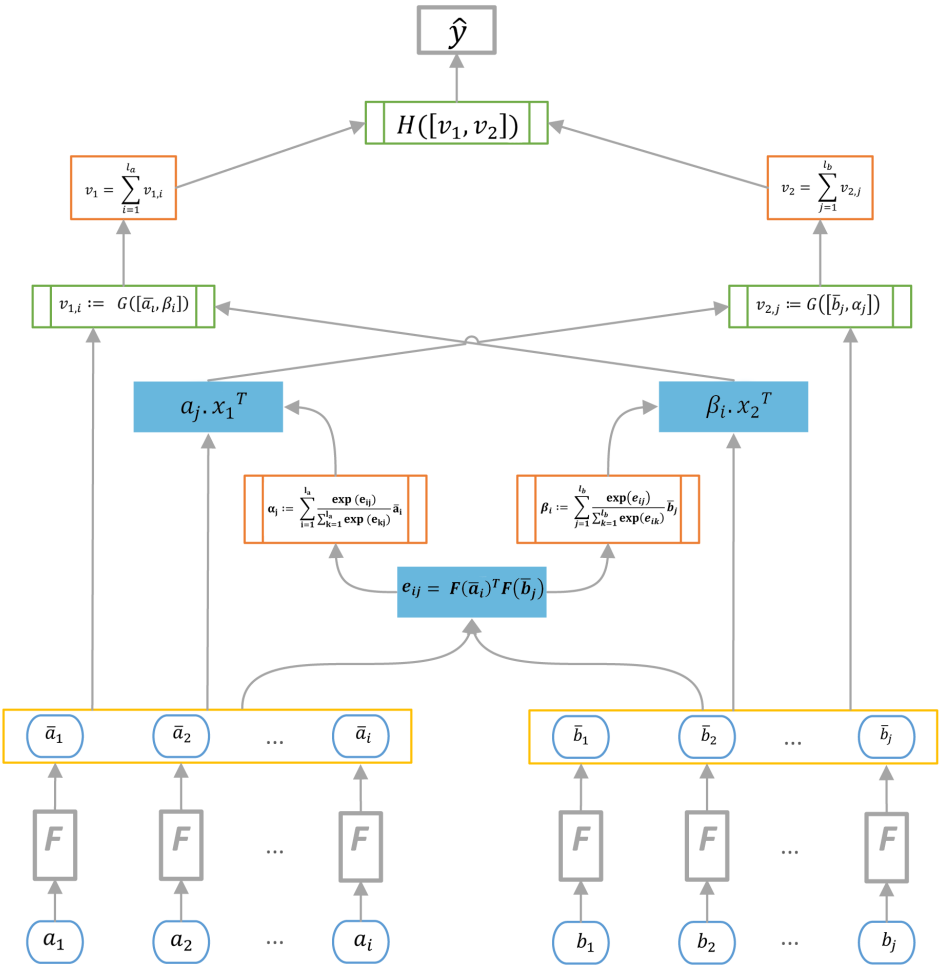

Figure 3.4. Decomposable Attention model architecture graph

3.3.2. Enhanced LSTM for Natural Language Inference

It is discussed that Decomposable attention model provides lightweight solution and achieves the state-of-the-art results. Another model that this work implemented is Enhanced LSTM for Natural Language Inference, ESIM. Unlike Decomposable attention model this approach takes the advantage of Bidirectional Long Short-Term Memory. Bidirectional LSTM method takes an input and process the information through forward and backward in time, this can also be represented as left-to-right and right-to-left. Research states that chain LSTM (building LSTM layers upon each other) would improve the accuracy. ESIM is examined under three title as follows input encoding, local inference, and inference composition.

Sentences are represented as $a = (a_1, a_2, a_3...a_{la})$ and $b = (b_1, b_2, b_3...b_{lb})$. a denotes premise and b denotes hypothesis pair. In this case $a_i \vee b_i \in \mathbb{R}^l$ is the embedding representations of l dimensions. These embeddings can be acquired by using pretrained weight or pre trained language transformer encoders. Aim is to predict correct label y that denotes relation between premise and hypothesis.

Input Encoding: As a building block this work implements Bidirectional LSTM method. LSTM utilizes set of soft gates together using a memory cell to control the message flow. This results with efficient modelling and preservation of long-distance information in a sequence. LSTM architecture can be explained as follows.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{3.12}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{3.13}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3.14}$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{3.15}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{3.16}$$

$$h_t = o_t \cdot \sigma_h(c_t) \tag{3.17}$$

Equation 3.12 to 3.17 represents the inner calculation of a LSTM cell.
In the formulation;

- $x_t \in \mathbb{R}^d$ represents input vectors of LSTM unit.
- $f_t \in \mathbb{R}^h$ represents forget gate's activation vector.

- $i_t \in \mathbb{R}^h$ represents input / update gate's activation vector.
- $o_t \in \mathbb{R}^h$ represents output gate's activation vector.
- $h_t \in \mathbb{R}^h$ represents the hidden state vector. Also, the output vector of LSTM unit.
- $\tilde{c}_t \in \mathbb{R}^h$ represents the cell input activation vector.
- $c_t \in \mathbb{R}^h$ represents the cell state vector.
- σ_g: represents sigmoid function.
- σ_c: represents hyperbolic tangent.
- σ_h: represents hyperbolic tangent.

LSTM units removes the major limitations of RNN architecture. RNN suffers from vanishing or exploding gradients. On the other hand, RNN allows more flexible architecture design but LSTM provides better gradient flow. This research uses BiLSTM that computes the information in both ways.

$$\overline{a}_i = BiLSTM(a, i), \qquad \forall i \in [1, ..., l_a] \tag{3.18}$$

$$\overline{b}_j = BiLSTM(b, j), \qquad \forall j \in [1, ..., l_b] \tag{3.19}$$

Equation 3.18 and 3.19 represents the first BiLSTM operation in ESIM model.

$\overline{a}_i$ and $\overline{b}_j$ are the hidden outputs obtained from BiLSTM unit at the time i for given input a and b.

Local Inference Modelling: Local inference utilizes a form of hard or soft alignment. This method allows to relate important sub-components of premise and hypothesis. This method states that using bidirectional encoding for alignment provides better results. In other word taking the attention of bidirectional output, results in better representations. Soft alignment is calculated with given formula.

$$e_{ij} = \overline{a}_i^T \cdot \overline{b}_j \tag{3.20}$$

Equation 3.20 in the formula $\overline{a}_i$ and $\overline{b}_j$ represents the outputs of BiLSTM layers that are used in input encoding stage.

Unlike Decomposable Attention model, no feed forward layer is used after attention. Instead of feed forward, attention weights are used to calculate inference over

sequences. It is known that local inference is calculated with attention between premise and hypothesis and the result denotes the relevance of the pairs. The relevancy between premise and hypothesis is expressed using e_{ij}. Relevance can be detailed as, where $\tilde{a}_i$ is the weighted sum of $\{\overline{b}_j\}_{j=1}^{l_b}$. This expressed that $\overline{a}_i$ related content in $\{\overline{b}_j\}_{j=1}^{l_b}$ will be selected and symbolize as $\tilde{a}_i$.Following formulation explains the weighted summation.

$$\tilde{a}_i = \sum_{j=1}^{l_b} \frac{exp(e_{ij})}{\sum_{k=1}^{l_b} exp(e_{ik})}, \qquad \forall i \in [1, ..., l_a] \tag{3.21}$$

$$\tilde{b}_j = \sum_{i=1}^{l_a} \frac{exp(e_{ij})}{\sum_{k=1}^{l_a} exp(e_{kj})}, \qquad \forall j \in [1, ..., l_b] \tag{3.22}$$

Equation 3.21 and 3.22 represents the weighted sums operation over attention weights.

ESIM enhances the local inference information using the attention, and weighted sums of given attended variables. Enhancement process takes the element vise product $< \overline{a}, \tilde{a} >$ and $< \overline{b}, \tilde{b} >$. It is hypothesized that this process would concertize inference information between pairs. Later, the information captured from element vise production is concatenated with original vectors.

$$m_a = [\overline{a}; \tilde{a}; \overline{a} - \tilde{a}; \overline{a} \odot \tilde{a}] \tag{3.23}$$

$$m_b = [\overline{b}; \tilde{b}; \overline{b} - \tilde{b}; \overline{b} \odot \tilde{b}] \tag{3.24}$$

Equation 3.23 and 3.24 represents the concatenation of elements

Local inference information is obtained using second BiLSTM layer which is also implies the chain LSTM architecture.

$$v_{a,i} = BiLSTM(m_a, i), \qquad \forall i \in [1, ..., l_a] \tag{3.25}$$

$$v_{b,j} = BiLSTM(m_b, j), \qquad \forall j \in [1, ..., l_b] \tag{3.26}$$

Equation 3.25 and 3.26 represents the second BiLSTM layer after concatenation.

