Generating Word Embeddings for the Azerbaijani Language and Their Empirical Evaluation on Intrinsic and Extrinsic Evaluation Tasks

By

Umid Suleymanov

Master Thesis

Submitted in Fulfillment of the Requirement
For the Degree of Master of Science

Supervisor:

PhD. Amir Masoud Rahmani

Advisor:

PhD. Behnam Kiani Kalejahi

Khazar University

School of Science and Engineering

2021

Baku, Azerbaijan

Acknowledgements

This document is written in the spring 2021 as a master thesis to fulfill the requirement for degree of master of science. As I have taken my specialization in Computer Science at Khazar University supported by different courses in the Artificial Intelligence field and also considering my previous research background on natural language processing, machine learning fields, it seemed appropriate for me to choose one of the state of art approaches namely word embeddings as a topic for my final thesis. As of writing this thesis, I am Leading Artificial Intelligence Specialist which makes a thesis on AI field very relevant with my practical experience in AI sector.

In the scope of this thesis, I will generate word embeddings for Azerbaijani language and conduct experiments to evaluate the performance of these embeddings both intrinsically and extrinsically. In this document I tried to cover all technical aspects of word embeddings by reviewing the current literatures and trends on this topic This is both a qualitative and an empirical research on word embeddings in which I will present and analyze the empirical results of word embeddings on various NLP tasks such as word analogy, sentiment analysis, text classification and etc. The readers of this document are presumed to have a basic knowledge on the machine learning technologies at a university level or to have some general knowledge or experience on these fields.

I would like to take this opportunity to thank all professors and instructors who taught us during our master program, the management of Khazar University and the Dean Office. Special thanks go to my thesis supervisors PhD. Behnam Kiani Kalejahi and PhD. Amir Masoud Rahmani for their extensive supports and valuable guidance in all aspects of this document. I would also like to thank my family for their supports during this time.

Umid Suleymanov

Baku, Azerbaijan

June, 2021

Abstract

Recently, the success gained by word embeddings and pre-trained language representation architectures has increased the interest to natural language processing significantly. They are at the core of many breakthroughs in Natural Language Processing tasks such as question answering, language translation, sentiment analysis and etc. At the same time, the rapid growth in the number of techniques and architectures makes the thesis on these topics very relevant for Azerbaijani as it is an agglutinative and low resource language. In this thesis, word embeddings will be generated using various architectures and the effectiveness of word embeddings will be analyzed through empirical research in different datasets containing various natural language processing tasks such as sentiment analysis, text classification, semantic analogy, syntactic analogy and etc. This thesis will also research natural language representation approaches from traditional vector space models to very recent word embedding and pre-training of deep bidirectional transformer architectures. Novelty introduced by each of the new technique, its main advantages and the challenges addressed by those have been addressed in this paper. I will also aim at explaining mathematical background where necessary in order to make distinction between architectures more clear.

Keywords

Word Embeddings, Machine Learning, word2vec, GLoVe, NLP, Sentiment Analysis, LSTM, Deep Learning, Semantic Analogy, fasttext.

Table of Contents

Chapter 1

1. Introduction

Natural Language Processing has always been one of the fascinating research topics as it is always believed that there is a connection between human intelligence and human natural language. Considering the fact that an intelligent being can best express its intelligence by communicating intelligently, it is utterly important to tackle the problems of natural language processing before reaching true general artificial intelligence. Even, the outstanding scientist and one of the creators of computer science, Alan Turing, suggested in his famous Turing test that a machine has intelligence if an interrogator cannot differentiate if he is talking to a machine or to a human behind a wall. This once more proves the importance of handling and generating natural language in artificial intelligence field. However, there has always been a challenging question to answer, namely, how to represent human language; the words, sentences, paragraphs so that it is possible for machines to process these and produce meaningful outputs. Considering the fact that, today we have translation, question answering systems, web engines which operates solely on human natural language and produce desired results, we can claim that we have progressed very far in representing human natural language.

The main concentration of this thesis will be on the generation and application of word embeddings which revolutionized the neural representation of natural language and led to many breakthroughs in different NLP fields. In the scope of this thesis, various state-of-art approaches and machine learning techniques will be devised for generating word embeddings. For being able to generate word embedding vectors which truly captures the semantic as well as syntactic meaning and relationship of words, data corpora with hundreds of millions of tokens have been collected from various sources such as news articles, books, scientific articles, poems, novellas written in journalistic, scientific and literary style. Word embeddings allow feature extraction from unstructured text data and has applications in almost all NLP tasks

such as question answering, text classification, sentiment analysis, translation systems and etc.

1.2. Background

Language representation models are intermediary layers encoding information in human natural language for processing by LSTMs, deep learning and other machine learning models. Language representation models can be broadly categorized under three main categories. Figure 1 describes these categories and names of a few well known techniques belonging to those categories. The first child node, Vector Space Models describes traditional frequency based language representation models which was mainstream until the introduction of neural network based language representation models. Beginning from Word Embeddings, the usage of neural networks for learning the word representations become popular and with the usage of word embeddings various architectures emerged. The last node, Contextualized Word Embeddings is deep learning based natural language representation generation techniques, which is now considered state-of-art in many natural language processing tasks.

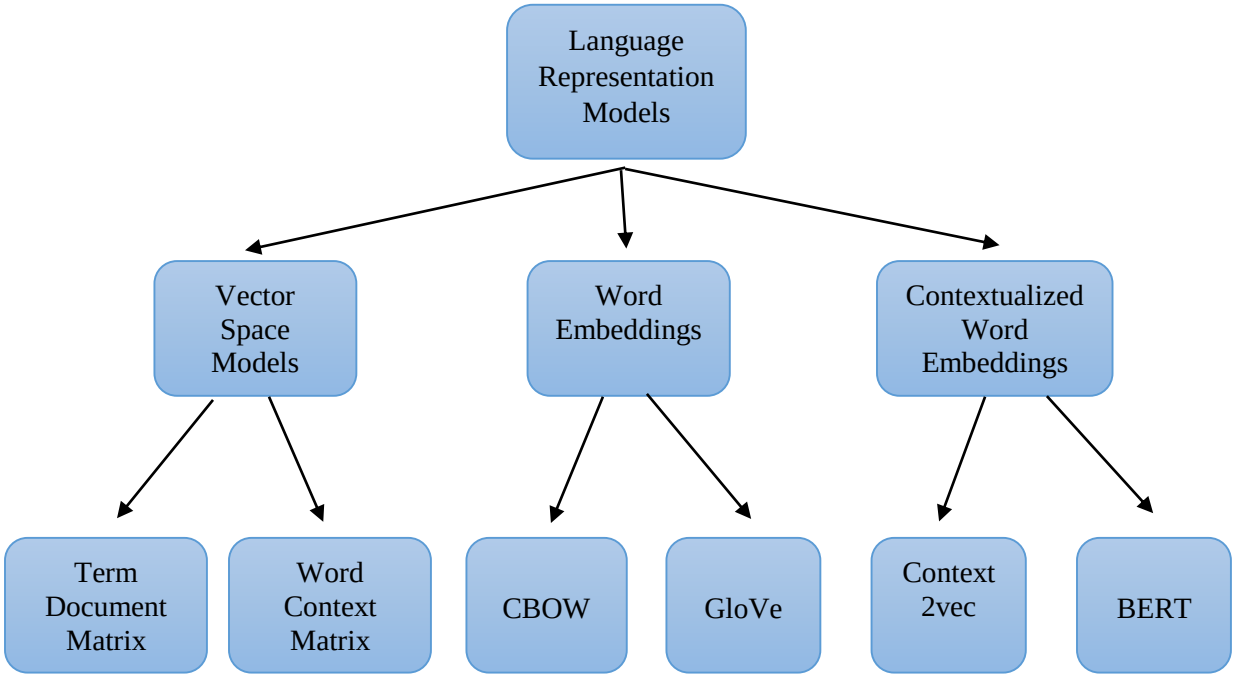

Figure 1: *Language Representation Models*

Recently, the success gained by word embeddings and pre-trained language representation architectures has increased the interest to natural language processing significantly. They are at the core of many breakthroughs in Natural Language Processing tasks such as question answering, language translation, sentiment analysis and etc. At the same time, the rapid growth in the number of techniques and architectures makes the thesis on these topics very relevant for Azerbaijani as it is an agglutinative and low resource language. In this thesis the effectiveness of word embeddings will be analyzed through empirical research in different datasets containing various natural language processing tasks such as sentiment analysis, text classification, semantic analogy, syntactic analogy and etc. This thesis will also research natural language representation approaches from traditional vector space models to very recent word embedding and pre-training of deep bidirectional transformer architectures. Novelty introduced by each of the new technique, its main advantages and the challenges addressed by those have been addressed in this paper. I will also aim at explaining mathematical background where necessary in order to make distinction between architectures more clear.

1.3 Related Work

There are numerous literatures discussing various parts of natural language representation techniques. Collados & Pilehvar (2018) present a survey on the recent approaches and challenges that language modeling architectures face when trying to encode the meaning of the words. They present the approaches under two main groupings: unsupervised sense representations and knowledge based representations. The evaluation techniques which exists for measuring the quality of representations are also covered in the paper by Collados & Pilehvar (2018). At the end, they conclude the paper by giving the applications of word representations and analyzing them according to different criteria such as sense granularity, interpretability, their relevance in various domains. This paper gives good overview of existing approaches on specifically for sense representations, however on the analysis part, more technical details could be provided.

The survey by Otter et al (2020) gives an overview of existing deep learning techniques and their implications on Natural Language Processing. The first part of the survey is dedicated to the general overview NLP and the discussion of recent

deep learning architectures. The discussion of deep learning architecture in the first part could be done shortly, as already NLP and deep learning are two huge fields, it would be better to discuss their intersection on more detail rather than separately different architectures. They have dedicated the next section to the application of deep learning in natural language core fields which cover language modeling, morphology, parsing and semantics (Otter et al., 2020). The survey paper by Otter et al (2020) is also concluded by giving the applications of deep learning in natural language processing.

Another survey focusing on representation of semantic meaning of words is done by Peter & Pantel (2010). The unique feature of this survey is that it tries to connect existing architectures and techniques with general natural language processing hypothesizes. The hypothesizes include statistical semantic hypothesis, bag of words hypothesis distributional hypothesis and etc. This makes this survey and the architectures it discusses very intuitive to understand for the researches who a new to the field. The survey also includes linguistic and mathematical processing of word vectors as separate chapters.

1.4 Research Problem

During the last decade word embedding approaches revolutionized the world of natural language processing completely. Word embeddings provide very accurate representation for human natural language learnt by various machine learning models. Azerbaijani is considered as a low resource language because of the scarcity of well-formed corpora available to the researchers. Moreover, the number of research targeting this both syntactically and semantically rich language is also less. It is also worth noting that Azerbaijani is an agglutinative language meaning words and grammatical clauses are mostly formed by the usage of prefixes and postfixes, making the research of the effectiveness of word embeddings concept for this language even more relevant and well suited for the natural language processing researchers and experts in the whole world. It is already an undeniable fact that word embeddings are very successful representation of human natural language, generating dense vector representation of human natural language which have been applied to many natural language processing tasks such as question answering, sentiment analysis, text classification and etc. very successfully often even achieving new state-of-art performance on these tasks. However, most of the research have

focused on languages like English, French, German and etc, and proved the effectiveness of word embeddings for these language families. This makes the following research question very valid: whether or not word embeddings approaches are a successful representation for an agglutinative language namely Azerbaijani? In order to answer this research question, I will conduct several experiments using word embeddings and try to quantitatively measure and analyze the results of word embeddings in both intrinsic and extrinsic evaluation task settings. These tasks are famously suggested by the creators of word2vec algorithm Mikolov and et al. and is the mainstream way to measure the effectiveness of word embeddings quantitatively. Note that special version of these tasks are specifically prepared for Azerbaijani language. I will apply semantic and syntactic analogy tasks and report the accuracy scores of word embeddings in these intrinsic evaluation tasks. For the extrinsic evaluation tasks, the effectiveness of word embeddings will be demonstrated in the sentiment analysis and text classification tasks and the accuracy scores will be presented and analyzed. After the experiments, I will discuss the effectiveness of word embeddings in this agglutinative language and provide the empirical results to the research question posed.

1.5 Contribution

This paper will encompass natural language representation approaches from traditional vector space models to very recent word embedding and pre-training of deep bidirectional transformer architectures. Novelty introduced by each of the new technique, its main advantages and the challenges addressed by those have been addressed in this paper.

The main contribution of this thesis will be the generation and the comparative analysis of word embeddings for an agglutinative language namely, Azerbaijani which is a low resource language. Agglutinative languages have very different kind of morphological structure. Therefore, how well machine learning based word embeddings will encode semantic and syntactic relations is still a research question that will be answered in this thesis. In the scope of this thesis, I will also apply the generated word embeddings to various NLP tasks such as word analogy, semantic similarity, sentiment analysis and etc. In the scope of this thesis, extrinsic evaluation tasks will be formed for word embeddings and empirical results will be analyzed. These tasks are called extrinsic as in these tasks embeddings are used outside of the

domain they are trained. Moreover, the intrinsic evaluation tasks will also be performed in this thesis in order to fully assess the effectiveness of word embeddings for Azerbaijani. As novelty, the effectiveness of generated word embeddings will be shown in agglutinative syntactic analogy tasks. In this thesis, all the works which brought novelty and significant changes to the already existing approaches on natural language representation techniques and provide the progression of architectures will also be collected and compared that of with Azerbaijani language. The approaches are compared according to various criteria such as accuracy, training speed, model complexity, the quality of generated word vectors and etc.

Chapter 2

Word Embeddings

2.1 Vector Space Models

Vectors are very well-known notation even before the advent of machine learning. The first usage of vectors for natural language representation can be attributed to Salton et al. (1975). The first usage of vectors was for representing documents as a set of vectors based on word usage frequencies. The underlying principle of this model so called statistical semantics hypothesis can be expressed as: people's intention when they use natural language can be deduced by statistical distribution of their word usage (Peter & Pantel, 2010).

In 1950, renowned mathematician and computer scientist Alan Turing's research focused on machine intelligence. Believing the question of "can machines think" to be so general as to border on the pointless, Turing held that if a computer, after proper programming, could perform well on a more specific measure of intelligence, machine learning might warrant further discussion. His idea combined concepts of machine intelligence and language as a measure of that intelligence in what he labeled the "Imitation Game". The game, more akin to a test, required a computer to use written language to demonstrate enough intelligence, expressed through language, adequately enough to fool an average human subject. To measure this intelligence, Turing suggested concealing both a human and a machine before sending them written messages. The machine and the human could then respond to those messages however they would like in a communicative process. The machine was determined to be successful if it could adequately fool the human subject with statistically significant results. Though this was not possible during his lifetime, Turing did predict that by sometime around the year 2000, scientists could program a computer well-enough that after a five-minute conversation, an average human would determine which interlocutor was a machine correctly less than 30 percent of the time. The idea that machines could develop the ability to "think independently" and pass Turing's imitation game came to be known as strong AI. The opposite of

strong AI, known as weak AI, entails computers only simulating human thought, though this does not entail real understanding. Following in Turing's footsteps, other scientists and philosophers developed their own ideas about artificial intelligence and the development of machine learning.

2.2 Term-Document Matrix

The term-document matrix is matrix where rows represent documents and columns represent statistical frequencies of vocabulary words. Because, this model does not take word order into consideration, this is also often called bag of words model (Peter & Pantel, 2010). Here bag correspond to mathematical set where uniqueness of elements is not mandatory, so that two documents one with word "car" 20 times is different from a document with word "car" only once.