After composition of local inference information, resulting vectors are sent to pooling phase where the model takes the mean and max values of the given vectors. By doing that, fixed size vectors are achieved to send to final classification layer.

Pooling: This phase uses max and mean pooling of vectors obtained from second BiLSTM. These vectors are later concatenated to form fixed size vector. Fixed size vector

is then passed to the last dense layers and softmax layer. Pooling strategy is explained as follows.

$$V_{a,ave} = \sum_{i=1}^{l_a} \frac{v_{a,i}}{l_a}, \qquad V_{a,max} = max_{i=1}^{l_a} v_{a,i} \tag{3.27}$$

$$V_{b,ave} = \sum_{j=1}^{l_b} \frac{v_{b,j}}{l_b}, \qquad V_{b,max} = max_{j=1}^{l_b} v_{b,j} \tag{3.28}$$

$$v = [v_{a,ave}; v_{a,max}; v_{b,ave}; v_{b,max}] \tag{3.29}$$

Equation 3.27 and 3.28 represents the average and max pooling operations. 3.29 represents the last concatenation operation before softmax.

After pooling, concatenated vectors are sent to feed forward network. As a last step outputs of feed forward layer fed in to softmax layer for final prediction. Softmax layer returns prediction scores for each label in the form of $\hat{y}$. Using argmax, model provides the final prediction score $\hat{y} = argmax_i \hat{y}_i$

Our work implements two best algorithms that achieved state of the art results in their time. Decomposable Attention model uses soft alignment and decomposes the problem in to feed forward layers. Their claim is that constructed model works as well as LSTM networks (Parikh et al., 2016). . Second model we implemented is ESIM, Enhanced LSTM for Natural Inference. ESIM (Chen et al., 2017) claims that using Bidirectional approach gives much better results. This is assumed because bidirectional layers consider the information in both ways, left-to-right and right-to-left.

Algorithm 2 ESIM

procedure ESIM(a_1,a_2,...,a_n and b_1,b_2,...,b_n)
    m = embedding array of length 64
    for i =1 to m **do**
        x_1, x_2 = BiLSTM$_1$($[a_i][b_i]$)
        $attention$ = Dot$[x_1][x_2]$
        e_1,e_2 = Softmax$[attention_{axis2}][attention_{axis1}]$
        _$x1$, _$x2$ = Expand_dims(x_2), Expand_dims(x_1)
        _$x1$, _$x2$ = Sum(Multiply([e_1,_$x1$]), Sum(Multiply([e_2,_$x2$])
        m_1 = Concatenate[$x1$, _$x1$, Subtract([$x1$,_$x1$]), Multiply([$x1$,_$x1$])]
        m_2 = Concatenate[$x2$, _$x2$, Subtract([$x2$,_$x2$]), Multiply([$x2$,_$x2$])]
        $y1$, $y2$ = BiLSTM$_2$([m_1], [m_2])
        $av1$, $av2$ = Max(Mean([m_1])), Max(Mean([m_2]))
        y = Softmax(Dense(Dropout(Concatenate([$av1$, $av2$]))))
    end for
    return predicted label
end procedure

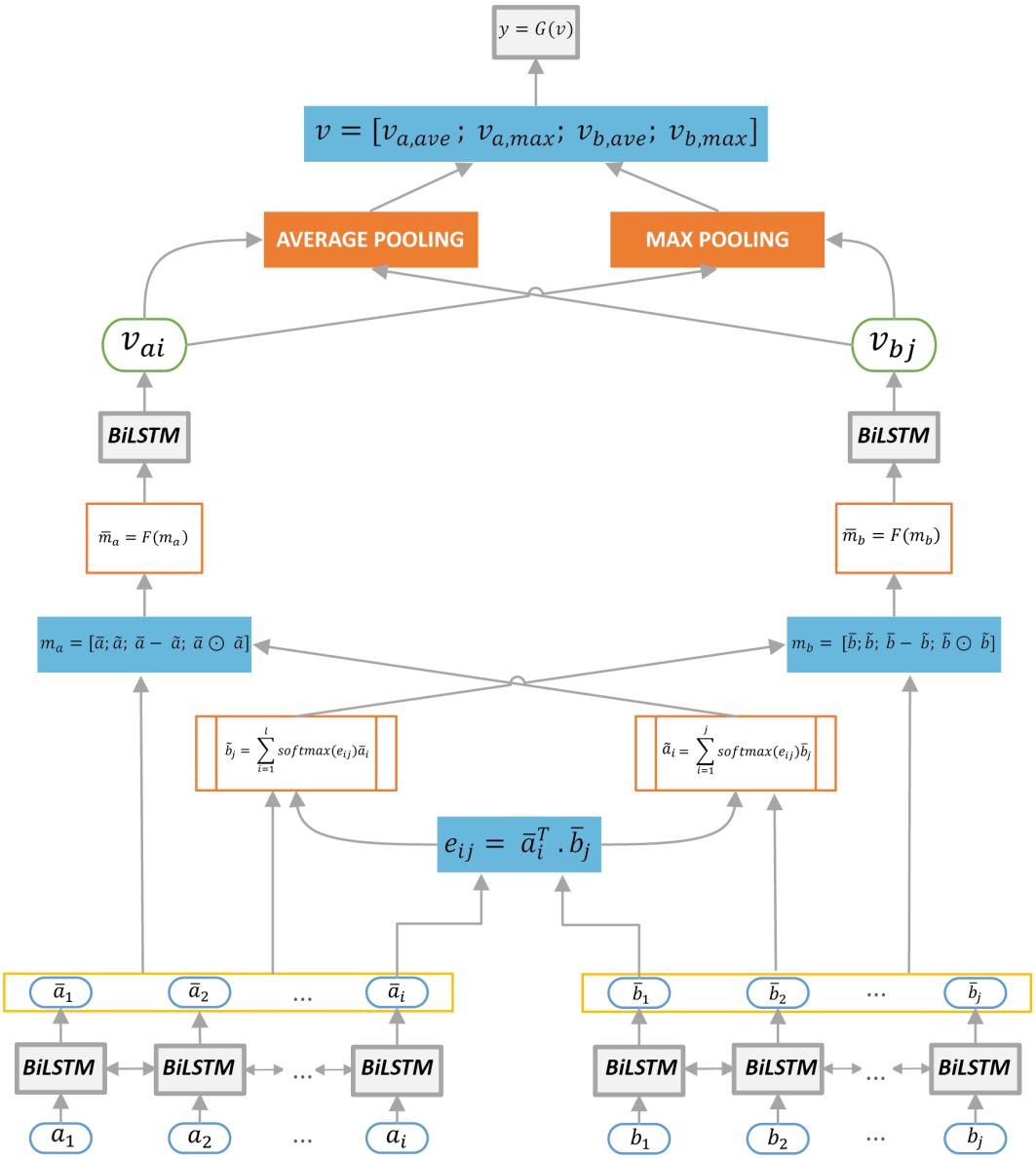

Figure 3.5. ESIM model architecture graph

CHAPTER 4

RESEARCH METHODOLOGY AND PROPOSED SOLUTION

4.1. Problem Definition

It is mentioned that technological advancements created an environment for people to share their opinions and thoughts. These environments can be collected under the roof of social media term. Social media can be defined as computer-generated interactive virtual environment where real individuals can express their interests, thoughts, ideas, and opinions in virtual communities, regardless of their cast and locations in the world (Kietzmann et al., 2011). Feelings and opinions that are expressed by real individuals carry high value information. This high value information is defined as the feelings of individuals and the problems they have faced, thoughts about the new political issues or opinions about something they experienced. High value information created by individuals are fetched and collected by highly sophisticated systems. Every bit of information has been started being collected each day when an individual comes online. Analyzing semantic relation on this high value information uncovers vital information. This information can provide answers such as if individuals that are arguing on a subject agree or disagree, what is the ratio of agreement and disagreement or if there is a relation between given opinions on a subject at all. This type of analysis is named opinion mining. This work downsizes the opinion mining problem to contradiction – entailment problem.

4.1.1. What Is Contradiction

Contradictory opinions arise when two given sentences, premise and hypothesis are not semantically complementary to each other. Using the same logic entailed opinions arise when given premise and hypothesis are semantically complementary. Contradictions

arise from relatively obvious features such as antonymy, negation, or numeric mismatch. They may also arise from complex differences in the structure of assertion and lexical contrast. Antonyms are words that are opposite to each other in their meaning. Negation stands for disproving the corresponding sentence. Numerical contradiction is finding mismatched numeric expression of similar sentences. Factive stands for word knowledge or lexical contrast between given pair. Structural contradiction is finding structural inconsistency between sentences. Lexical, is so like structural (grammar) contradictory happens when an irregular morph is used with a specific lexical item. Date time contradictory happens when two similar sentences use two different time expressions.

Table 4.1. Contradiction types example (Source: de Marneffe et al., 2008)

Premise	**Hypothesis**	**Contradiction Type**
Capital punishment is a catalyst for more crime.	Capital punishment is a deterrent to crime.	Antonym
A closely divided Supreme Court said that juries and not judges must impose a death sentence.	The Supreme Court decided that only judges can impose the death sentence.	Negation
The tragedy of the explosion in Qana that killed more than 50 civilians has presented Israel with a dilemma.	An investigation into the strike in Qana found 28 confirmed dead thus far.	Numerical
The bombers had not managed to enter the embassy.	The bombers entered the embassy.	Factive
Jacques Santer succeeded Jacques Delors as president of the European Commission in 1995.	Delors succeeded Santer in the presidency of the European Commission.	Structural
In the election, Bush called for U.S. troops to be withdrawn from the peacekeeping mission in the Balkans.	He cites such missions as an example of how America must “stay the course.”.	Lexical.

Opinionated text data created by real individuals contain opinions such as positivity – negativity, agreement – disagreement or entailment - contradiction. This work addresses the problem of opinion analysis on the basis of pairwise contradiction analysis. As can be understood from the table, there is a variety of options for contradiction types. Our work proposes a solution that can use all or at least most of these features with a simple pipeline. Currently, thanks to the state-of-the-art language models, contradiction detection can be done end-to-end without compromising the lexical and contradictory

features.

Natural language inference task introduced a set of corpora to serve as benchmarks. Each corpus is created to overcome the limitations caused by its predecessors. BERT emerged as the most powerful contextual representation scheme. In this work, state of the art learning-based algorithms are used to build a classifier that can successfully categorize premise and hypothesis pairs.

4.2. Research Question

High value information inside the collected raw data can be mined using appropriate tools. Mining information from raw data requires some set of standards. These standards can be defined as pre-processing, feature extraction, building a learning-based model, testing the model on human annotated ground truth data and real-life field test. This work addresses the opinion mining task as an opinion classification problem. To be specific, opinions that are expressed around a topic are classified as entailment or contradiction.

In text analysis, hand-crafted domain specific data and annotation by real experts is an important requirement. A second consideration is whether this hand-crafted data can be generalized to other contradiction problems.

Another issue to consider is the possibility to create a robust solution. Some research suffers to create a generalized solution. In NLP area, this usually occurs due to the structural differences of text data. Some other limitations occur due to lack of training data to build a solution. Thus, algorithms must be tested in their capability to generalize a solution using limited or context dependent corpora.

4.3. Proposed Solution

Natural language inference research area has human annotated ground truth corpora that provide text examples as pairs. Text pairs are defined as premise and hypothesis. Each pair annotated with three labels *contradiction, entailment and neutral*. These datasets are SNLI Stanford Natural Language Inference corpus, MNLI Multi-Genre Natural Language Inference corpus and ANLI, adversarial Natural Language Inference corpus.

All these corpora differ from each other based on their purpose.

In order to build a solution, text inputs must be represented as numerical values. This representation process is named as feature extraction or vectorizing. There are variety of feature extraction models such as TF – IDF , word weights, sentence weights and pretrained encoder transformer methods. Dense feature representers uses pretrained word weights and their corresponding vocabularies to represent text inputs. Word embedding based solutions are widely used in natural language area until transformer-based networks introduced. Word embedding solutions do not preserve the word's context when representing the word as numerical value. In a situation where same word is used in different context, word embedding based representers assigns the same value to the word and does not preserve the context. On the contrary transformer-based representations assign different weights for each situation. Thus, enables to preserve context when representing words as features.

It is discussed that language inference problem is proficient on hand crafted human annotated corpus. Also, it is clear to say that there are variety of feature extractors provided by different sources. To provide a language inference solution, a classifier model is required. Classifier learns from the data and builds a neural network-based model. Literature shows that deep learning models outperform the conventional machine learning models. In the light of these knowledge, this work implements two well-known language inference models. This work utilizes Decomposable Attention Model for Natural Language Inference and Enhanced LSTM for Natural Language Inference models to propose a solution to language inference problem. Proposed solution is based on using both pretrained and transformer encoders contextualized word weights with two deep learning models, Decomposable Attention and ESIM. Also, our work is the first research that uses ESIM (chain LSTM), Enhanced LSTM for Natural Language Inference with BERT transformer encoder contextualized word embeddings.

To test the models' performance in relation to training data, various combinations are considered in terms of training and test set partitioning. As a last variant, each set is merged into one set gradually. First, SNLI and MultiNLI sets are merged and the model is trained using contextualized word embeddings. Later, all sets are merged into one corpus and the same training phase is repeated. Each model is tested on each corpus's test set to provide better benchmark results. Initially, this work aimed to solve the inference problem at the sentence level. In the preliminary phase, exploratory data analysis was performed to analyze the relationship between premise and hypothesis. As part of the exploratory

data analysis, sentence level fixed size vectors are obtained using BERT. The sentence embeddings are taken from the last layer of [CLS] token in BERT. Decomposable Attention model is trained with those fixed size sentence embeddings. Obtained results showed that in our case the relationship between premise and hypothesis is better to studied on the word level. Thus, our work focused on the word level alignment for inference analysis.

Once the attention is turned to representations of words, Decomposable Attention and ESIM are trained with both pretrained and contextualized word representations. While implementing this solution, BERT's actual word embeddings are taken from the trained TensorFlow model and each embedding is associated with a vocabulary index. These embeddings are pretrained word embeddings and for each token 1024 dimensional embeddings are generated. The inclusion of pretrained BERT embeddings provided a variation in embedding size. The results state that word embedding dimension size affects the overall accuracy. Embedding size effect on accuracy results is also studied in the original paper (Devlin et al., 2019).

During our work, Google released a TensorFlow-hub implementation of BERT. This implementation comes with an optimized Keras layer. This enabled our work to compute Contextualized BERT embeddings with the mentioned learning-based models to achieve near state-of-the-art results.

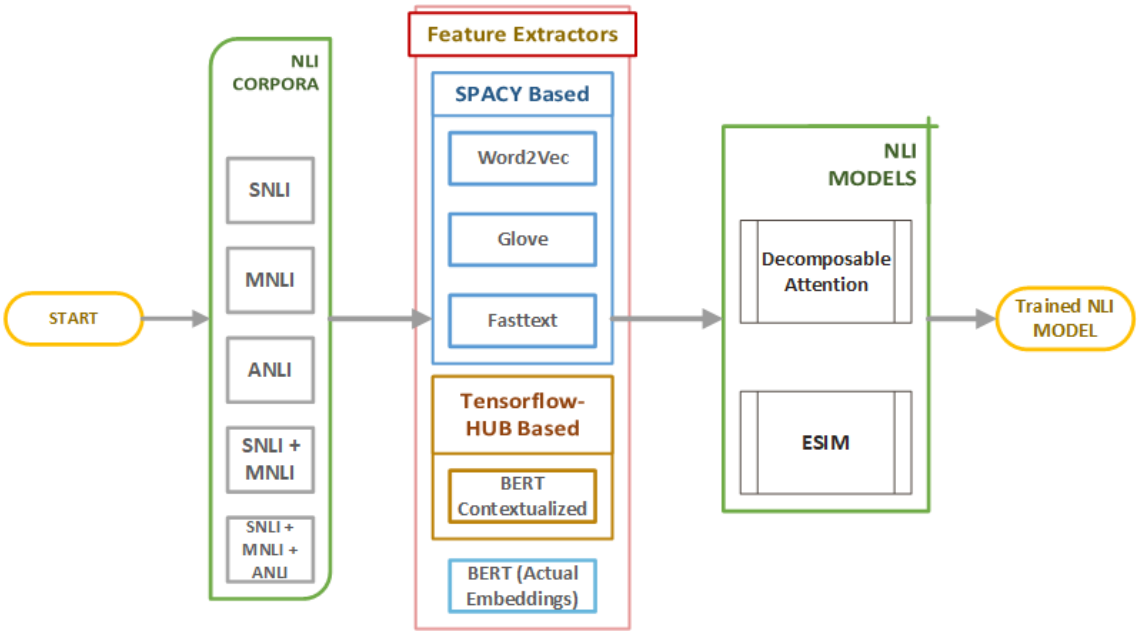

Figure 4.1. Proposed solution detailed flowchart

4.4. Attention Visualization

Inference relation between premise and hypothesis is encoded at the word level. Moreover, learning-based models act like a black box in the training phase. Due to this black box behavior, it is hard to reveal the word level justification for the connection between premise and hypothesis. This relation can be interpreted with the overall score using accuracy results, but still, it does not provide any insights about how the model decides. Apart from the accuracy and several benchmark results, in order to see how learning-based model decide whether given inputs entail or contradict, we implemented attention visualization. Attention is calculated using the weighted sums of dot products between premise and hypothesis word embeddings. To construct attention visualization scores, sums are extracted from the model in prediction time. TensorFlow models can be conditioned to export layer outputs using their layer names. Later, element-wise product operation is applied on the extracted weights and the outputs are normalized. Normalized weights are then reshaped based on the token shapes that are obtained from the premise and hypothesis pairs.