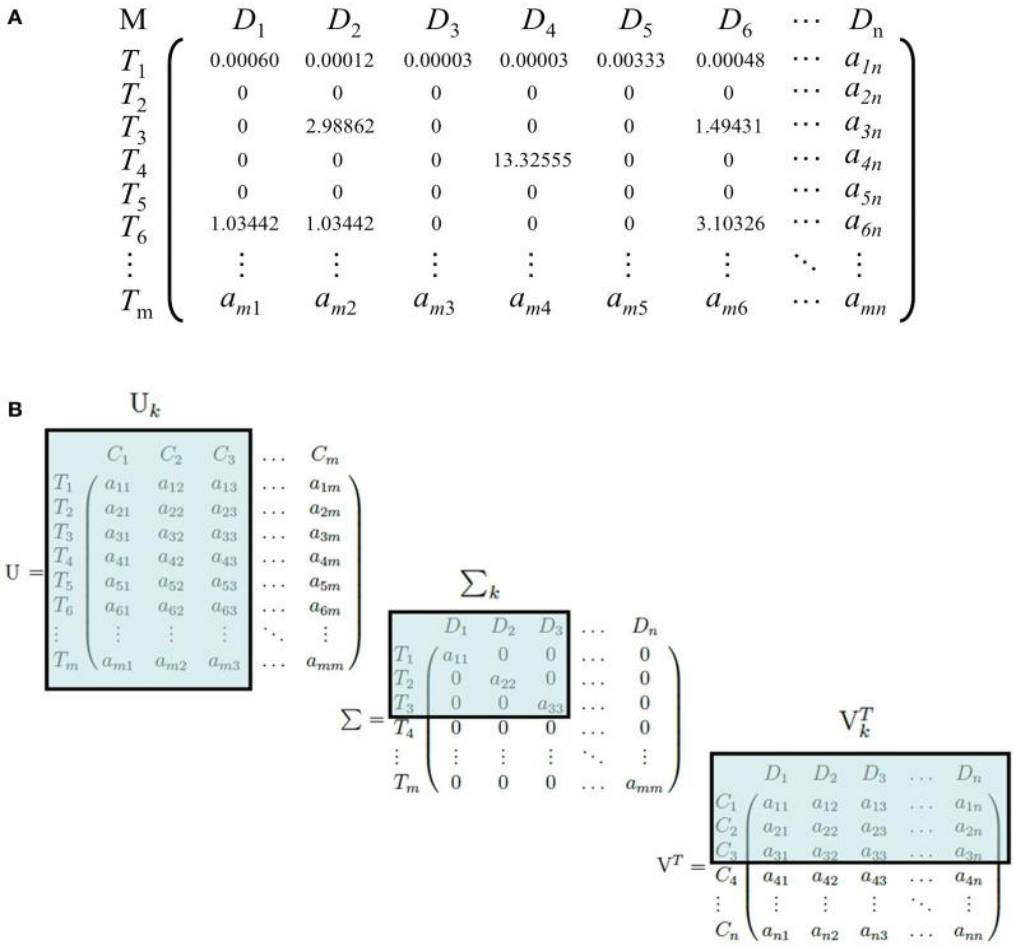

Figure 2: *Term Document Matrix Calculation*

An example of a term-document matrix calculation is shown in Figure 2 where M represents term-document matrix, D represents the set of documents in the dataset, and T represents all the unique terms contained in the whole corpora. As shown in the example, T_1 is a term which occurs frequently in the corpus, therefore having lower weight compared to other terms such as T_3, T_4, and T_6 which are comparatively infrequent terms and therefore have inflated weights. Note here that, each vocabulary word regardless of whether it is used in document or not, should be present in documents column vector. If document does not contain this word, then this words frequency for this document will be zero. By this rule, term-document matrix contains all the words in dictionary and all documents, where i, j-th element is the frequency of j-th vocabulary word in i-th document (Peter & Pantel, 2010). In most cases, as document do not contain all vocabulary words, the number of zeros, hence sparsity will be high. However, documents talking about the same topic, will most probably use same vocabulary words in roughly same frequency. Hence, these two document's vectors in our term-document matrix will have very close numbers for the frequency of these vocabulary words. In other words, these two vectors will be close to each other in document vector space (Salton et al., 1975). This representation although very intuitive, proved itself very useful in many natural language processing tasks, such as document clustering, information extraction, document classification, sentiment detection and etc (Collados & Pilehvar, 2018).

2.3 Word-Context Matrix

The document-based vector space models played a significant role in later development of the word vector representation models (Collados & Pilehvar, 2018). Lund & Burgess were first to explore the possibility of using word's context words' frequencies to construct a word-context matrix (1996). The intuition was very similar to that of statistical semantics hypothesis, namely, this time, instead of documents having similar distribution of vocabulary usage, we had words having similar distribution of vocabulary usage in their proximity. For example, if the words "spoon" and "fork" in very large text corpus, have been used with similar context words, such as "plate", "knife", "eat" and etc. with similar statistical distribution, this intuitively means "spoon" and "fork" are similar objects and should be near to each other in word vector space this time.

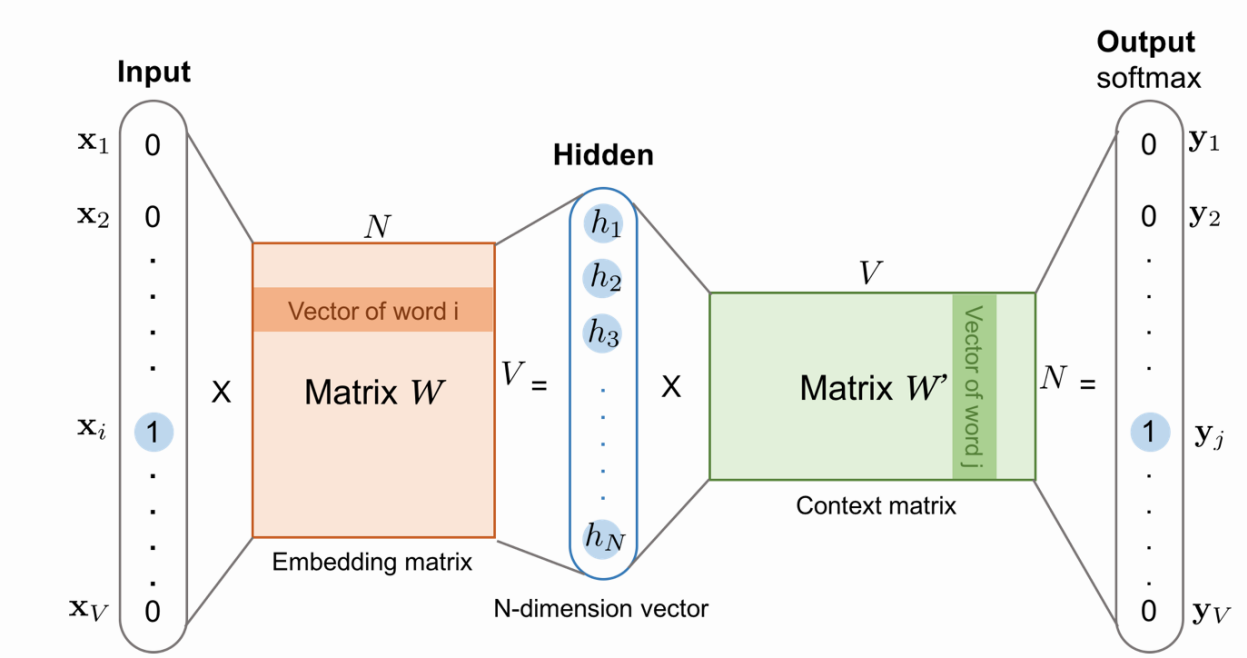

Figure 3: *Word-Context Matrix*

However, these vector space models have their weakness as well, namely, they were producing vectors which have very large number of dimensions, as they were required to be the same size as the size of vocabulary in order to construct the matrix (Collados & Pilehvar, 2018). Researchers Landauer & Dooley applied an approach called Latent Semantic Analysis, which was essentially dimensionality reduction technique for very high dimensional vectors in order to deal with these curse of dimensionality (2002).

2.4 Word2Vec

The prevailed usage of neural networks and especially deep neural networks had inspired researchers to take advantage of them in learning word vectors which has not the drawbacks of high dimensionality. The initial of those models was developed by Mikolov et al (2013). They were using neural network for predicting context words and interestingly at the end, the weights that neural network has learned were representing words in vector space very precisely.

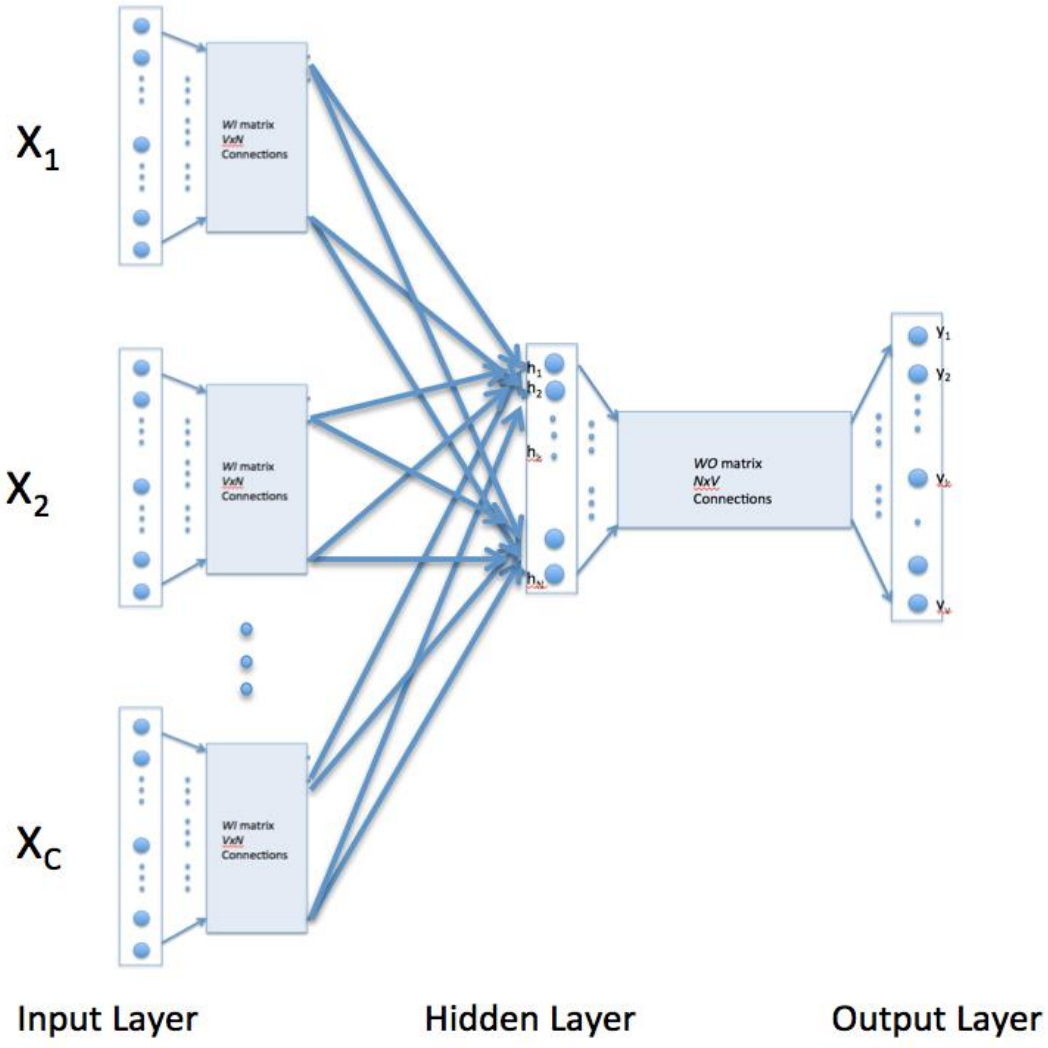

Figure 4: *The Architecture of Continuous Bag of Words Model*

Note here that, different from other machine learning techniques, in their approach, the output was of very less significance, but the most significant part was weights between neurons, which is used to represent each word. The architecture proposed by Mikolov et al (2013) was called word2vec and it has two different implementations based on the objective function: Continuous Bag-Of-Words (CBOW) and Skip-gram.

2.4.1 Continuous Bag-Of-Words (CBOW)

Continuous Bag-Of-Words (CBOW) is a neural network which takes m words from a given words context and tries to predict the target word (Mikolov et al., 2013a). While trying to predict the word, the weights of the neurons for the given word are adjusting themselves according to the context words. More specifically, this model tries to optimize for the given loss function:

$$E = -log(p(\overrightarrow{W_t})) \quad (1)$$

In equation (1), w_t represents the word that is to be predicted, and W_t is an array of words that is in the vicinity of the w_t. In other words, given context words array W as input, CBOW trains a neural network to optimize for the prediction of the target word.

In the input layer, m one hot encoded words are accepted and the output is one vocabulary size vector which represent the probability of the target word being i-th element in the vocabulary. Another advantage of this model architecture is its training speed. Mikolov et al (2013a) showed that the model training time is proportional to:

$$Q = N \times D \times D \times log_2(V) \quad (2)$$

where, N represents the number of the context words to be used as input, D is the dimensionality of the projection layer and V is the vocabulary size of the whole corpus. Considering the fact that, very big text corpara containing more than several billion words are used in training these models, we can conclude that training speed is one of the important factors.

Add picture

2.4.2 Skip-gram

Skip-gram model is essentially Continuous Bag-Of-Words (CBOW) model input and output exchanged. Meaning, this architecture tries to estimate the distribution of surrounding words given only the target word (Collados & Pilehvar, 2018). This new architecture has interesting properties. One of them being the more

surrounding words that this model tries to predict, the more the quality of the resulting word vectors at the end. Again, the time complexity is given by:

$$Q = C \times (D + D \times log_2(V)) \quad (3)$$

where, C is the longest distance between words, D is the dimensionality of the projection layer and V is the vocabulary size of the whole corpus.