Before the prediction operation, all text inputs must be tokenized. Tokenized words are then fed into the learning-based model both for training and prediction purposes. For our problem, before the prediction time, the model saves the tokens. Then, it combines the reshaped attention weights and tokens to form a graph that represents attention visualization.

4.5. Research Environment

This work uses a variety of feature extractors and natural inference corpora. Thus, a need was emerged for frameworks to keep this research simple and well optimized. This work's research environments are based on two frameworks. Pre-trained word weights-based approaches are used Spacy (Honnibal et al., 2020) natural language processing framework infrastructure. Spacy provides Glove (Pennington et al., 2014) word embeddings out of box. As for Word2Vec (Mikolov et al., 2013) and Fasttext (Mikolov et al., 2018), special tweaking is required. Spacy provides a tool in its structure to manually implement word embeddings except Glove. To reduce model size, spacy hypothesizes that mapping close word representations into one would provide smaller model outputs

without creating any distortion on representations. This hypothesis is justified by creating an environment using both Spacy and Stanford Glove embeddings. Results showed that there is only 0.3% difference between each result. Considering the trade-off between model output and tiny accuracy drop, it is efficient to use Spacy-based Glove embeddings. Considering this result, this work implements both Fasttext and Word2Vec pre-trained word embeddings on Spacy's infrastructure including the tokenizer. This approach does not include BERT's actual word embeddings and its vocabulary. BERT's pretrained word embeddings are outside the scope of Spacy.

This work also used TensorFlow and Keras deep learning frameworks. It is obvious that current Deep Learning solutions require highly optimized and well-tuned frameworks to exploit the benefit of deep learning. Even with highly tuned frameworks, out of memory issues are likely to happen due to the complexity of problems. Our work uses Keras API on top of TensorFlow deep learning framework. Using Keras provides high optimization and less training code complexity. This work experienced the other benefits of TensorFlow such as TensorFlow-hub. Our initial attempt to build a custom Keras layer for BERT-based contextualized word embeddings is suffered from out of memory issues. BERT is a memory intensive transformer network. Therefore, using a custom Keras layer based on BERT is likely to fail. This work utilizes $\text{BERT}_{\text{Large}}$ using TensorFlow-hub. TensorFlow-hub provides an optimized Keras layer for both word piece tokenization and BERT contextualized embedding extraction. With a small tweaking, our work implemented BERT contextualized embeddings to the state-of-the-art natural language inference models. This work presents model performances using each NLI corpus. This comparison provides a benchmark. Apart from using those corpora, this work provides model performance on social media data to determine bounds on real-world data. The real-world data that are used for this purpose are the argument data crafted by Ivan Habernal. Their work aims to find a more convincing evidence between test pairs on different topics. Luckily, their work provides two different viewpoints of the same topic. For example, one corpus's topic is shaped around enabling plastic material usage over the world, the other corpus's topic is in opposite direction and discusses banishing plastic usage. Each corpus comes with two text pairs. With some data pre-processing, our work takes one argument from each corpus. Thus, the resulting dataset contains two arguments that are contradictory. Thus, the results obtained from this test phase are expected to be contradictory (Habernal and Gurevych, 2016).

CHAPTER 5

EXPERIMENTAL RESULTS

5.1. Exploratory Data Analysis

It is mentioned that this research considered the possibility of classifying opinionated sentences without the need of training complex models. To accomplish that, sentence representations are extracted from BERT's CLS token. Fixed size sentence vector representations are then analyzed with quartile analysis to see if there is any relation between given pairs. Results show that the sentence level classification does not help. This also proves that natural inference problems should be worked on token (word) level.

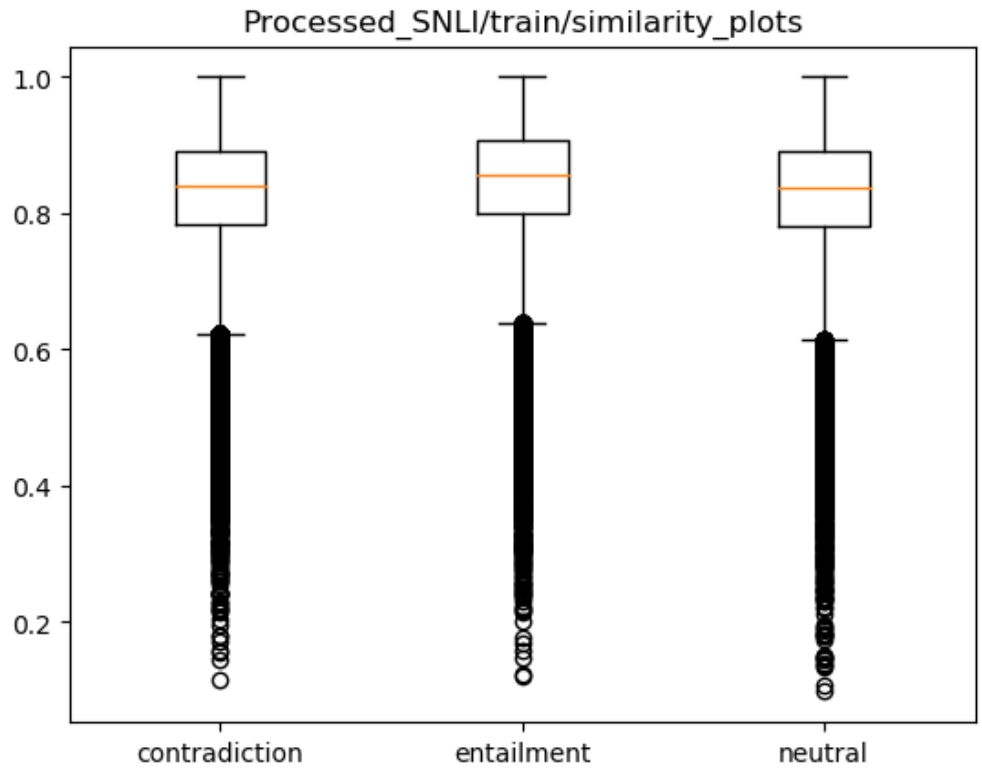

Figure 5.1. Sentence based SNLI quartile similarity analysis result

Figure 5.1 shows that on the sentence level there is no significant difference between classes. Sentence level contradiction analysis may require some additional info such as semantic role labeling or topic information.

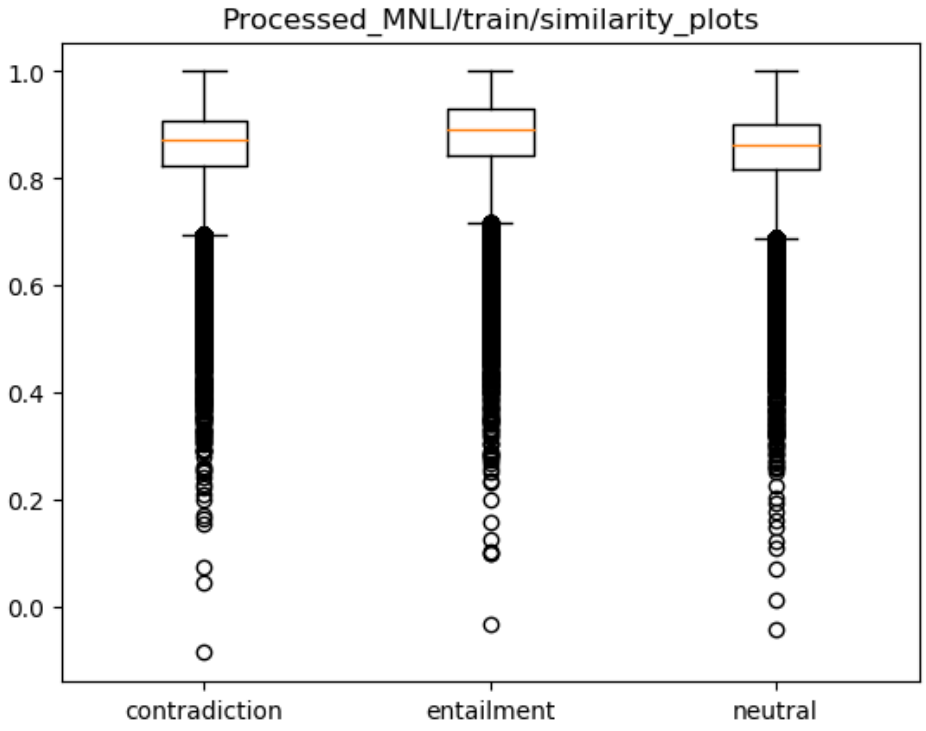

Figure 5.2. Sentence based MultiNLI quartile similarity analysis result

It is known that SNLI corpus is originated around simple and similar genre therefore it would be hard to detect significance between pairs on sentence level. In contrast to SNLI, MultiNLI is built using different genres therefore it is hypothesized that it may be possible to detect significance on MultiNLI pairs on sentence level. Figure 5.2 result shows that even on MultiNLI, pre-trained sentence embeddings are not efficient to detect contradiction on sentence level using similarity approach. In contrast to similarity approaches, sentence level analysis requires additional information and complex sentence originated models to detect contradiction.

Results from figures states that there is no distinguishable difference between sentence encodings of each class. Results provides some insights but to be sure, sentence encodings are trained with Decomposable Attention model and no more than 46% accuracy is achieved. Sentence encodings of each corpus for Decomposable Attention model is achieved using BERT's sentence encoder token [CLS]. When obtaining sentence encod-

ings, fine-tuning is not applied on BERT. In most cases fine-tuning on BERT is suggested but this method requires high computational power due to BERT's complex architecture. To be completely sure, classic taking-mean approach is also applied when testing similarity on contradictory pairs method. When taking mean, each token's contextualized word embedding is obtained. Later, mean of all word embeddings are calculated to obtain fixed size 1024 dimensional sentence embeddings. Results from taking-mean similarity approach are not different than [CLS] based sentence similarity approach. Applying two sentence embeddings strategy ensures that to detect contradiction on sentence level requires fine-tuned BERT or additional information from sentence pairs. Exploratory data analysis results ensures that word-level contradiction analysis is more efficient, thus our work focused on word-level contradiction classification.

This work aims to present a solution to natural language inference problem. Natural language inference problem is another complex task of NLP to analyze opinions created by individuals. This work downsizes the opinion mining to contradiction – entailment classification problem. Inference problem takes two text inputs and gives a prediction score that indicates whether given sentences entail or contradict. This chapter presents the experimental results for the models trained on natural language inference corpora using various types of feature extractors. In the training phase, inference algorithms are trained with similar parameters as follows:

Table 5.1. Hyper-parameters for the learning-based model

	ESIM	**Decomposable Attention**
Batch Size	32	1024
Epoch	20	50
Learning Rate	0.0004	0.001
Maximum Sentence Length	64	64
Hidden Neuron Size	300	300
Early Stopping	5	5
Dropout	0.5	0.2

Results are given separately for each natural language inference corpus type using two state of the art algorithms. To keep this work simple, SNLI corpus is selected to be a standard benchmark. All models are trained on SNLI corpus with all feature extraction methods. This work finds the ESIM algorithm more capable to successfully detect inference. For other corpora, only ESIM algorithm is used to train new models and results are presented in the following sections.

5.2. SNLI Corpus Trained Model Results

Table 5.2. Decomposable Attention model accuracy scores on SNLI

	Train	Dev	SNLI Test	MNLI Matched	MNLI Mismatched	ANLI Test
Word2Vec	**85.79**	85.51	85.16	**59.66**	60.08	29.62
Glove	85.28	85.99	85.50	59.18	59.97	30.28
Fasttext	85.78	**86.14**	**85.51**	59.47	**60.40**	**32.06**
BERT Actual Embeddings	85.06	85.27	85.04	59.37	59.99	30.34
BERT Contextualized Embeddings	—	—	—	—	—	—

Table 5.3. ESIM model accuracy scores on SNLI

	Train	Dev	SNLI Test	MNLI Matched	MNLI Mismatched	ANLI Test
Word2Vec	87.62	85.86	85.16	62.29	60.42	30.56
Glove	89.21	86.44	85.03	59.00	57.93	29.85
Fasttext	87.82	87.30	87.10	61.05	60.12	28.91
BERT Actual Embeddings	**90.45**	87.72	87.03	62.24	60.93	27.56
BERT Contextualized Embeddings	89.45	**88.44**	**88.00**	**66.90**	**67.75**	**31.59**

SNLI is the first large corpus introduced by Stanford Natural Language Processing Group. This corpus is human annotated and contains over 500K examples. With the introduction of SNLI leader board, it became standard to train every new deep learning inference model with SNLI corpus. The following results are given separately based on the algorithm SNLI trained with.

Decomposable Attention model is only trained with pre-trained word embeddings. There are two reasons why Decomposable Attention is not trained with BERT. First, Decomposable Attention model works best with 1024 batch size. Such a big batch size do not work well with BERT when extracting embeddings on the fly approach. The bigger the batch size the more the graphic ram consumption. After few iterations code throws OOM error. Second, when batch size is lowered to 32 same as ESIM. Decomposable Attention model can not achieve more than 40% accuracy. It is assumed that Decomposable

Attention architecture works well with larger batch size but this size of batch size throws OOM error on BERT side. Thus, Decomposable Attention only used with SNLI corpus to provide solid comparison between ESIM and Decomposable Attention architectures. Compared to Decomposable Attention, ESIM can successfully train inference model both using pre-trained and contextualized word embeddings. Individually, Decomposable Attention model achieves the best accuracy using Fasttext. Considering ESIM results, BERT actual word embeddings fail to achieve the best result only with 0.47% difference.

ESIM achieves near state-of-the-art result when trained on SNLI corpus using BERT contextualized word embeddings. ESIM-BERT only fails to achieve the best result on the training set accuracy and this result hypothesizes that BERT contextualized embeddings based solution are more prune to overfitting. Table 5.3 shows that low training set accuracy on ESIM-BERT can be interpreted as contextualized word embeddings provide much better representation information without compromising context structure and robustness to overfitting.

It is stated that this work uses SNLI for global benchmark. Results show that ESIM based solutions are superior to Decomposable Attention based solutions. From now on presented results will only contain ESIM based results.

5.3. MNLI Corpus Trained Model Results

Table 5.4. ESIM model accuracy scores on MNLI

	Train	**Dev**	**SNLI Test**	**MNLI Matched**	**MNLI Mismatched**	**ANLI Test**
Word2Vec	75.77	72.84	61.44	72.13	72.94	28.42
Glove	**80.91**	73.07	62.76	73.42	74.12	29.11
Fasttext	78.79	73.35	62.97	73.94	75.01	27.11
BERT Actual Embeddings	79.40	76.88	**68.22**	77.20	76.82	29.84
BERT Contextualized Embeddings	80.76	**80.43**	66.15	**79.40**	**80.02**	**30.84**

MultiNLI corpus is created to expand the limits of SNLI. SNLI is constructed around specific genre. In contrast to SNLI, MNLI is constructed composing 10 different genres. Results show that for pre-trained word embeddings-based inference mod-

els trained on MultiNLI corpus, BERT actual embeddings proves success over others on SNLI test corpus. When contextualized information is considered, the difference between results becomes different. The only bad result is achieved on evaluating the model on SNLI test set. This can be interpreted as the difference between corpora. As for MNLI corpus based results, ESIM-BERT achieves 79.40% for matched and 80.02% for mismatched accuracy. Last reported results on ESIM architecture with Glove on MNLI is 72.4% and 71.9% for matched and mismatched respectively according to MultiNLI official web site. Our results shows that Contextualized word embeddings performs much better on NLI tasks by achieving nearly 8% accuracy difference compared to pre-trained word embeddings results provided by MNLI web site.