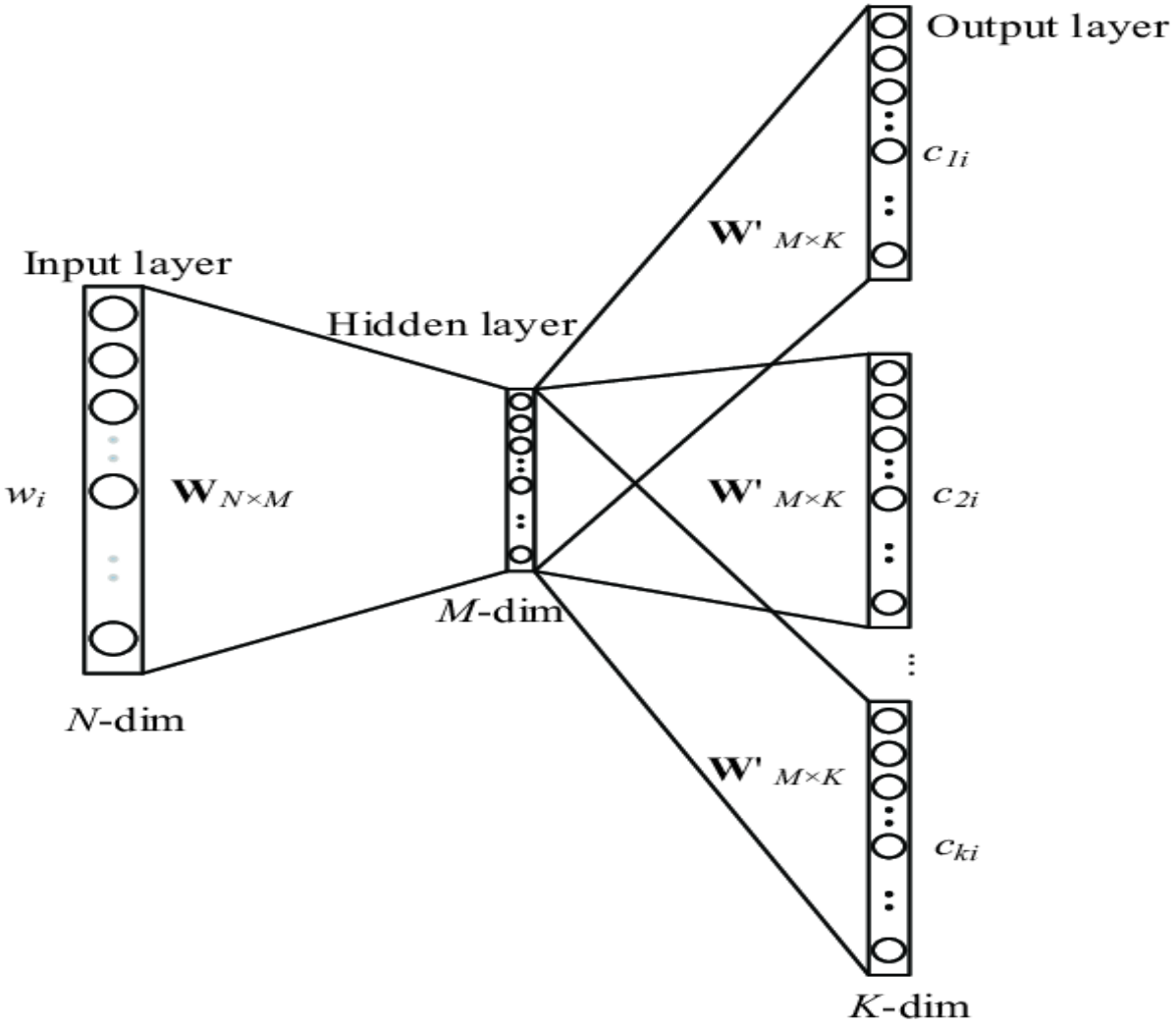

Figure 5: *The Architecture of Skip-gram Model*

Mikolov et al (2013d) demonstrated that vectors learned by these neural network architectures encode linguistic regularities inside them. The first type of those regularities is natural language's syntactic regularities. This includes base/comparative/superlative degrees of adjectives, singular, plural, possessive, non-possessive forms of common nouns and various tenses of both irregular and regular verbs (Mikolov et al., 2013d). One example of this can be shown as following; Let's take the word vector of the word "cities" and subtract from it the vector representation of "city". Then let's add this distance to the vector representation of word "man". If the neural network has been trained successfully, the above operations will yield the vector representation of "men" word. Additionally, it is worth noting that, every vector operation that is valid, such as vector addition, subtraction, vector cosine similarity and etc. are valid for word vectors which makes them very powerful in almost any domain of applications. This means we can tell if two words are similar by just measuring cosine similarity between their vector representations. Even more astonishing than that is the fact that, these word vectors also encodes semantic information inside them (Mikolov et al., 2013d). Semantic information is directly related to the meaning of the words, without considering syntactical and grammatical aspects. As an example, we can show famous king, queen relation. More specifically, if the vector representation for the word "woman" is subtracted from the vector representation of "man", and the resulted vector is added to the vector of the word "king", the output will be the vector representation of the word "queen".

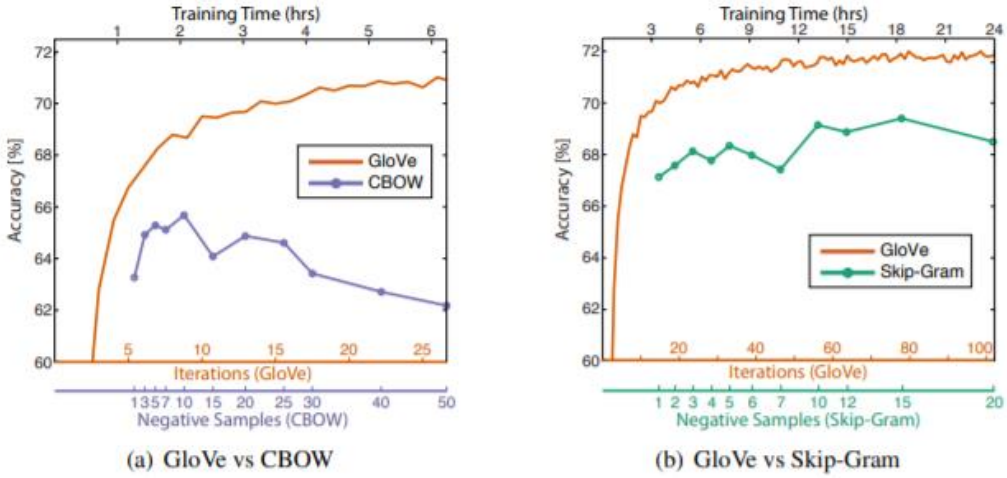

Figure 6: *Comparison of GloVe and word2vec models*

This once more proves that word2vec word vectors preserves semantic information about the words they vectorize. Levy and Goldberg (2014) later showed that skip-gram with negative sampling is an implicit factorization of pointwise mutual information matrix of target and context words.

2.5 GloVe

Pennington et al. (2014) introduced a novel technique for learning word embeddings. The model called GloVe for Global Vectors, combine two successful approaches, namely, global matrix factorization and the prediction of context words by neural networks approaches (Pennington et al., 2014).

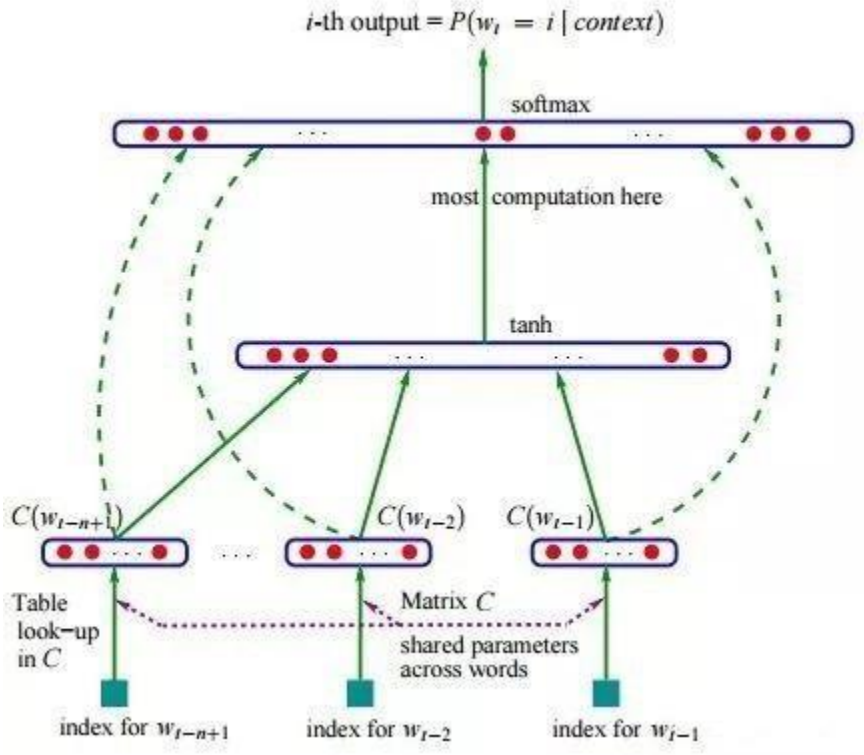

Figure 7: *Distributed Embeddings Learning*

This model also improved the score of word2vec approach on various word analogy tasks, by attaining 75.9% accuracy against the 68.4% accuracy of Continuous Bag-Of-Words model. The output of the model is 2 equivalent vector representation spaces for words. These spaces however, are not the because neural networks usually

are initialized randomly. The researchers also revealed that summing these two vector spaces generates a vector space which is less prone to overfitting and noise.

2.6 Contextualized Word Embeddings

Word embeddings discussed so far were all static, meaning the model learns the vector representation of each word from a very big text corpus and generates a vector space for the vocabulary and then these representations do not change unless we train the model again. Whenever we want to use these embeddings in downstream tasks we just take the vector each word and depending on the type of the natural language processing task perform operations over these vectors or even train a deep neural network architectures using these vectors as inputs. Consequently, we are stuck at using the same vector representation for the word for example “bank”, no matter in which context it is used. Collados & Pilehvar (2018) refer to this phenomenon as meaning conflation deficiency, in other words, the ineffectiveness of representing words with multiple meanings. Even for the monosemous words, sometimes it is desirable to have slightly different vectors depending on the context that this word is used. It seems that the main drawback of the statically generated word embeddings is that they do not consider the context that the word is used after training.

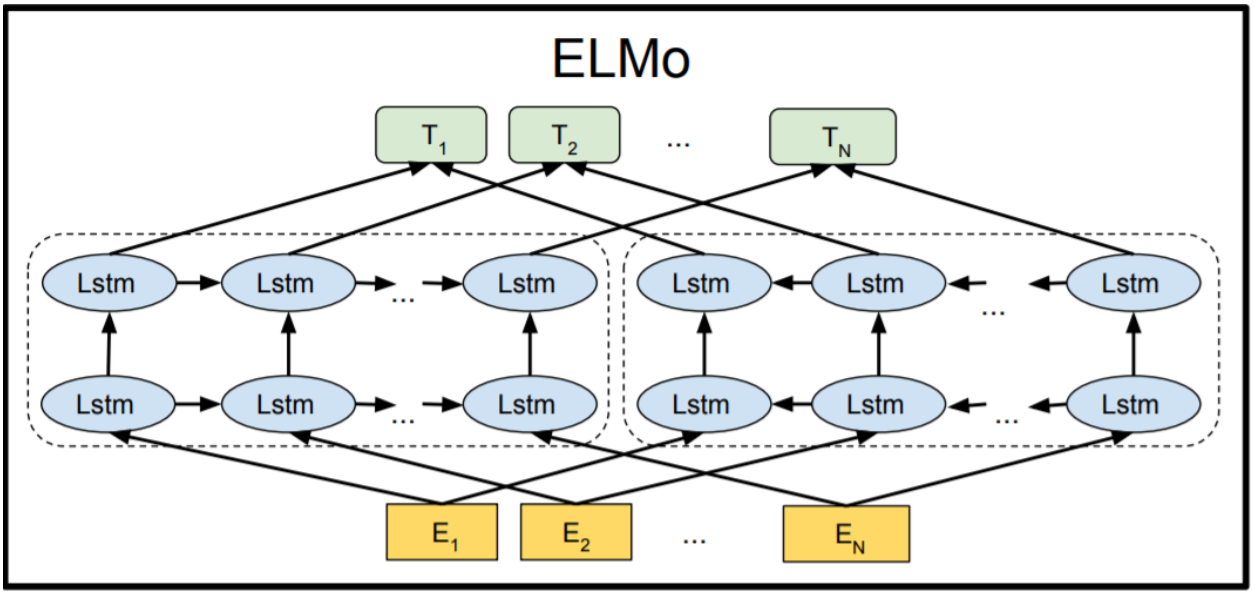

Figure 7: *ELMo Architecture*

This type of learning is useful as words can often have various degree of semantic depending on the place of use in a sentence or context. Figure 7 depicts an example model ELMo which is able to learn these types encodings in the context of the sentence. Thus it is desirable to have an architecture which even after training, can assign a vector representation to word based on the surrounding words as well. Hence, word will have different vectors each time it is used with different context. The architectures which aims at implementing this are called contextualized word embeddings.

Architecture	Main Topic	Strengths	Weakness	Novelty
Continuous Bag-Of-Words	Learning dense vector representation of words using big unlabeled text corpora and training neural networks for language modeling	Introduced and popularized the usage of neural networks for language representation.	Cannot capture the co-occurrence statistics of the whole corpus by taking only the local context	Using neural network for predicting the target word by taking context words of given window size
Skip-gram	Skip-gram model is Continuous Bag-Of-Words (CBOW) model input and output exchanged	Vectors learned by these neural network architectures encode linguistic regularities inside them	Implicitly factorizing pointwise mutual information matrix of target and context words	Using neural network for predicting the context words by taking target word
GloVe	By taking into account the statistical word frequencies while learning word embeddings can improve the quality of generated vectors	Generation of better word vectors, improvement on various NLP tasks	Not taking into account the dynamic context of the word while generating word vectors. These vectors are leant only once.	Combined two successful approaches, namely, global matrix factorization and the prediction of context words by neural networks approaches

Table 1. *Comparison between different architectures presented in Word Embeddings*

Table 1 summarizes different architectures. It pinpoints the strengths, weaknesses and as well as the novelty brought by the discussed model architectures.

2.6.1 Context2vec

Context2vec (Melamud, Goldberger, & Dagan, 2016) is a pioneering work in the natural language field which targeted generating contextualized representations for words. Context2vec uses bidirectional LSTM for generating context representation (Melamud, Goldberger, & Dagan, 2016). For any sentence S, the resulted representation is obtained by concatenating two vectors, one is generated by LSTM by going over words from left to right and the other is generated by LSTM by going from right to left:

$$biLS(w_{1:n}, i) = ILS(l_{1:i-1}) \oplus rLS(r_{n:i+1}) \quad (4)$$

where, lLS-stands for left to right LSTM and rLS-stands for right to left LSTM. Next, if we denote the context of a word by c, then the contextual representation can be expressed as:

$$\vec{c} = MLP(biLS(w_{1:n}, i)) \quad (5)$$

where MLP denotes multilayer perceptron and biLS denotes bidirectional LSTM vector representation of the context. This model architecture is able to learn embeddings for target words with different context lengths (Melamud, Goldberger, & Dagan, 2016).

2.6.2 BERT

As seen in the previous section, the architectures until Bidirectional Encoder Representations from Transformers (BERT) was using left to right and right to left LSTM or other neural network language models and then concatenating them to get the desired output. However, as language modeling is bidirectional it would be best to make the models bidirectional as well. However, there is two problems on the way. The first is in order to have proper probability distribution, directionality is needed. The second is more deep one, which is the possibility of target words being

visible to themselves in bidirectional encoders. The solution proposed by Devlin, Lee and Toutanova (2019) is to make some percentage of input words unrecognizable in other words masking. It is known that making the percentage of masked words very little increases the training time. And doing the reverse, namely, increasing the masking percentage leaves very little context for the model to learn from. In the proposed architecture, researchers masked the 15% percent of the input words randomly (Devlin et al., 2019).

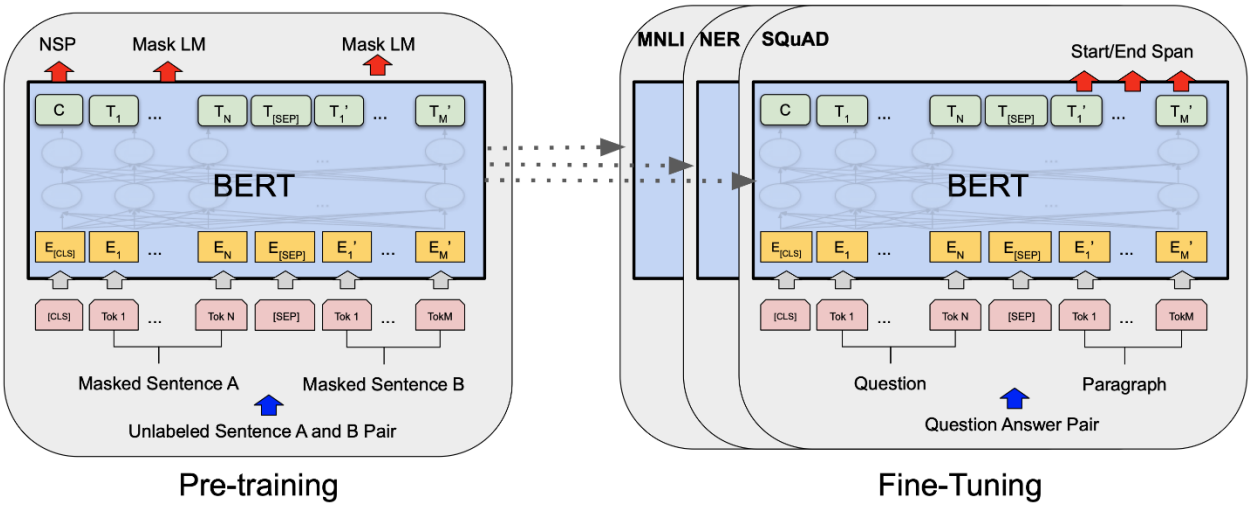

Figure 8: *BERT Learning Architecture*

Moreover, out of these 15% masked words, researchers replaced 10% with other random words. In some natural language processing tasks, not only the order of words, but also the order of sentences carries information. That is why, the proposed model also learns whether a given sentence follows the other one or not. The models architecture consists of multi-headed self-attention, feed forward layers and positional embeddings (Devlin et al., 2019). Moreover, the advantage of using Encoder architecture over using LSTM can be the effectiveness of Encoders in long-ranged word contexts.

2.6.3 Multi-Task Deep Neural Network (MT-DNN)

Multi-Task Deep Neural Network (MT-DNN) allows training on different NLU problems. Xiaodong Liu et al. (2019) proposed MT-DNN as an improvement over BERT architecture by enhancing the architecture suggested by Lui et al. (2015). The suggested model has the advantage of merging two successful unrelated approaches,

namely, multi-task learning and language model pre-training. The Multi-Task Deep Neural Networks are considered state-of-art models for language modeling which successfully pushed the GLUE score to 82.7%, meaning 2.2% improvement over previous state of art results (Xiaodong Liu et al., 2019). The model architecture consists of lexicon encoder, transformer encoder which allows to generate contextualized word embeddings using bidirectional Transformer neural network architecture. The main difference of Multi-Task Deep Neural Networks and BERT is that, the former is able to learn using two objectves; multi-task and pre-training at the same time, while the latter is able to only learn thorough pre-training (Xiaodong Liu et al., 2019).

2.7 Conclusion

Table 2 gives an overview of important evaluation parameters for language representation learning architectures:

Architecture	Training Speed	Global Matrix Factorization	Dynamic generation of embeddings depending on the context	Complexity of Neural Networks used for learning word representations
word2vec	√	-	-	-
GloVe	√	√	-	-
Context2vec	-	-	√	√
BERT	-	-	√	√

***Table 2**. Important evaluation parameters for language representation learning architectures*

As it is clear from the table, complex neural network based architectures with millions of parameters provide the ability to generate word embeddings dynamically depending on the context of the word used in the sentence. However, to be able to provide this dynamic word embedding generating models sacrifice training speed and complexity, meaning they took a lot more time and training resources to train.

The distinguishing feature of GloVe approach by Stanford University researchers is the fact that Glove takes into account the statistical word frequencies while learning word embeddings which as shown empirically improves the quality of generated vectors while not contributing too much to the complexity of the model. As a result, it shows itself on the generation of better word vectors, improvement on various NLP tasks. Architecture allowing dynamic generation of word embeddings dependent on the context enables most accuracy in encoding the meaning of given tokens and their relationships.

Chapter 3

Sentiment Analysis

3.1 Introduction

The success attained by machine learning and deep learning techniques recently in various tasks such as machine translation, computer vision and etc., has brought considerable attention to the application of these techniques to natural language and linguistics. Sentiment analysis is a branch of Natural Language Processing (NLP) which aims to determine particular attitudes and emotions from a given text by using machine learning and computational linguistics. As a result of prevailing social media trends and widespread use of electronic forms for collecting customer reviews, there has been remarkable growth in the number of written texts which contain subjective information. Collecting reviews as both grading scales and also as free texts allows customers to express their opinions more clearly using the wide capabilities of natural language. Although natural language is very convenient for expressing opinions and attitudes, its statistical analysis and accompanying computer processing retains a number of difficulties that require comprehensive research. A number of other factors such as the presence of exceptions in grammatical rules, the ambiguity of texts under different contexts, the use of irony in natural language, and the possible inclusion of both positive and negative attitudes inside one review are present, thereby making sentiment analysis a difficult task for machines. Machine learning techniques allow us to determine the sentiment of a given text more effectively. In the learning phase, machine learning models learn from a plethora of reviews that are labeled according to their sentiment category. Next, according to the parameters it learned in the learning phase, the model is able to predict the sentiment labels of reviews that it has never seen. In the scope of this thesis, sentiment analysis will be applied as an extrinsic evaluation task for evaluating the performance of generated word embeddings.

3.2 Machine learning and Sentiment Analysis

Machine learning, as a branch of artificial intelligence, is based on the idea that computer systems learn from data and can identify patterns in the data and then make determinations with little to no human involvement. Essentially, it is a type of data analysis which automates model creation. Sentiment analysis is a branch of Natural Language Processing (NLP) which aims to determine particular attitudes and emotions from a given text by using machine learning and computational linguistics. As a result of prevailing social media trends and widespread use of electronic forms for collecting customer reviews, there has been remarkable growth in the number of written texts which contain subjective information. Collecting reviews as both grading scales and also as free texts allow customers to express their opinions more clearly using the wide capabilities of natural language. Furthermore, it enables them to describe in more detail a particular service or product with which they are not satisfied. Although natural language is very convenient for expressing opinions and attitudes, its statistical analysis and accompanying computer processing retains a number of difficulties that require comprehensive research. A number of other factors such as the presence of exceptions in grammatical rules, the ambiguity of texts under different contexts, the use of irony in natural language, and the possible inclusion of both positive and negative attitudes inside one review are present thereby making sentiment analysis a difficult task for machines. Machine learning techniques allow us to determine the sentiment of a given text more effectively. In the learning phase, machine learning models learn from a plethora of reviews that are labeled according to their sentiment category. Next, according to the parameters learned in the learning phase, the model is able to predict the sentiment labels of reviews that it has never seen.

Sentiment analysis can be applied to text at two different levels, namely at the document level or at the sentence level. In a document-level analysis, the whole content of the user's review is taken and the machine learining model is trained on these reviews. On the other hand, in sentence-level analysis, user reviews are split into sentences and macine learining models learn from these sentences after each has been labeled according to their sentiment category. From now on we will be dscussing document-level sentiment analysis.

3.3 Learning Phase

Training a model for detecting the sentiment of a given text includes several steps. Initially, textual data from different sources are collected and only the ones which have sentiment polarity is selected. Then, each review is labeled as either positive, negative or neutral considering its sentiment. As machine learning model learns from the data, the correctness of the labels and the relevance of the training data plays a significant role in training a highly accurate machine learning model. In the next phase, text preprocessing procedures are appled to the labeled data. Text preprocessing includes lemmatiation or stemming of words, removal of highly frequent stop words, part of speech tagging and etc. As machine learning models are trained using data in numeric format, the next phase is vectorizing the text, in other words, converting textual data into a numeric format. The vectorization of textual data is considered to be an active research area and depending on the characteristics of the textual data at hand, one or another vectorization technique can lead to the highest accuracy. Subsequently, the vectorized text and labels are used to train varius machine learning models as seen in Figure 9.

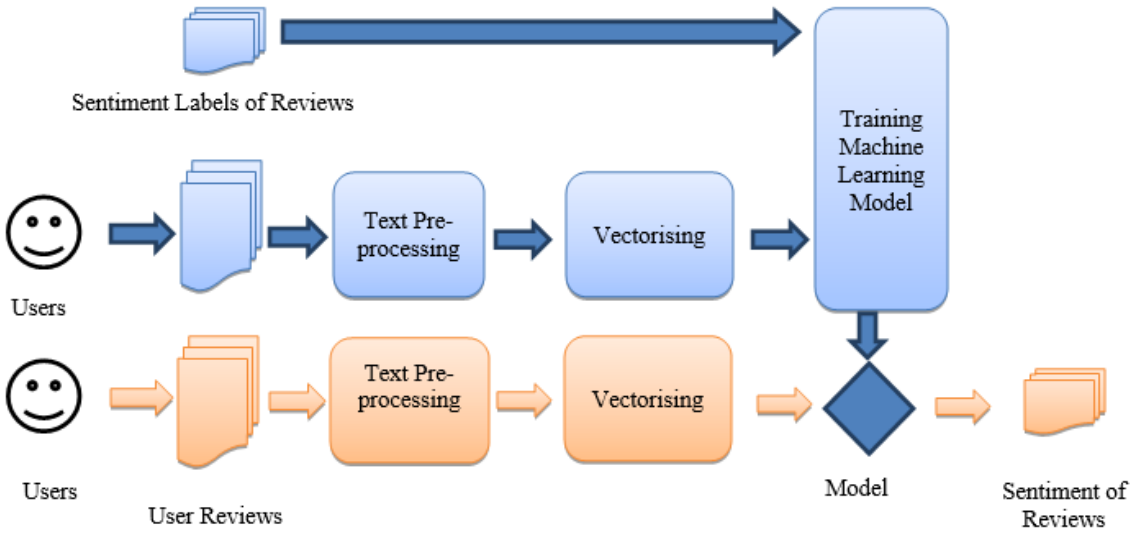

Figure 9: *Architecture of Machine Learning Model for Sentiment Analysis*

There are various approaches to measure how well a given machine learning model has learnt from the data. One of the most consistently used techniques is measuring the accuracy of the model. The question is how to measure the accuracy of a given model optimally? Let's assume we have 10000 user reviews for a particular product for each of which we know the sentiment category. We want to build a machine learning model on this data, so that the sentiment labels of all subsequent user reviews can be predicted by this model. In order to have a better model, we can train our model using 10000 whole labeled reviews and afterwards, select 1000 reviews and have our model predict their sentiment labels. As we already know the true sentiment labels of these 1000 reviews, we can calculate the accuracy of the model by dividing the number of correct predictions to the number of samples, namely 1000. This means, if the model correctly predicts the labels of 874 reviews, its accuracy will be 87.4 percent. However, our model has already seen and learnt from these 1000 reviews during its training phase, but it has not seen the reviews which will be written by users is the future which are the model's real test. In this case, we cannot state the true accuracy of the model regarding the data it has not seen yet. Therefore, what we need is the accuracy of the data from which our model has not learnt. The best way to measure this accuracy is splitting our 10000 samples into 9000 training samples from which our model will try to learn and 1000 test samples which our model will not see in the training phase. After the training phase is finished, we can use these 1000 sample reviews as true test cases and calculate the accuracy of the model.

3.3 'Chinese Room' Argument

Although there are numerous advanced methods for training a machine learning model and measuring how well it has learnt the task, there are also philosophical arguments against the idea of learning machines. First published in 1980 by philosopher John Searle, the "Chinese Room" argument, a philosophical thought experiment and commentary on the concept of strong AI, opposes the idea that machines could be programmed with intellectual abilities that are functionally equivalent to those of human beings. Searle held that it was impossible machine learning could develop a mind or consciousness, despite potentially seeming like it could think.

Searle imagines that research in the field of artificial intelligence has successfully crated a computer which behaves as if it understands the Chinese language, such that it can be given Chinese characters and then produces other Chinese characters answering the prompt. It does this so well that it passes the "Turing test" by consistently displaying responses which are able to convince a Chinese speaker that it is not a machine but rather a fellow Chinese speaker. Though this machine would pass the Turing test, Searle argues that the machine wouldn't actually understand Chinese, it would simply be simulating the ability to understand the language as it is simply following programming. To argue his point, Searle places himself in a closed room, with an English version of the computer program along with all of the other supplies he would need to correspond with the interlocutor. As messages are transferred through the door, he would process them according to the instructions and, similarly, produce messages in Chinese himself. Searle then argues that, if the computer was able to pass the test this way, that he would also be able to pass by running the program manually, without actually understanding or speaking the language. Like the computer in the thought experiment, Searle holds that he doesn't understand Chinese, as he was only following the scripted program he was given. Searle's argument rests on the idea that machines won't be able to develop understanding or intentionality and thus their "thinking," no matter how human it appears, is not actually the same. Machines lack a "mind" and thus, for Searle, the idea of strong AI fails. Though machines may be able to exhibit super-intelligent behavior due to more and more advanced programming, the Chinese

3.4 Introducing Inaccuracies in Human Language

Humans, however, rarely learn a language perfectly; mistakes do occur, even in speaking their first language. The mistakes of a first language typically differ from errors typically made during second language learning, which can be the result of a number of causes. Error analysis, the subfield of linguistics which examines the making of errors, has resulted in a number of studies examining the subject. Errors can be divided into two main categories, interlingual or those errors occurring at least in part because of the speaker's native language, or L1, and intralingual errors, those errors occurring due to the learning process itself.

Language, as the means by which human beings convey messages but also attitudes, irony, and emotions, represents a difficult task for machine learning. This is, in part, because there is frequently a mismatch in what is said and what is meant. Other factors adding to the difficulty for machine learning to conduct sentiment analysis are the presence of exceptions in rules, ambiguity due to different contextual factors, the use of irony and uncertainty or the inclusion of both positive and negatives attitudes inside the same expression.

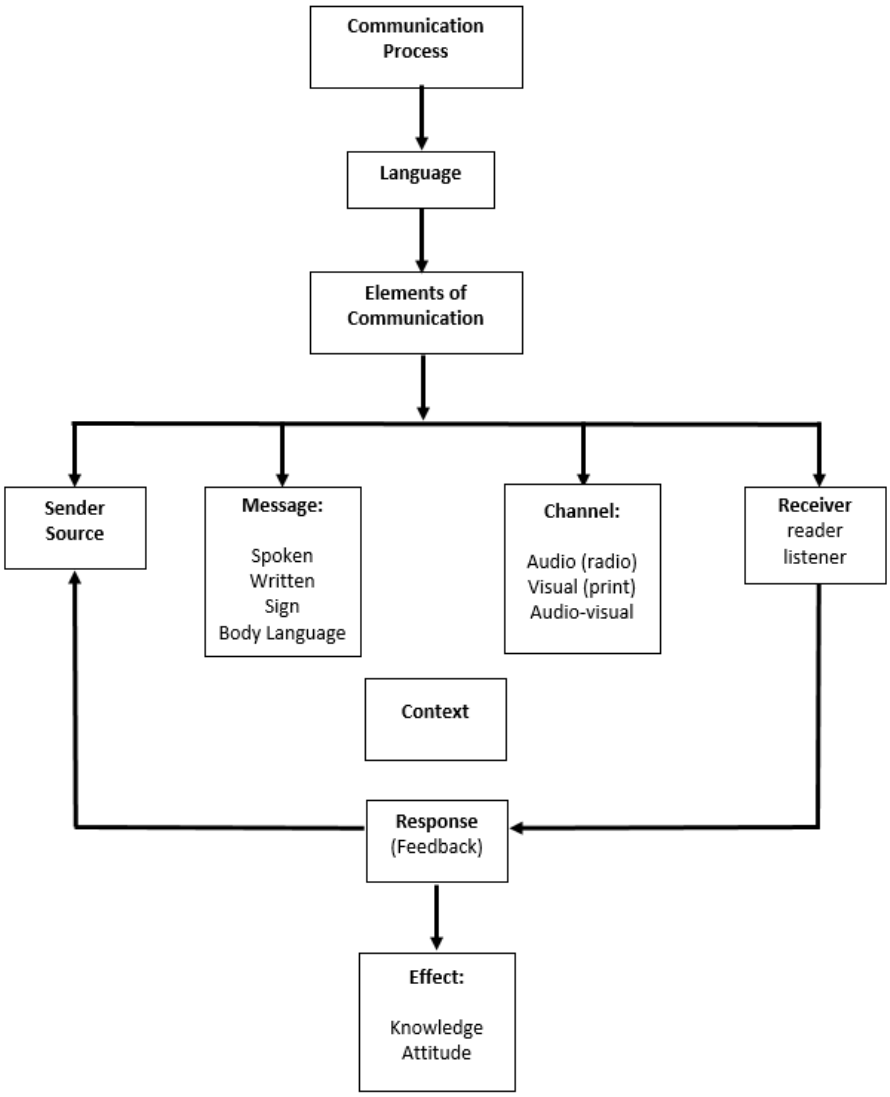

Figure 11: *Elements of Communication*

Different from the concept of machine learning, as machine learning models are not following a traditional computer program, but rather are given the parameters and then given data which it processes and learns from. If there are mistakes present

in the data, the machine learning model will learn and incorporate these mistakes, and, if the data is insufficient, the computer would learn imperfect Chinese, just as human beings, if they learn incorrectly, make errors.