5.4. ANLI Corpus Trained Model Results

Table 5.5. ESIM model accuracy scores on ANLI

	Train	**Dev**	**SNLI Test**	**MNLI Matched**	**MNLI Mismatched**	**ANLI Test**
Word2Vec	62.21	40.38	43.20	39.88	33.15	33.18
Glove	66.51	41.75	42.34	40.02	39.56	32.17
Fasttext	68.06	40.38	44.23	40.38	41.76	31.48
BERT Actual Embeddings	**70.93**	41.18	**47.65**	49.38	41.76	31.48
BERT Contextualized Embeddings	61.80	**40.06**	45.22	**53.46**	**53.75**	**42.34**

Adversarial NLI corpus is created using human – model in the loop approach. Thus, it enables never ending corpus creation and more challenging benchmark results. This approach aimed to create more challenging premise – hypothesis pairs. Table 5.5 shows that models trained on ANLI corpus with various types of embeddings usually perform poorly on test corpora. The poor performance can be attributed to the structure of the ANLI and its creation logic. it is mentioned that ANLI is created by fooling best BERT model to provide much more challenging benchmark. Thus, requires new model architectures and additional information such as SRL (Semantic Role Labeling) , word knowledge or fine-tuning on BERT. Other reason why our models perform poorly on ANLI is that ANLI corpus is composed of long context and short hypothesis as it showed

in table 3.5. Context is composed of three to five sentences which exceeds our 64 sentence length threshold. One can increase the threshold but this method comes with its own problem such as when using BERT with longer sentence length, BERT requires more memory on graphic ram and increases the computational complexity on the model. In the following section our work proposed a work-around for this problem which decomposes the problem in to two sub-problems.

Despite the poor performance of the trained models, BERT based ESIM achieves the best result using both pre-trained and contextualized embeddings. Contextualized BERT embeddings outperforms BERT pre-trained actual embeddings. Results show that preserving the context information with contextualized word embeddings helps models to perform better even on the challenging Adversarial NLI corpus. Nevertheless, even with BERT contextualized word embeddings with chain BiLSTM architecture, our models are not prune to overfitting.

5.5. Downsizing Corpus Using Semantic Relation

In the previous section it is mentioned that our models perform poorly on ANLI corpus. This poor performance reasoned with two explanation. First is ANLI corpus constructed fooling the best BERT model to provide challenging never-ending benchmark. Second is long context sentences. First problem requires more sophisticated solutions such as new model architectures or additional semantic information when training NLI model. Second problem can be solved in two ways. First way is to increase the sentence length threshold up to 256 or 512. This solution comes with its own problem such as requiring more graphical memory or computational power since increasing threshold also increases the embedding size that is fed to model. Second way is to downsize the model using semantic relation.

Our proposed solution downsizes the context - hypothesis pairs to single sentence that are most relevant to each other. Downsized corpus achieved using $\text{BERT}_{\text{Large}}$ with cosine similarity. In our solution each context and hypothesis pairs first converted to contextualized word embeddings based representation. Each sentence representation pair compared with each other using cosine similarity. Sentence pairs and their semantic relation scores are stored. In the last step pairs with highest semantic relation scores are extracted to construct a new downsized corpus. Key point in this experimental test is

that our work used $\text{BERT}_{\text{Large}}$ transformer which preserves the context of the sentence when representing the given text. Thus, reasons the use of cosine similarity method when calculating the semantic relation.

Result of our semantic relation downsized ESIM-BERT ANLI corpus model is provided in the below table 5.6

Table 5.6. Downsized ANLI model results with $\text{BERT}_{\text{Large}}$ and ESIM

	Train	**Dev**	**Test**
ANLI Normal	61.80	40.06	42.34
ANLI Semantic Pre-processed	**70.15**	**41.02**	**43.96**

As it is provided in the table 5.6 our vanilla semantic extraction method improves the accuracy on test data by 2%. Downsizing the corpus method combined of two different steps, first is finding semantic relation between pairs and second is training downsized corpus with NLI networks. This method is developed for two reasons, first is for low computational power working environments. By using semantic relation extraction one can downsize the corpus to single sentence pairs and train a new NLI model without compromising the sentence length. Second is for document level semantic relation without considering computational power. This method assumes that working environment can provide enough computational power regardless of the sentence length. In document level approach one can increase the size of input length to 512 tokens and extract semantic related paragraphs to construct downsized version of documents and train NLI model to find contradictions. In both scenarios one must always use contextualized representers in order to achieve meaningful results. Conventional pre-trained word embeddings do not consider the context of the given text and produces same vector for same token regardless of their meaning in the sentence, this may cause performance decrease when extracting semantic relations.

5.6. Combined Corpora Trained Model Results

It is known / hypothesized that learning-based algorithms perform better with more training data. Relying on this hypothesis, this work trains the ESIM model with BERT embeddings both contextualized and pre-trained (actual embeddings). Previous

results show that BERT based NLI models outperform others. Thus, the effect of corpora combination is tested on the ESIM model with BERT feature representations.

To combine all corpora, each corpus is converted to SNLI format and saved as jsonl file. Later, all SNLI formatted corpus is combined in two different order. First is SNLI – MNLI combination. Second is SNLI – MNLI – ANLI combination. Reason our work used different combination is that ANLI is created in different way and our work wanted to present the effect of ANLI corpus on the behaviour of trained model.

Table 5.7. ESIM model accuracy scores on SNLI – MNLI – ANLI combined corpus with BERT actual embeddings

	Train	**Dev**	**SNLI Test**	**MNLI Matched**	**MNLI Mismatched**	**ANLI Test**
SNLI	90.45	87.72	87.03	62.24	60.93	27.56
MNLI	79.40	76.88	68.22	**77.20**	76.82	29.84
ANLI	70.93	41.18	47.65	49.38	41.76	31.48
SNLI – MNLI	86.10	82.15	**87.15**	76.79	77.40	27.40
SNLI – MNLI – ANLI	85.32	76.84	86.98	77.10	**78.60**	**41.93**

Our first experiment is conducted using BERT actual embeddings with SNLI – MNLI corpus. BERT actual embeddings provides 1024 dimensional vectors with really small dictionary. This compact and high performance solution is provided thanks to BERT's full tokenizer. Table 5.7 shows that SNLI – MNLI – ANLI combined NLI model performs best on SNLI test data. This can be reasoned with the diversity of MNLI has positive effect on single genre SNLI test data. On the other hand NLI model trained with MNLI corpus performs best on MNLI Matched test data. Simple explanation for this result is that Matched test data is constructed from MNLI train set distribution. As for Mismatched MNLI test data, SNLI – MNLI – ANLI combined corpus NLI model performs best with this test corpus. MNLI Mismatched test data examples comes from a distribution that are not seen in training set. ANLI's diverse and challenging train set distribution reinforces the model to perform better on the data that are not seen in training set. NLI model performance on ANLI test data is resulted as expected. ANLI has challenging corpus and only ANLI train data included models can perform better on its test set. But improvement on this set is achieved by increasing the diversity with SNLI and MNLI corpora. Increased diversity with augmented train corpus resulted better accuracy on ANLI test set without needing contextualized word vectors. Regardless of this result the fact remains the same, contextualized word embeddings are far more superior to the

pre-trained word embeddings.

Our next experiment on combined corpus is conducted using $\text{BERT}_{\text{Large}}$ contextualized word embeddings. Considering the previous results it is confident to say that context related embeddings performs better with most of the NLI problems.

Table 5.8. ESIM model accuracy scores on SNLI – MNLI – ANLI combined corpus with BERT contextualized embeddings

	Train	**Dev**	**SNLI Test**	**MNLI Matched**	**MNLI Mismatched**	**ANLI Test**
SNLI	89.45	88.44	**88.00**	66.90	67.75	31.59
MNLI	80.76	80.43	66.15	**79.40**	**80.02**	30.84
ANLI	61.80	40.06	45.22	53.46	53.75	**42.34**
SNLI – MNLI	86.15	83.26	86.81	78.51	79.40	29.18
SNLI – MNLI – ANLI	84.94	77.49	87.22	78.82	79.84	42.20

Table 5.8 shows that contextualized word embeddings based NLI models perform well regardless of the corpora size and diversity. When examining the accuracy scores on test sets, each corpus is performed best on their respected test corpus. But this should not be miss interpreted. Combined corpus contextualized embedding trained NLI models perform as good as individual corpus trained NLI model.

When examining results on combined corpus wise, it can be seen that contextualized word embedding based NLI model performs best among all of them.