This is how the machine in the Chinese room is taught Chinese. There is one point here, the machine learning model is never given instructions, namely, it is not following a computer program, it is given data and it learns from the data. If there are mistakes in the data, it will learn these mistakes; if the data is not enough, it will learn imperfect Chinese just as we, humans will. That is if left a human in a room with a book which has a lot of English-Chinese sentence pairs, human will also learn Chinese by using patterns and regularities. Given a new test sentence in English, human will generate a Chinese sentence.

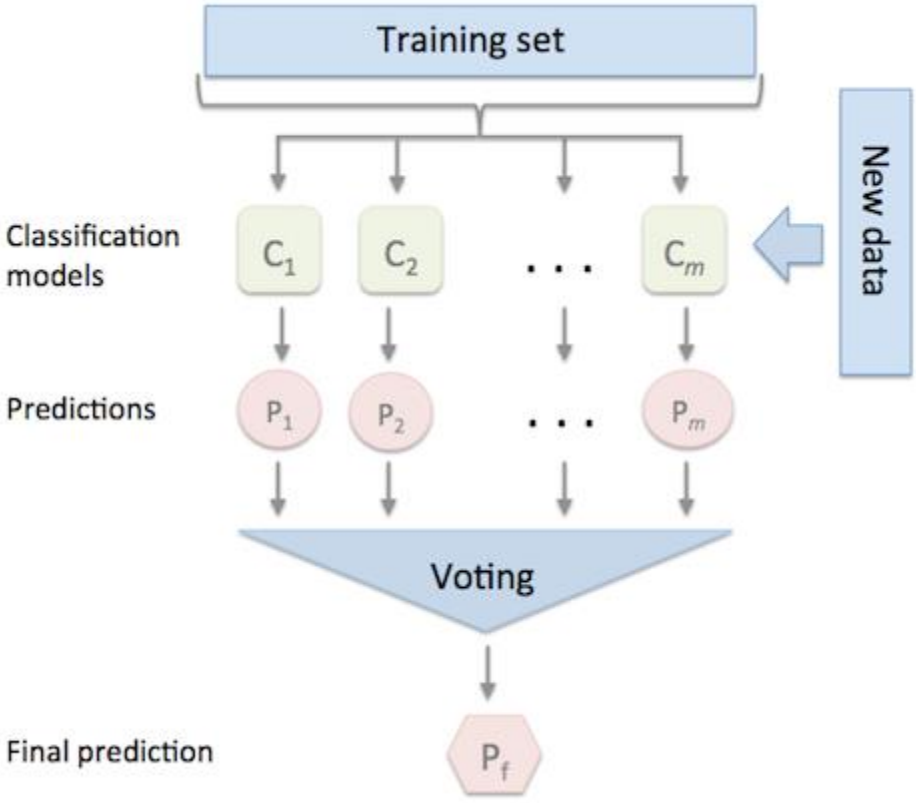

Figure 12: *Architecture of Machine Learning Model Using Ensemble Learning*

If we continue with the use of Chinese as our example, the challenge then is in measuring how much Chinese the machine actually knows. Though the situation outlined in the Chinese Room seems improbable, the concept is philosophically sound. However, a number of critics have proposed various replies, some of which have applicability to machine learning. The most common reply to the Chinese Room is referred to as the "Systems Reply". In this response, the entire system is used as an example of an entity which understands Chinese. While Searle himself may not understand the language, if the "entity" is seen as the room in its entirety along with the human and the necessities to communicate, then this means that the system itself understands the language.

There are numerous machine learning techniques based on various mathematical approaches. The accuracy of these models can be compared for determining which model performs best on the given dataset. Sometimes, the best result comes from ensemble learning, where different models learn together and the output is determined according to the majority voting amongst them. Ensemble learning, in the case of the Chinese Room would mean there was not a single person, but many. Therefore, once input was received, determining the output would come from a vote within the group with the response being determined by the reply with the most support.

3.5. Conclusion

In this section, we discussed machine learning techniques and the ideas of two renowned scientists namely Turing and Searle in the context of machine intelligence. Sentiment analysis is a task of machine learning where computers are required to determine the sentiment of given text. We discussed how machine learning is implemented to solve the problem of sentiment analysis. Moreover, Turing's belief was that it is possible to measure the intelligence of a machine by utilizing the human natural language in what he labeled as 'Imitation Game". The game, more akin to a test, required a computer to use written language to demonstrate enough intelligence, expressed through language, adequately enough to fool an average human subject. To measure this intelligence, Turing suggested concealing both a human and a machine before sending them written messages. The machine and the human could then respond to those messages however they would like in a communicative

process. Searle imagines that research in the field of artificial intelligence has successfully crated a computer which behaves as if it understands the Chinese language, such that it can even pass the “Turing test”, Searle argues that the machine wouldn’t actually understand Chinese, it would simply be simulating the ability to understand the language as it is simply following programming. We then discussed a counter argument to Searle’s idea, namely “Systems Reply” which assumes the entirety of the system understand the language not just parts of it.

Chapter 4

Implementation and Empirical Results

4.1 Introduction

In order to answer the research question and measure the quality of word embeddings empirically, very large data corpora for Azerbaijani have been formed and different machine learning architectures such as word2vec, fasttext have been trained. Word embedding vectors of various sizes have been generated and their scores have been measured on syntactic as well as semantic evaluation tasks. Accuracy score has been chosen as metric in evaluation tasks. As an extrinsic evaluation task, the generated word embeddings will be used in sentiment analysis task and the results will be shown. This section will describe the data corpora, trained machine learning architectures, their hyperparameters and comparison of their accuracy scores on various training settings. The datasets used for intrinsic and extrinsic evaluation will also be described in this section.

4.2 Data Corpus

For being able to generate word embedding vectors which truly captures the semantic as well as syntactic meaning and relationship of words, data corpus with hundreds of millions of tokens have been collected from various sources such as news articles, books, scientific articles, poems, novellas written in journalistic, scientific and literary style. Generating word embeddings requires very large text corpora dataset with millions of tokens. That is why for the implementation part very large text corpora dataset for Azerbaijani have been created. This large text corpus is formed as a result of parsing Azerbaijani written books, textbooks, novels, poems, and notes of different categories.

In the empirical experiments conducted for the thesis, the number of tokens collected and parsed in this very large text dataset numbered around 164 million tokens. Deep learning and natural language processing tasks requires this type of large scale data in order to get satisfactory results.

4.3 Data Pre-processing

To measure and conduct experiments with different dataset sizes, the main data corpus is divided into two datasets. The size of the first dataset is 55.5 million tokens and as this dataset is formed mainly by parsing Azerbaijani news articles, we named this dataset News dataset. The second dataset consists of approximately 109 million tokens, and this dataset is formed by parsing Azerbaijani textbooks, poems, fiction and nonfiction books, and therefore it is named Books dataset. And the main data corpus is combination of these two datasets containing in total 164 million tokens, and named as Combined Dataset.

Corpus Name	Token Count	Distinct Word Count	Content of corpus
1st corpus	55 540 859	643 470	News
2nd corpus	108 635 059	3514282	Books
3rd corpus	164 175 918	3802820	Combined

Table 3. *Dataset Descriptions*

Having 3 datasets with different sizes allows to conduct experiments with different sizes and to measure the effect of increasing dataset size on the performance of the models. During the training of one of the largest and state of art language model GPT-3, researchers Brown and et. al. has also partitioned their data corpora into smaller datasets, such as Common Crawl, Wikipedia, WebText and etc. The machine learning architecture they have built, is one of the largest language models ever built with 175 billion parameters. This gigantic model has been trained on almost all the text data available in internet, books, websites, blogs, forums and etc. GPT-3 achieved very strong performance on diverse tasks such as machine translation, question-answering and etc. without fine-tuning on the specific tasks, with an approach named zero-shot learning.

Below is violin plot visualization of sentence counts in News dataset.

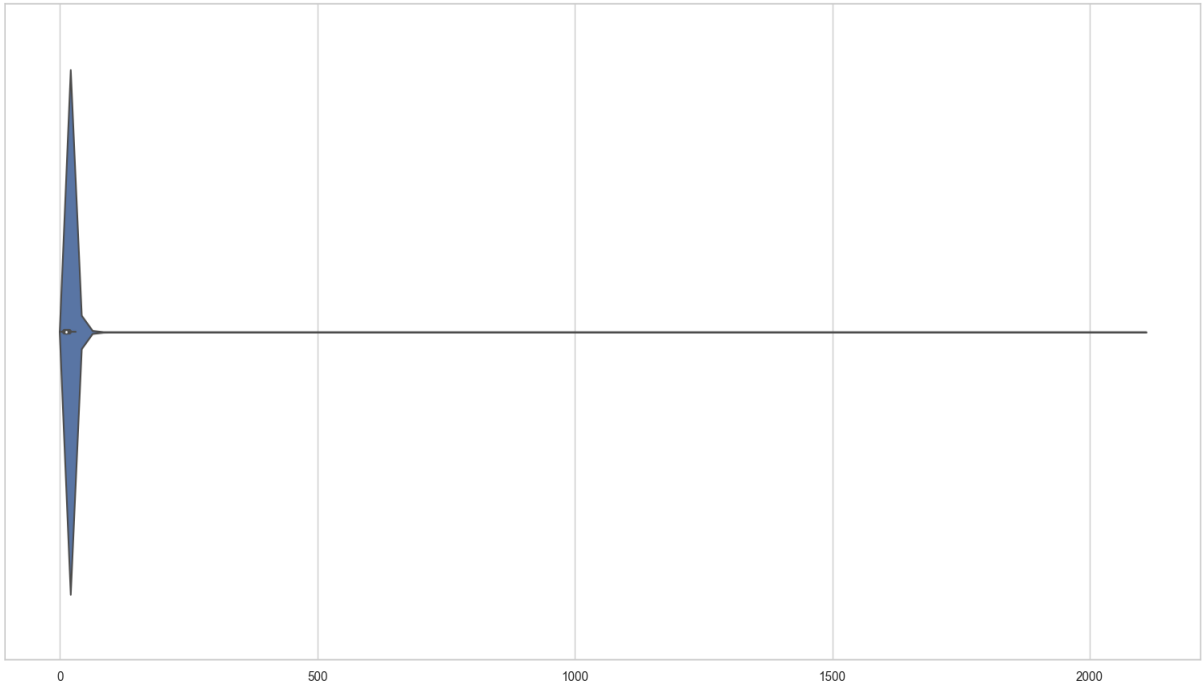

Figure 13: *Violin Plot Visualization of Sentence Counts in News Dataset*

By looking at Figure 13, we can see that there exists outliers in the right side of the graph. Moreover, distribution of the sentences in the graph is left skewed, meaning most of the sentences are short sentences with mostly token counts less than 100 tokens. This pattern is intuitively correct as in news articles we expect to find sentences with less than 100 words mostly. In order to apply appropriate preprocessing steps I decided to explore the dataset more and prepared the following descriptive statistics of the News dataset.

Corpus Name	**News Corpus**
Number of tokens	55 540 859
Distinct Word Count	643 470

Overall Sentence Count	3 854 116
Mean token count	14.41
Standard Deviation	8.41
25% percentile	9.00
50% percentile	12.00
75% percentile	18.00
max	2,109.00

Table 4. *Dataset Descriptive Statistics*

Several interesting points can be observed from the descriptive statistics given in Table 4. Firstly, we can see that the number of unique token count is 643 470 whereas the number of overall tokens is 55 540 859. Meaning, in the news dataset we have 643 470 distinct words, which is indicating statistically that Azerbaijani is an agglutinative language, namely a lot of new, distinct words are created by combining morphemes, prefix and postfixes to the root form of the words. The process itself is called agglutination. As machine learning based vector space representation approaches learns and generates distinct word vectors for each distinct word tokens, the research question seems even more appropriate; how effective the word embedding will preserve the syntactic and semantic relationship of tokens with agglutination, and will the relation be observed when using vector operations in high dimensional embedding space.

Another interesting point observed from descriptive statistics in Table 2 is about percentiles and median. The average length of sentences in News dataset is 14.41 words, with 75% of news articles having less than or equal to 18 words. We can observe the outliers having maximum length of 2109, therefore I decided to remove all the outliers having sentences more than 100 words. After removing outliers 99.9% of the data remained. And below Figure 14, depicts the distribution of the data after outlier removal.

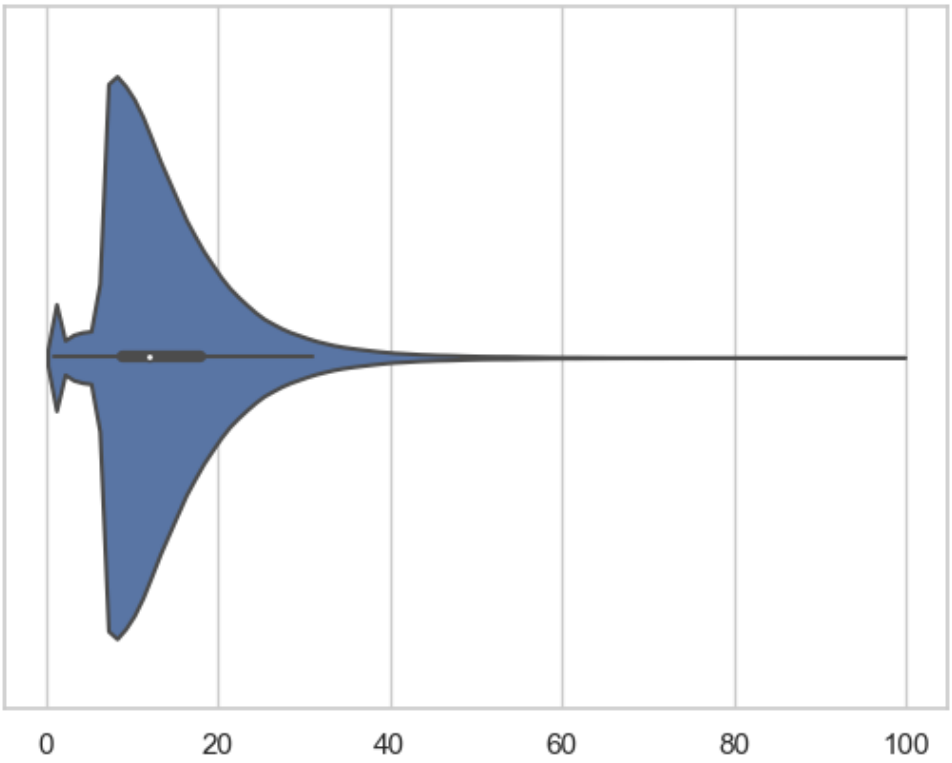

Figure 14: *Distribution of the Data after Outlier Removal*

The descriptive statistics for Books dataset is given below in Table 5. This dataset is formed mainly from sources such as books, scientific articles, poems, novellas, textbooks written in journalistic, scientific and literary style. Having data corpora with different writing styles is important because word embeddings are learnt from context by machine learning models and a word should be present in as many various contexts as possible in order to be able to learn accurate vector representation for that word. Moreover, word embeddings are generally learnt once and afterwards, learnt word vectors are used in different downstream tasks which can be from various domains. Therefore, in order to perform satisfactorily in tasks from different domains, it is beneficial to learn word embeddings from as much diverse sources and contexts as possible. Books dataset contains not only scientific styles for the machine learning model to learn but also contains fiction and nonfiction literature works which is close to daily language usage style.

Corpus Name	Books Corpus
Number of tokens	108 635 059
Distinct Word Count	3 514 282
Overall Sentence Count	17 215 783
Mean token count	6.31
Standard Deviation	65.17
75% percentile	5.00
max	6890.00

Table 5. *Dataset Descriptive Statistics*

The descriptive statistics for Books dataset given in Table 5 is also indicating the presence of outliers. Therefore, necessary outlier removal procedures have been applied to the Books dataset.

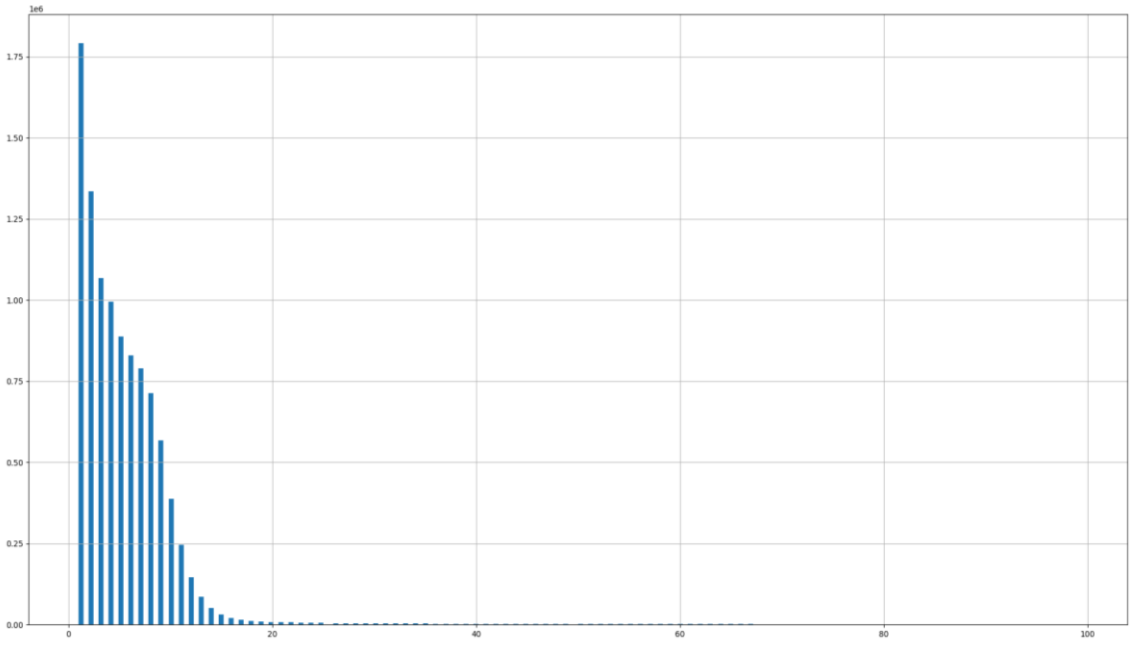

Figure 14: *Histogram of the Sentence Token Counts after Outlier Removal*

4.4 Intrinsic Evaluation Experiments

Word embeddings approaches aims at vectorising the word tokens in high dimensional vector space so that the semantic and syntactic meaning of words are encoded in the vectors. What does that mean? That means as a result of word ebeddings, we get one vector for each word. And these vectors encode the meaning of words in vector space. How do these vectors preserve the meaning of words? To explain it more explicitly let's explicate it on an example. For instance, let's show an example 50 dimensional word vector for the word "Bakı" (Baku) that is generated by us in the scope of this thesis:

```
array([ 0.04427981, -0.01703207, -0.7023115 , -0.17552207, -0.12401582,
       -0.23362264, -0.4472588 ,  0.4980079 ,  0.4010127 , -0.45398593,
        0.26293465, -0.57307976,  0.0377945 ,  0.0342879 , -0.04536356,
       -0.10185803,  0.32134488, -0.2125738 ,  0.61199456,  0.28370285,
        0.10053606,  0.9282368 , -0.1241583 , -0.41881907,  0.4221995 ,
        0.9906296 , -0.03054628, -0.13216494, -0.1286333 , -0.63236845,
        0.28124377, -0.15902765,  0.17776072, -0.09113472,  0.4791054 ,
        0.32985336,  0.5393359 ,  0.12777816,  0.31756562, -0.29991594,
       -0.10142764,  0.33389965, -0.28160125, -0.59327453,  0.34725884,
        0.23039858, -0.580794  ,  0.01770324, -0.53385746,  0.07866098],
      dtype=float32)
```

Figure 15: *Word Vector for the Word "Bakı" (Baku)*

As we can see we have vector of fifty real numbers, $v \in \mathbb{R}^{50}$ for the word "Bakı" (Baku). Word embeddings generates vector for each word. Let's take another word "Sumqayıt" (Sumgayit) another city of Azerbaijan:

```
array([-0.11583662, -0.18128696, -0.9891449 , -0.05143925,  0.09286433,
       -0.44098786, -0.10412007,  0.5027127 ,  0.35769895, -0.17526196,
        0.41654456, -0.93584317, -0.08847107, -0.1368799 , -0.11311584,
       -0.25519097, -0.0050356 , -0.37478158,  0.12255943, -0.06330165,
        0.37942642,  0.5055724 , -0.5446811 , -0.02403036,  0.35736835,
        1.2140739 , -0.52403295, -0.12159846, -0.52550757, -0.41677123,
       -0.04929043, -0.03967765,  0.06188106, -0.0964994 ,  0.4778679 ,
        0.7312138 ,  0.8848946 , -0.04759882,  0.3306458 , -0.3475847 ,
       -0.04108227,  0.2600318 , -0.1028622 , -0.46767274,  0.104118  ,
       -0.06570275, -0.5446721 , -0.13403416, -0.9723647 ,  0.1828169 ],
      dtype=float32)
```

Figure 16: *Word Vector for the Word "Sumqayıt" (*Sumgayit*)*

We know that both of these words are Azerbaijani city names, so these words are semantically close words. But we claim that word embeddings encode the meaning of words in vectors. So the vectors of these two words should be close, is it true? We know that we can calculate cosine similarity in order to determine similarity of two vectors. So, let's denote word vector for Baku as B and word vector for Sumgayit as S and calculate the cosine similarity of two above vectors for Baku and Sumgayit:

$$\cos(|B,S|) = \frac{B\ S}{\|B\|\ \|S\|} = 0.8196250200271606$$

So as we can see, the word vectors of Baku and Sumgayit is close, namely cosine similarity of these two vectors is 0.8196250200271606 which means the vectors are close as well. So as we can see, words which are close semantically, their vectors are also close. This one of the most astonishing characteristics of word embeddings; it is possible to do vector operations over word vectors and they preserve semantic meaning.

Let's think of a word which is completely different from the word Baku. For example, the word, "yaz" (spring). The word "yaz" (spring) is a season name and "Bakı" (Baku) is a city name, so the concepts are completely different. Let's, look at the word vector of "yaz" (spring).

```
array([-0.36285448,  0.18483354, -0.59953916, -0.00372236,  0.14859486,
        0.10220399, -0.47828498,  0.63415825, -0.51452917,  0.6520089 ,
        0.12769957, -0.03357286,  0.25700346, -0.69815874, -0.13221292,
       -0.01692516, -0.19212571,  0.23589605,  0.08457457,  0.29259962,
        0.3549149 , -0.18300714, -0.24609438, -0.07886773, -0.59280914,
       -0.5592991 , -0.12676093, -0.8738083 ,  0.4753873 , -0.16438718,
        0.8344018 , -0.13653673, -0.02788766, -0.55649066,  0.39981252,
       -0.15484698,  0.00382874,  0.1313495 ,  0.7891889 , -0.3864827 ,
        0.1692797 ,  0.44605657, -0.8221587 ,  0.09000088, -0.01626531,
        0.15983292,  0.65505844,  0.11190719, -1.5420232 , -0.11014554],
      dtype=float32)
```

Figure 16: *Word Vector for the Word "yaz" (spring)*

If we calculate the cosine similarity of word vectors "yaz" (spring) denoted as Y and "Bakı" (Baku) denoted as B:

$$\cos(|B,Y|) = \frac{B\ Y}{\|B\|\ \|Y\|} = 0.15399832$$

We can see that word vectors are not close with a similarity score of 0.15399832. Hence, words which are close semantically, their vectors are also close. Words which are not related to each other, their vectors are not similar. Therefore, word embeddings are considered as very powerful representation for human natural language.

So we showed that word embeddings generates word vectors so that words with close meaning and relationship have close vectors in vector space. Can we visualize this? Figure 17 presents the visualization of the word "Bakı" (Baku) and 10 closest word vectors to this word in 3 Dimensional space; $v \in \mathbb{R}^3$:

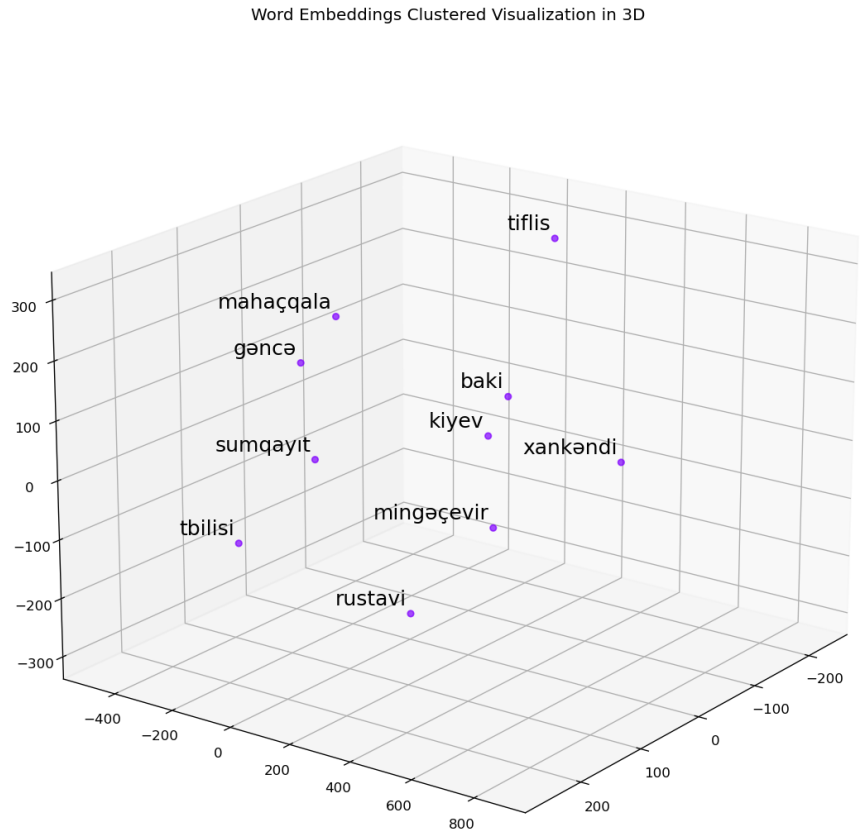

Figure 17: *Word Embedding model, Visualization of the word vector "Bakı" and top 10 closest vectors in 3D space*

As we can see top 10 closest vectors to the word "Baku" are all city names, such as "Kiyev" (Kiev), "Xankəndi" (Khankendi), "Sumqayıt" (Sumgayit) and etc. This is similar to human thought process as we humans also kind of group closely related

words in our brains. Therefore, we can say that word embeddings represents the meaning of words and human natural language mathematically. This is of course very long-waited and desired representation in the field of natural language processing. Because, these representations, more specifically, high dimensional word vectors can be used to train very successful machine learning and deep learning architectures. Having vectors which captures the meaning of words is revolutionary even so that algebraic operations over these vectors also preserve the meaning of words.

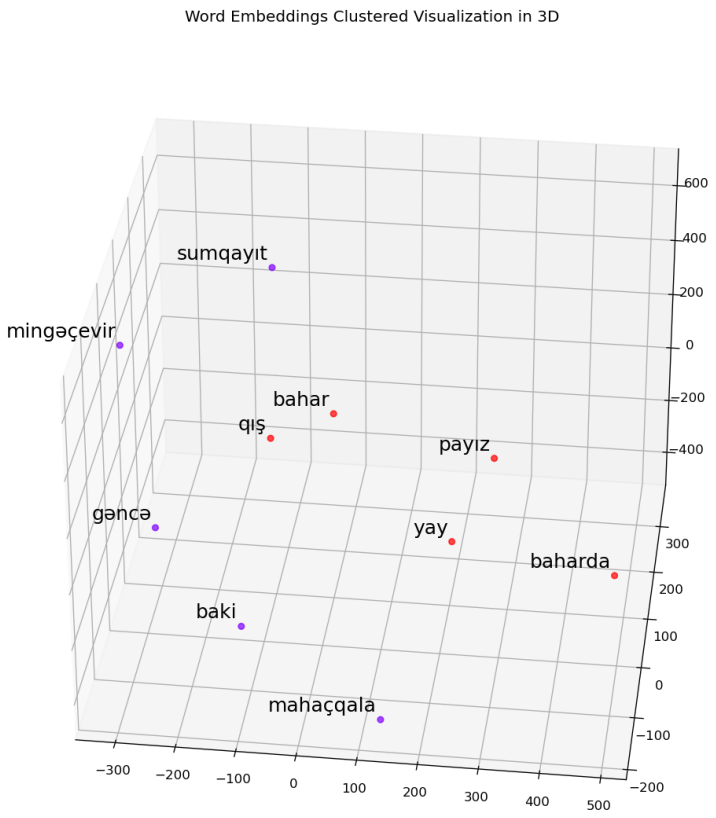

Figure 18: *Word Embedding model, Visualization of the word vectors "Bakı" (Baku) and "yaz" (spring) together with top 5 closest vectors to them in 3D space*

Figure 18 demostrates the power of word embeddings even more clearly. In the above figure, we visualize 2 unrelated words that we previously talked about "Bakı" (Baku) and "yaz" (spring) in the 3 dimensional vector space. We also visualize top 5 closest vectors to each of them. We can clearly see that season names "payız"

(autumn), “yay” (summer), “qış” (winter), and even the poetic, Persian rooted version of the word “bahar” (spring) having grouped together in vecotor space. And other city names together with “Bakı” (Baku) grouped together in vector space. We will talk more about visualizations of word embeddings in Section 4.3.

4.4.1 Experimental setup

Can word embedding vectors do more, will they preserve meaning if we add, subtract these vectors? If they preserve meaning, can they answer more complex analogy questions given in human natural language? It turns out the answer is yes and this is the root of intrinsic evaluation methods. Namely for the intrinsic evaluation, batches of analogy questions are used and the performance of the model is measured in terms of accuracy. If the model output for the given analogy task is the same as correct answer to the analogy, then question is assumed to be correctly answered and false otherwise. Before presenting intrinsic evaluation results, let’s explore the concept of analogy questions on one example:

What will be the answer of the following analogy question:

- ”girl” => “woman”,
- X => “man” , what is X?

The answer is “boy” by analogy, because “girl” is as similar to a “woman” as a “boy” is similar to a “man”. Can word embeddings answer this question? Can it predict that “boy” is as similar to “man” as “girl” is similar to “woman” by using vectors? It turns out that word embeddings capture the meaning and relationship of the words so well that it can answer the analogy questions. But how? By using algebraic operations with vectors. So, words are vectors now and we know that we can add, subtract do other operations with vectors. So what will happen if I add word vectors or subtract word vectors? If I formulate this question with vectors this can be like: what is “kişi” (man) – “qadın” (woman) + “qız” (girl)? So without vectos this question is analogy, given “girl” => “woman”, X =>”man”. If we ask what is X, by analogy the answer is “boy”. So with word embeddings we take, the word vector “qız” (girl) and subtract the word vector “qadın” (woman) and add to this the vector “kişi” (man).

vector (“qız” [girl]) - vector (“qadın” [woman]) + vector (“kişi” [man])

As a result, we get a new vector. And this vector amazingly turns out to be the answer of our analogy question namely, word vector for the word “oğlan” (boy).