5.7. Attention Visualization Results

In previous chapters, it is mentioned that learning-based models act like a black box. This behavior restrains the training time evaluation. Thus, researchers can inspect model behavior on test-time only using test sets. Most learning-based algorithms provide evaluation metrics such as accuracy in training time. Sometimes provided evaluation metrics may not be enough to understand how learning-based algorithms relate the given two text inputs. This work provides attention visualization to show which features are attended to decide the relation between text inputs. To do so, learning-based model is conditioned to provide attention weights on test time. This can easily be achieved just by defining layer names on training time and extract the information on test time using the same layers. The following example is a ground-truth entailment example used in the

Decomposable Attention model paper. This work used these text pairs with trained ESIM inference model.

premise: *In the park Alice plays a flute solo.*

hypothesis: *Someone playing music outside.*

It should be noted that attention visualization method works regardless of the given input size. To have better interpretable results, it is advised to use attention visualization with short sentences for better comparable results. The given example is used with ESIM-based NLI models. To get a better grasp of the differences between word representations, Glove – BERT Actual Embeddings – BERT Contextualized embeddings are used. It is assumed that this embedding selection would provide more insights about the effect of embedding method on the trained NLI model.

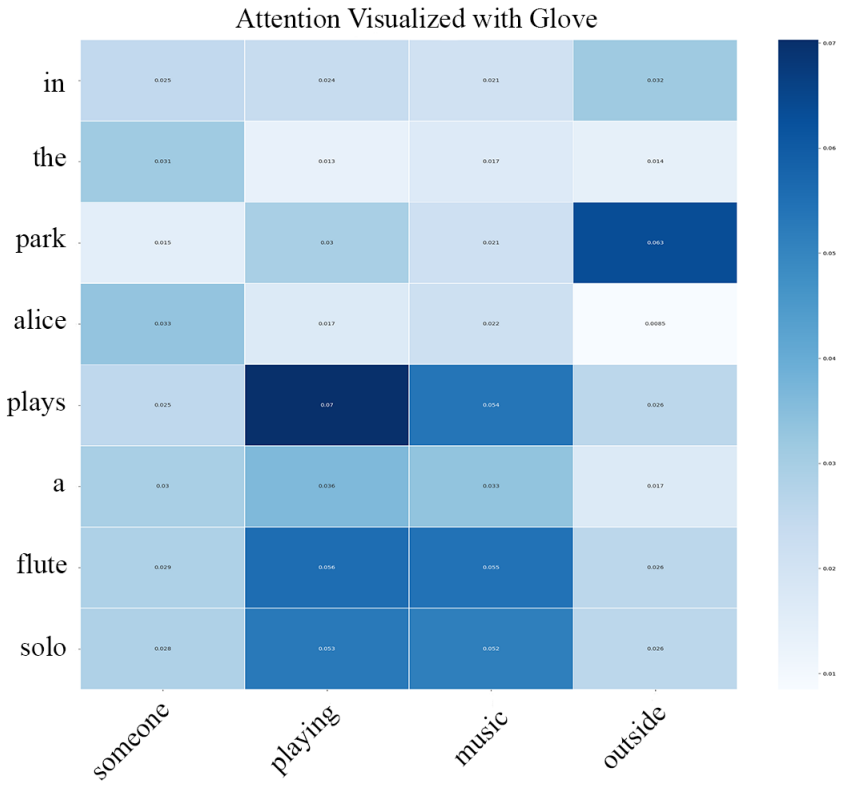

Figure 5.3. Attention heat-map obtained from Glove ESIM Model

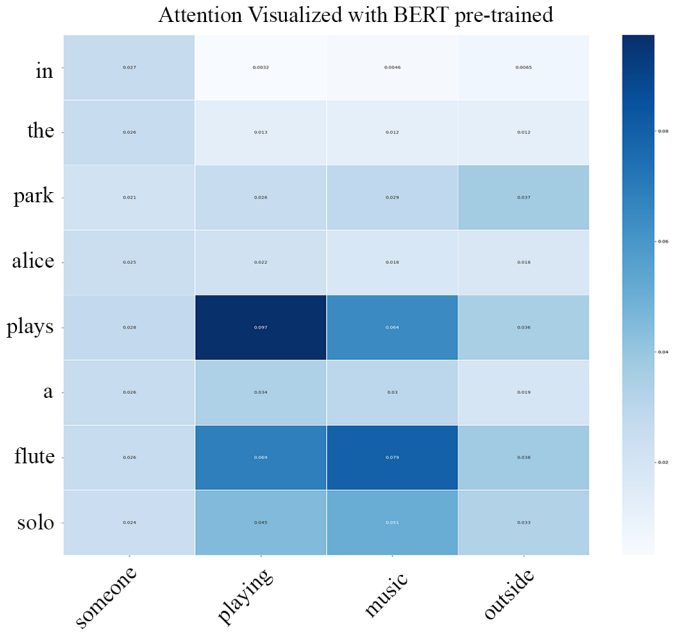

Figure 5.4. Attention heat-map obtained from BERT pre-trained ESIM Model

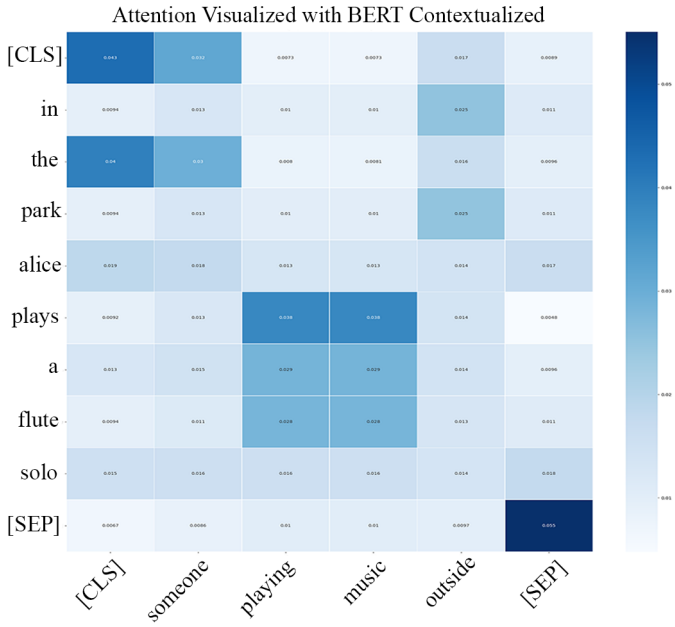

Figure 5.5. Attention heat-map obtained from BERT contextualized ESIM Model

The obtained attention heat-maps show that when using pretrained word embedding-based methods, models tend to attend to the same information but with different scores. Both Glove and BERT actual embeddings models attend to *playing, a, flute, solo* - *playing, music* tokens with different scores. Major difference between Glove and BERT actual embeddings, Glove also attends to *alice* - *someone* tokens and indicates a relation score between person name and person definition. Second difference is Glove attends to *park* - *outside* token and indicates relation between these two tokens. BERT actual embedding method only focuses on *plays, flute, solo* and *playing, music* tokens and determines the inference relation based on these tokens. This can be interpreted as Glove is consist of more than two million unique words thus, gives Glove better vocabulary coverage.

In contrast to other pre-trained models, BERT contextualized model does not attend on *solo* token and does not relate this token with other words. BERT contextualized mostly attends on *play, a, flute* and *playing, music*. Secondly, this model also attends on *in, park* and *outside* with less score and determines the inference relation based on this heat-map. Contextualized word embeddings based models attends directly to the context of the sentence. Considering only context of the sentence gives better judgement to the model when classifying sentence pairs as entailment or contradiction.

5.8. Model Test on Real-Life Sentence Pairs - UKP Corpus

Table 5.9. NLI model real-life example test using UKP corpus

Topic	Contradiction	Entailment	Neutral
Banning Plastic Bottles vs Allowing	63%	0.06%	35%
Christianity vs Atheism	66%	3.8%	29%
Evolution vs Creation	55%	4%	39%
Firefox vs Internet Explorer	39%	5%	54%
Gay Marriage	46%	8%	45%
Having Lousy Father vs Fatherless	7%	16%	75%
Tv vs Books	38%	14%	47%

In earlier chapters, our work mentioned about the UKP Corpus (Habernal and Gurevych, 2016). UKP corpus is created using common crawl technique and labeled with

human annotators. Labeling phase is conducted using Mechanical Turk. UKP corpus is consist of 16 different topics. Each topic has two different sub topics, in other word idea. Their research interest is to find the most convincing argument. It is expected to have two sentences that talks about the same topic which tries to prove its argument is stronger. To give clear structure about data lets pick *Evolution* topic, this topic has two sub topics as *Evolution – Creation*. These two sub topics are saved as individual files. Each sub topics has sentence pairs called *argument1* and *argument2*.