A. vector (girl) - vector (woman) + vector (man) = vector(boy)

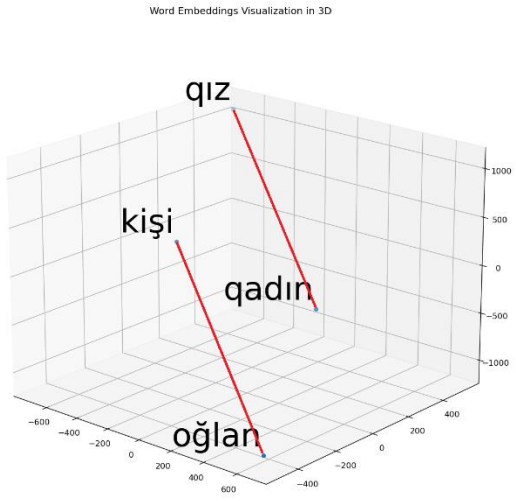

Figure 19: *Word Embedding model, Visualization of the vector operations for analogy A in 3D space*

Word embeddings is also able to encode very subtle semantic relationship between words. For example, we can use vectors and algebraic operations to find capital city, country relationships. A sample capital coutry semantic evaluation question is as follows. If for “Azərbaycan” (Azerbaijan) it is “Bakı” (Baku), then it must be “Moskva” (Moscow) for “Rusiya” (Russia). By vector notation:

B. vector(Azərbaycan) – vector(Bakı) + vector(Rusiya) = vector(“Moskva”)

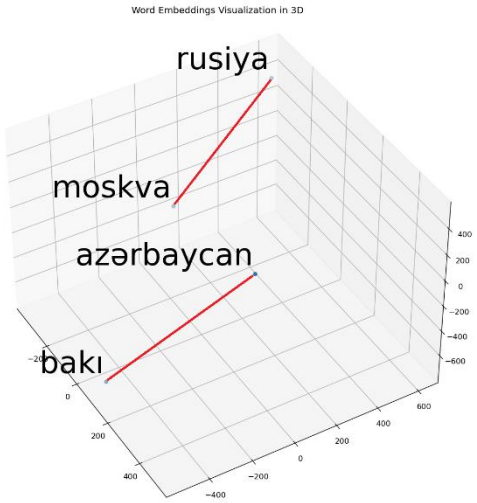

Figure 20: *Word Embedding model, Visualization of the vector operations for analogy B in 3D space*

Note that this only one instance of capital country semantic evaluation task for demonstation, for experimental measurements and results batches of these analogy questions are used and accuracy score is evaluated.

We can also find other semantic relations from word embeddins as well such as masculine feminine forms of words, lexical suffixes and etc. The fact that word embeddings architectures learn this relationships from only data corpus without any specific grammar rules is truely the distinguishing feature of them.

Word embeddings not only capture this type of semantic relationships, but also capture syntactic relationships as well. Syntactic relathions can be of the following form, “məktəb” (school) => “məktəbdən” (from school), “ev” (home) => X. Formulated in vector notation as follows:

C. vector(məktəbdən) – vector(məktəb) + vector(ev) = vector(“evdən”)

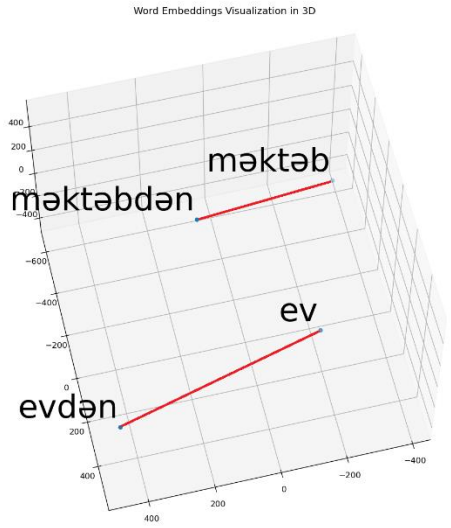

Figure 21: *Word Embedding model, Visualization of the vector operations for syntactic analogy C in 3D space*

Here, word embeddings generated by us truly answers the word vector “evdən” (from home). Note that this type of relation is valid only in Azerbaijani language because of agglutination. So, as novelty we will show how well word embeddings encode the syntactic relations with agglutination. We have prepared semantic and syntactic evaluation tasks specifically for Azerbaijani and the accuracy results of different word embedding architectures for Azerbaijani will be given in the next section in these intrinsic evaluation tasks.

For training and running experiments machine with the following parameters have been used. Processor is Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz 3.19 GHz, with an installed memory (RAM) of 16.0 GB (15.9 GB usable). System type is 64-bit Operating System, x64-based processor. Sytem has 12 logical processors over 6 main cores.

4.4.2 Evaluation Metrics

To evaluate the performance of machine learning models in experiments, scientific researchers employ various statistical rates, and metrics depending on the nature of experiments. As an evaluation metrics for the experiments we have used accuracy score. Accuracy score is used extensively in machine learning experiments as it is able to quantify the quality of predictions precisely.

For intrinsic evaluation experiments we have prepared extensive sets of analogy questions. For each analogy question in the batch, model predicts a word vector. The closest word vector to the predicted vector is calculated. If the word vector is the same as the answer of analogy, the prediction is considered true. In all other cases the prediction is considered false. At the end, the accuracy score is calculated by summing the number of all true predictions divided by the number of all predictions:

$$\frac{1}{n_{samples}} \sum_{i=0}^{n} 1(\acute{y}_i == y_i)$$

4.4.3 Intrinsic Evaluation Results

In this section we present the results of intrinsic evaluation for Azerbaijani word embeddings. As we mentioned earlier, word analogy tasks consisting of semantic and syntactic analogies are chosen for evaluating the performance of the generated word embeddings for Azerbaijani. In the following chapters we present the results for each architecture and provide detailed analysis.

4.5 CBOW Architecture

Below table shows the Continuous Bag of Words architecture's results on word analogy tasks given as percent accuracy. Highest accuracy scores are shown in bold, underlined format.

Vector Dim.	Size	Accuracy Score (out of 1)		
		Syntactic	Semantic	Overall
300	55M	64.5	36.2	50.3
300	108M	77.2	45.5	61.3
300	**164M**	**83.9**	**45.5**	**64.7**

Table 6. *Intrinsic Evaluation results for CBOW architecture*

As we can observe from the table, the accuracy of model increases in accordance with the size. And the model attains its highest accuracy with the largest dataset size available with the size of 164 million. This indicates that having even larger dataset could contribute increasing model accuracy. Surely, the relevance of the dataset and its structure still plays huge role. Highest accuracy score achieved on syntactic task is 83.9 %, while for the semantic task the score is 45.5 %. We can see the trend that CBOW architecture generally does better in capturing syntactic relationships, therefore have higher scores in syntactic evaluation tasks. Note that intrinsic analogy questions considered very hard task in machine learning. For reference, the score of the model for English language trained by Mikolov and et al. in semantic analogy task is 57.3 %. And for syntactic analogy task, the score for English language is 68.9%. Thus overall score for English is 63.7%. For Azerbaijani, we have 64.7% accuracy.

4.5.1 Model Analysis: Vector Length

Word embedings are dense vector representations of the words which aims to encode the meaning of words in vectors. Encoded word vectors can be of different dimensions for example 50 dimensions, 100 dimensions, 500 dimensions and etc. Vector dimensions plays a huge role in the quality of the word vectors generated. Two dimensional encoded word vector most probably carries less information than

let's say 50 dimensional word vector. Therefore, we present and analyze the intrinsic evaluation scores for different vector dimensionalities.

Vector Dim.	Size	Accuracy Score (out of 1)		
		Syntactic	Semantic	Overall
50	164M	61.3	28.8	45
100	164M	72.6	39.4	56
200	164M	82.3	40.9	61.6
300	**164M**	**83.9**	**45.5**	**64.7**
500	164M	80.6	39.4	60

Table 6. *Comparison of Intrinsic Evaluation results for CBOW architecture for Combined data corpora*

In table 6, comparison of accuracy scores with different choice of word vector dimensionality is given for CBOW architecture trained on combined data corpora. The results are indicating that increasing vector dimensionality generally increases the accuracy score. This can be explained by the fact that vectors with higher dimensionality can preserve more information. Having extra dimensions for storing information allows the architecture to learn even more subtle semantic and syntactic relationships of words. Therefore, word vectors with high dimensions have more capability to capture the meaning of words successfully. The score however goes

slightly down when we increase the vector dimensionality to 500 dimensions.

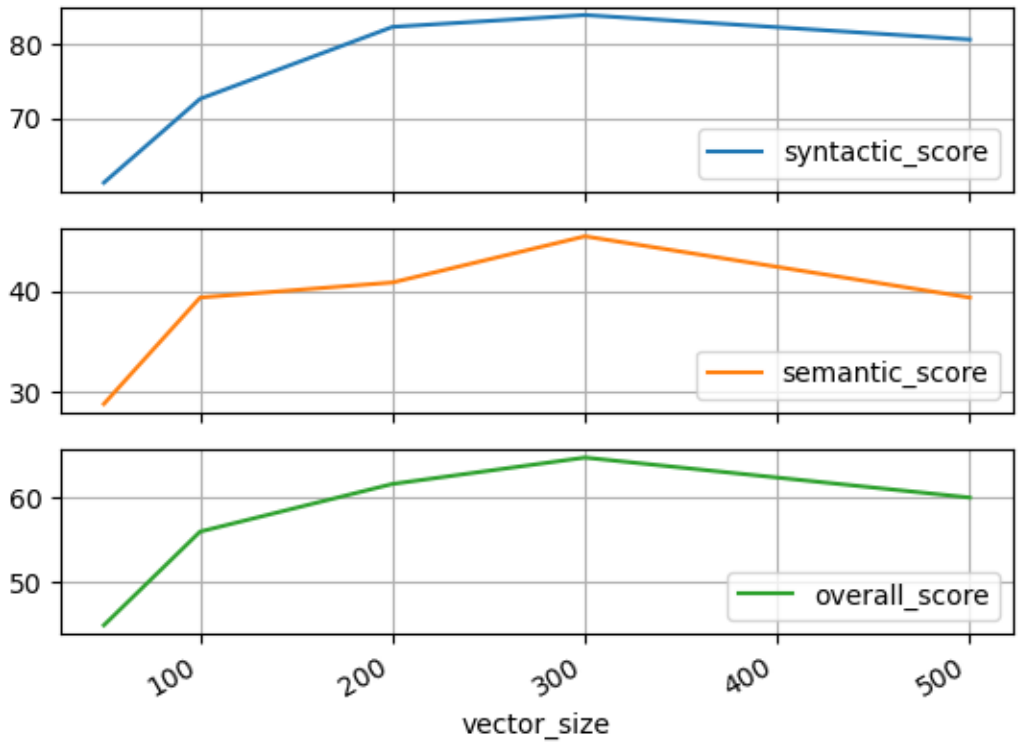

Figure 22: *Intrinsic Evaluation results at various vector sizes*

We can observe the relationship between increasing vector dimensionality and accuracy scores in the Figure 22, where we visualize the corresponding accuracy scores obtained from using word vectors for various vector dimensionalities. From Figure 22, we can see that model learns syntactic relations rapidly. Meaning word vectors with lower dimensionality such as 100 dimensions can still capture syntactic relations. However, for learning semantic relations higher vector dimensionality is needed. As we can observe from the graph, the accuracy score for semantic task is growing sharply until vector dimensionality of 300. This is because semantic meaning and relationship of words are more difficult to learn and they require more information to be encoded in the vectors. As vectors with higher dimensionality can encode more information, they are more successful at capturing semantic relationships.

4.5.2 Model Analysis: Corpus Size

Having comprehensive and large scale data corpora plays a huge rule in achieving high accuracies, as language models needs to learn the vector representations of words from a lot of different contexts. In this section we analyze the effect of corpus size on the intrinsic evaluation accuracies.

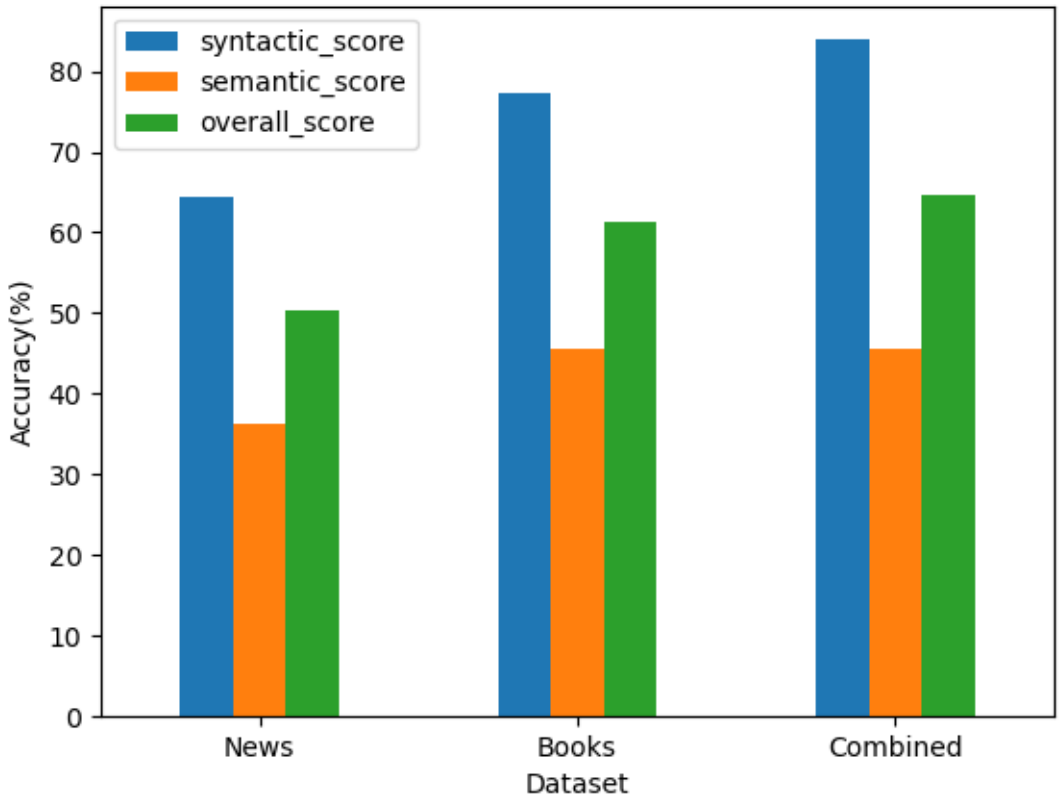

Figure 23: *Accuracy results on analogy tasks for 300 dimensional word vectors for CBOW architecture trained on datasets of different size*

Above table visualizes the syntactic, semantic and overall accuracy scores of CBOW architecture for word vectors of 300 dimensionalities on the datasets of different sizes. News dataset is of size 55 million token, while Books and Combined datasets have 108 million and 164 million tokens respectively. Here, we can observe

that the accuracy score increases as the dataset size is increasing. The explanation is that in larger datasets model sees words in a lot more contexts allowing it to learn more accurate vector representation for the words.

4.6 FastText Architecture

FastText architecture allows accounting for sub word information by using character n-grams. This is very useful for Azerbaijani language, as it is an agglutinative language. Being an agglutinative language makes handling sub word structures even more important as most of the syntactic relations are at sub word level. Thus, we can expect that this architecture will have higher syntactic accuracy score in intrinsic evaluation tasks. In the Table 7 below, we report detailed accuracy scores for fastText architecture, with word vector dimensionality of various sizes trained on 3 types of datasets. Highest scores for each dataset size grouping is shown as bold and underlined. Vector dimension represents the number of word vector dimensions learnt by the model.

Vector Dim.	Size	Accuracy Score (out of 1)		
		Syntactic	Semantic	Overall
50	108M	78.9	36.4	57.7
100	108M	84.2	**39.4**	**61.8**
200	108M	84.2	34.8	59.5
300	108M	**86**	34.8	60.4
500	108M	84.2	25.8	55

Table 7. *Intrinsic Evaluation results for fastText architecture*

We observe that sub word information improves the accuracy score for syntactic tasks, while learning these types of relations even for smaller vector dimensions. In contrast, we observe that enriching word embedding vectors with character level information did not contribute to semantic evaluation score, it even slightly decreased it.