Using this information our work did some data manipulations. We picked *argument1* from *Evolution* and *argument1* from *Creation* and created a contradiction pair using *argument1* sentences from different topics. Process is continued for *argument2* from same sub topics. Thus, our work achieved real-life semi-ground truth test data for contradiction class. This manipulation can also be used to create entailment pairs just by merging topics, it is known that each sub topic's sentence pairs talk about same idea.

Table 5.9 results shows that Plastic Bottle, Religion and Evolution topics meets the expectations and return mostly contradictory results. Firefox, Gay Marriage, and TV topics return below %50 contradiction result. On the other hand Having Lousy Father topic returned as neutral. This table also shows that it is easier to detect contradiction than detecting entailment. In real life examples, models are tend to classify entailed examples as neutral. This miss classification behaviour is open for improvement and can be fixed with additional word knowledge or semantic information.

CHAPTER 6

CONCLUSION

A variety of feature representation methods is introduced along with the natural language inference corpora. Each feature representation addresses drawbacks to its predecessor and provides a new solution. Thus, it results in a better performance. The NLI research area was limited in the training and testing corpus size. In 2015, Stanford Natural Language Group introduced SNLI corpus that removes those limitations. Not long after SNLI corpus, it is stated that SNLI is heavily structured thus it cannot provide a good benchmark. To remove these limitations, MultiNLI corpus is introduced. MultiNLI is composed of both verbal and written English.

All advancements in natural language inference area is re-shaped with the introduction of the bidirectional encoder BERT. BERT is a specialized transformer network that can provide both contextualized embeddings and enables fine-tuning for language processing tasks. BERT set the new state of the art results in eleven language processing tasks including MultiNLI. This improvement also made current benchmark methods obsolete. This problem is addressed by Facebook AI team and the team introduced a new Corpus. New corpus is formed using human-model in the loop approach. This approach is specialized to be never ending corpora creation and provides much more challenging examples.

This work followed all these given advancements in natural language inference tasks. The proposed solution used all text representation methods with all corpora to provide comparable results. Results showed that contextualized word embeddings are superior to other text representation schemes. This is achieved by preserving the context information when representing the features. Also, our work shows that inference problems are better solved with bidirectional learning-based networks. Bidirectional networks learn from the data in both left-to-right and right-to-left. Our work contributes to NLI research area in two ways. This work compares all the text features extractors to provide variety of results. Comparison can be understood best with SNLI corpus. SNLI corpus is trained with both Decomposable Attention and ESIM. Secondly, to our knowledge this is the first work that uses chain LSTM structure (ESIM) with contextualized word embed-

dings (BERT). Results show that using BERT with ESIM provides near state-of-the-art results for natural language inference problem.

This research is built on top of SNLI inference problem yet performance results from other corpora are presented along with SNLI. It is assumed that this heavy work would provide more information and differences between NLI corpora can be observed. Our work obtained the best accuracy results with $\text{BERT}_{\text{Large}}$ ESIM inference learning model. To our knowledge, this is the first work that uses $\text{BERT}_{\text{Large}}$ contextualized embeddings with ESIM inference algorithm. After long hyper-parameters tuning, our ESIM-BERT achieved 88.44% Dev accuracy and 87.99% 88.00% test accuracy. Previous works conducted ESIM - Glove and achieved 88% and ESIM – ELMO achieved 88.7% test accuracy. It is also reported by some individual researchers, re-constructed ESIM – Glove can only achieve 86.68%. Considering this information apart from the accuracy posted on SNLI leader-board, our solution holds the best result among ESIM models. It is reported that the current state of the art is 92.1% according to SNLI leader-board. Our results fall behind the current state of the art result with 4.1%. Since release of BERT, newest researches are focused on additional information when fine-tuning the BERT such as semantic role labeling and etc. Fine-tuning with additional information approaches explains the accuracy difference between our result and state of the art result. Our work provides end-to-end solution with high accuracy without needing additional information. It is safe to say that our ESIM-BERT model trained with SNLI corpus achieves near state-of-the-art result. This work also implemented the MultiNLI. It is reported that ESIM – Glove can only achieve 72.4% accuracy for matched test set and 71.9% accuracy for mismatched set. Our ESIM-BERT solution achieves 79.40% matches accuracy and 80.02% mismatched accuracy. Considering the official results from MultiNLI web page, our solution holds the best ESIM-based result. It is also reported on Kaggle page, MNLI based solutions can achieve up to 90% accuracy but there is no access to code or trained model. To our knowledge, the best official accuracy result comes from $\text{BERT}_{\text{Large}}$ with 86% accuracy. Comparing results, our work achieves close enough result for MNLI but there is room for improvement.

Results show that our work can achieve up to 88% score on SNLI set which is the main domain of this research. Current state of the art results achieved by considering the background information and word knowledge such as Semantic Role Labeling. To improve our results, this work will consider the SRL and word knowledge information with Chain LSTM (ESIM) using contextualized word embeddings (BERT).

REFERENCES

Azman, S. N., I. Ishak, N. M. Sharef, and F. Sidi (2017). Towards an enhanced aspect-based contradiction detection approach for online review content.

Baccianella, S., A. Esuli, and F. Sebastiani (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*.

Badache, I., S. Fournier, and A.-G. Chifu (2018). Contradiction in reviews: Is it strong or low? In *BroDyn@ECIR*.

Banko, M., M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni (2008). Open information extraction from the web. *Commun. ACM 51*, 68–74.

Bowman, S. R., G. Angeli, C. Potts, and C. D. Manning (2015). A large annotated corpus for learning natural language inference. In *EMNLP*.

Chen, Q., X.-D. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen (2017). Enhanced lstm for natural language inference. In *ACL*.

de Marneffe, M.-C., A. N. Rafferty, and C. D. Manning (2008). Finding contradictions in text. In *ACL*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Downey, D., O. Etzioni, and S. Soderland (2005). A probabilistic model of redundancy in information extraction. In *IJCAI*.

Esuli, A. and F. Sebastiani (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*.

Griffiths, T. R. and M. Steyvers (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America 101*, 5228 – 5235.

Habernal, I. and I. Gurevych (2016). What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 1214–1223. Association for Computational Linguistics.

Haddadan, S., E. Cabrio, and S. Villata (2019). Yes, we can! mining arguments in 50 years of us presidential campaign debates. In *ACL*.

Harabagiu, S. M., A. Hickl, and V. Lacatusu (2006). Negation, contrast and contradiction in text processing. In *AAAI*.

Honnibal, M., I. Montani, S. Van Landeghem, and A. Boyd (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Karahoca, A., D. Karahoca, and E. Buyuk (2017). Conflict analysis for turkish debates using text mining and text segmentation techniques.

Kietzmann, J. H., K. Hermkens, I. P. McCarthy, and B. S. Silvestre (2011). Social media? get serious! understanding the functional building blocks of social media. *Business Horizons 54*, 241–251.

Li, L., B. Qin, and T. Liu (2017). Contradiction detection with contradiction-specific word embedding. *Algorithms 10*, 59.

Lingam, V., S. Bhuria, M. Nair, D. Gurpreetsingh, A. Goyal, and A. Sureka (2018). Deep learning for conflicting statements detection in text. *PeerJ Prepr. 6*, e26589.

Mikolov, T., E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin (2018). Advances in pre-training distributed word representations. *ArXiv abs/1712.09405*.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. *ArXiv abs/1310.4546*.

Miller, G. (1995). Wordnet: a lexical database for english. *Commun. ACM 38*, 39–41.

Mukherjee, A. and B. Liu (2012). Mining contentions from discussions and debates. In *KDD*.

Nie, Y., A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela (2020). Adversarial nli: A new benchmark for natural language understanding. *ArXiv abs/1910.14599*.

Parikh, A. P., O. Täckström, D. Das, and J. Uszkoreit (2016). A decomposable attention model for natural language inference. In *EMNLP*.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation. In *EMNLP*.

Ritter, A., S. Soderland, D. Downey, and O. Etzioni (2008). It's a contradiction - no, it's not: A case study using functional relations. In *EMNLP*.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *NIPS*.

Veen, K. V. (2016). Contradiction detection between news articles. Technical report, Univserity of Amsterdam.

Voorhees, E. (2008). Contradictions and justifications: Extensions to the textual entailment task. In *ACL*.

Williams, A., N. Nangia, and S. R. Bowman (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.