4.7 Model Analysis: Comparison of word embedding models

We showed that fastText architecture achieves higher accuracies in syntactic analogy questions, while for semantic questions the performance is worse. In this section, we present the comparison of two architectures. We keep window and dataset size static and report the overall accuracy of these two architectures trained on Books Dataset.

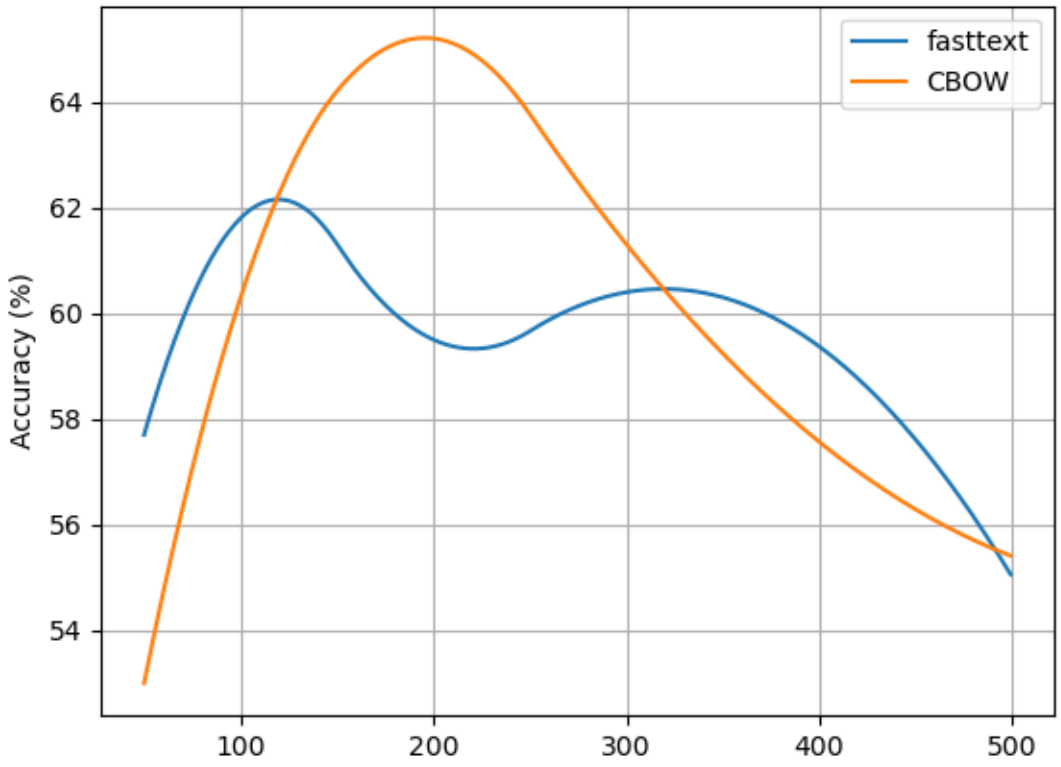

Figure 23: *Accuracy of CBOW and fastText architectures trained on Books Dataset as a function of vector size*

In Figure 23, we visualize the comparison of fastText and CBOW architectures. We present accuracy scores of fastText and CBOW architectures as a function of vector size. As it is observed from the graph, CBOW architecture attains more overall accuracy score than fastText architecture. We can observe that for smaller vector sizes fastText learns the word vector representations more successfully. However, after vector size 120, CBOW begins to dominate fastText architecture till vector size of 300.

4.8 Extrinsic Evaluation Experiments

After training on large scale datasets, word embeddings can be used in downstream natural language processing tasks. Downstream natural language processing tasks can be from different domain than the word embedings were trained on, hence the name extrinsic. Extrinsic evaluation is another technique for measuring the quality of word vectors produced by word embeddings architectures. To conduct the experiments, word vectors are taken and each word token is projected to a vector for further processing. The layer vectorising word tokens is frequently called Embedding layer, after which dropout, convolution, LSTM, dense and pooling layers are added. Below, we present details of the machine learning architecture, that we have built.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 100, 300)          32691600
_________________________________________________________________
dropout_7 (Dropout)          (None, 100, 300)          0
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 96, 64)            96064
_________________________________________________________________
max_pooling1d_7 MaxPooling1  (None, 24, 64)            0
_________________________________________________________________
lstm_14 (LSTM)               (None, 300)               438000
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 301
=================================================================
Total parameters: 33,225,965
```

Figure 24: *Architecture of our machine learning model*

Architecture learns task specific parameters through iterations over bathes of data. The tasks can be different. For extrinsic evaluation, we have chosen sentiment analysis task. In this task, given social article, model should predict the sentiment of the article. Model is first trained on labeled social news articles, and then its score is measured on hold-out validation data.

4.8.1 Evaluation Metrics

To evaluate the performance of machine learning models in experiments, scientific researchers employ various statistical rates, and metrics depending on the nature of experiments. Evaluation metrics for extrinsic evaluation tasks are determined depending on the extrinsic evaluation task. For instance, BLUE score can be applied for measuring the quality of machine translation tasks. For sentiment analysis, the sentiment label is predicted for each article and we employed accuracy score to compare the grand truth label with predicted label.

Machine learning model that we have trained predicts the sentiment label of each social news article in validation set. For each social news article in the batch, model predicts a sentiment label. If the predicted sentiment label is the same as the grand truth label, the prediction is considered true. If the prediction is different from grand truth label, the prediction is considered false. The accuracy score is calculated by summing the number of all true predictions divided by the number of all test in the validation set:

$$\frac{1}{n_{samples}} \sum_{i=0}^{n} 1(\acute{y}_i == y_i)$$

4.8.2 Sentiment Analysis Dataset Description

For sentiment analysis task, the dataset is relatively balanced dataset, negative social news articles constituting 62.6% of all news articles in the dataset. Note that, the same distribution is used for test set, in order to evaluate the performance of the model precisely.

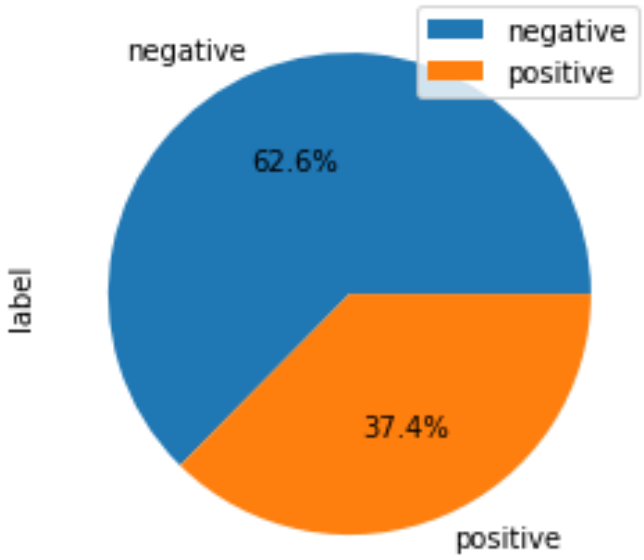

Figure 25: *Distribution of sentiment labels in sentiment analysis dataset*

4.8.3 Sentiment Analysis Implementation Results

Word vectors generated by word embeddings architectures are applied to sentiment analysis task. The score of the machine learning architecture reflects the quality of word vectors. The better the word embedding vector, the better model is able to learn the extrinsic evaluation task. We have used frozen embedding vectors as embedding layer in sentiment analysis task and trained our machine learning architecture. Our machine learning architecture consists of Embedding Layer, Dropout layer for regularization, 96 x 64 Convolution Layer, MaxPooling Layer, LSTM layer and fully connected dense layer. The architecture has overall 33 225 965 parameters. More detailed view of our architecture is given in Figure 24.

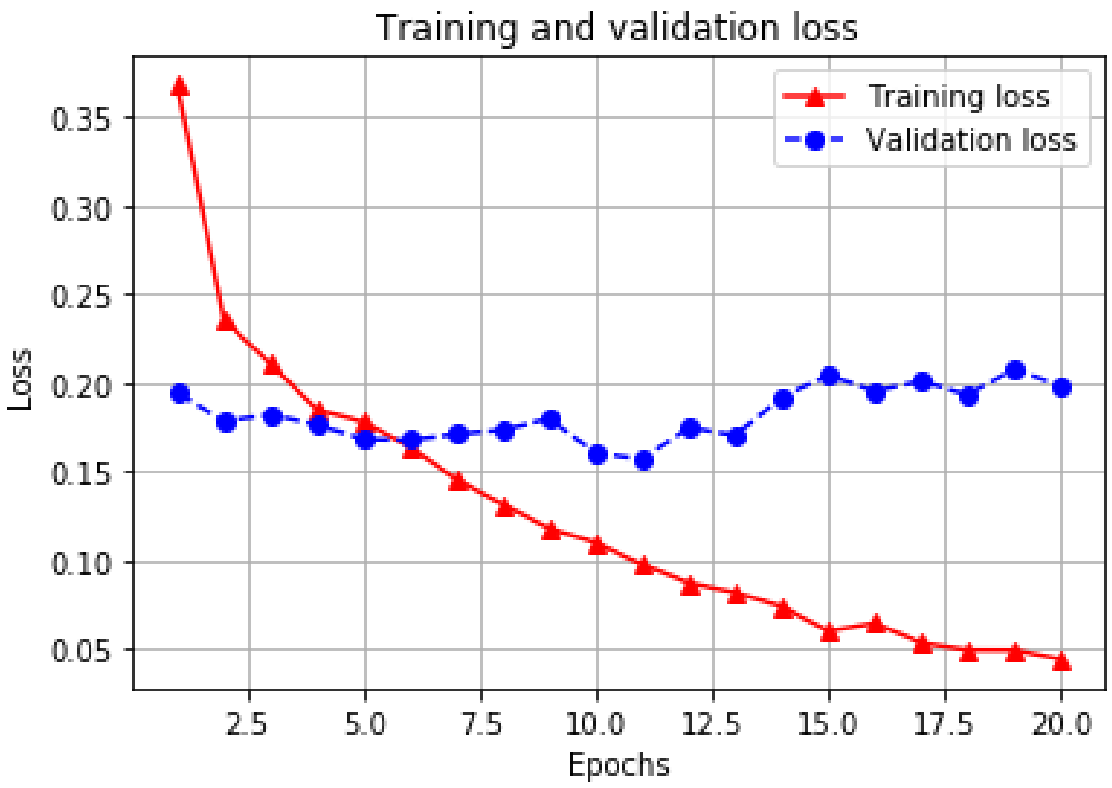

Figure 26: *Training and Validation loss of the model as a function of epochs*

In the above figure, we present the training and validation loss of the model architecture as a function of epochs. As we can observe the training loss is decreasing as we increase the number of epochs, while validation loss has a static pattern around 0.20. This means that the model has learnt from the dataset sufficiently, and epoch count of 20 is ample. Thus, increasing epoch counts further will lead to overfitting. Having higher validation loss than training loss is normal as model has not seen the validation data, and consequently, validation loss is naturally higher than training loss. We also observe that the values for validation and training loss is close, meaning, the model has learnt the pattern successfully on training set. Therefore, for the test set that model has never learnt, it has approximately close validation loss. Monotonic steady decrease on the training loss indicates that the model has adequate learning capacity, with the given number of learning parameters of approximately 33 million parameters.

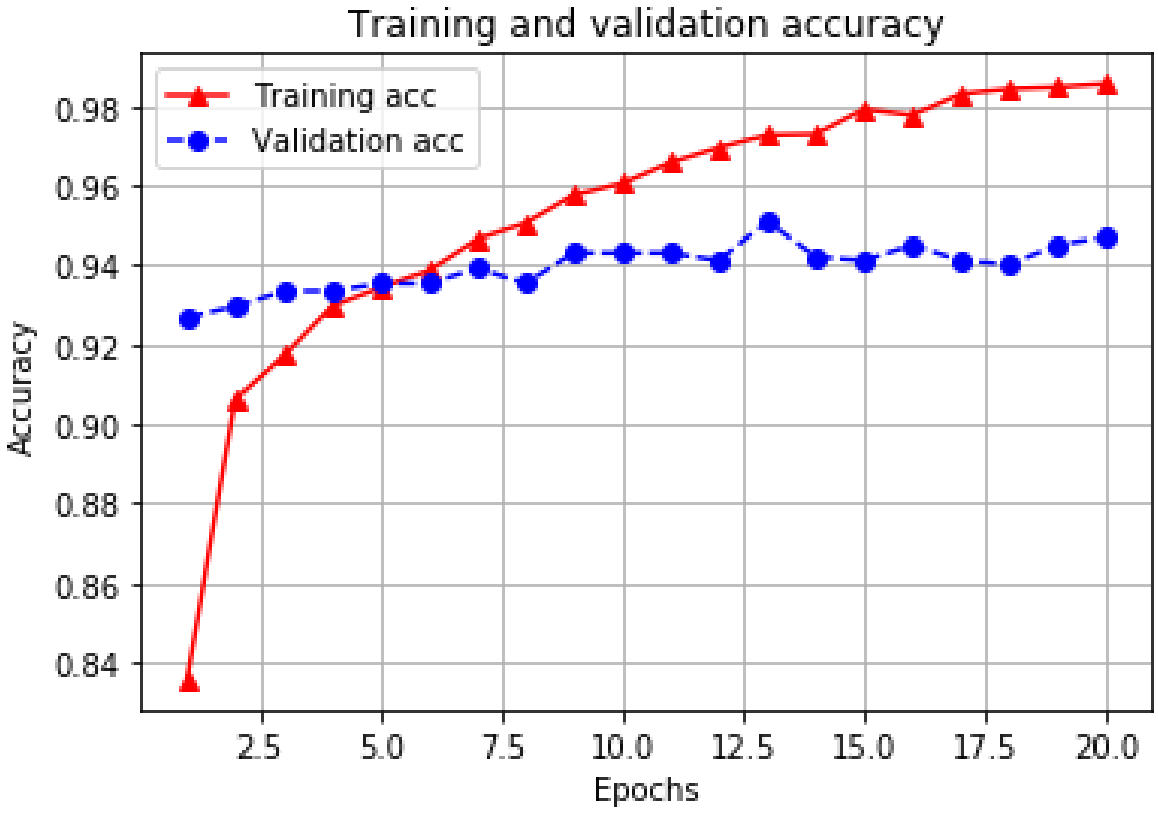

Figure 27: *Training and Validation accuracy of the model as a function of epochs*

In the Figure 27, we visualize the accuracy scores of our model on sentiment analysis task given as a function of number of epochs. For test set our architecture have achieved 95.36% accuracy, which shows that word vectors have successfully encoded the semantic meaning of words. Word vectors generated by fastText model architecture have reached 94.38% accuracy score. In the above figure, we also observe steady increase in validation accuracy. While validation accuracy has steady pattern 93-95% percent accuracy intervals. Training accuracy begins from 84% accuracy and increases sharply to 90% percent accuracy on 2 second epoch. Afterword's, validation accuracy follows a gradually growing pattern. Peaking above 98% accuracy. At the same time, test accuracy follows progressively increasing pattern. We observe a sudden spike at epoch 13, following the peak, the accuracy score continues fluctuation pattern till epoch 20, oscillating between 94-96

percent accuracy intervals. The final accuracy score of our architecture on test set is 95.36 percent. We conclude that word embeddings are successful representation for the words in the extrinsic evaluation task of sentiment analysis.

Chapter 5

Discussion and Conclusion

5.1 Discussion

This thesis paper explored natural language representation approaches from traditional vector space models to very recent word embedding and pre-training of deep bidirectional transformer architectures. In the scope of this thesis various machine learning architectures have been built and word embeddings have been generated for Azerbaijani language. We have used very large text corpora for generating these embeddings. We showed architecture details, as well as their advantages over previous techniques for each new approach. Traditional vector space models, on top of which recent approaches have been built, were discussed in section 2. We talked about their underlying principles and the areas where these approaches needed improvements. Furthermore, we talked about word vector representation approaches, which attracted a lot of attention from researches in recent years for their success in various natural language processing tasks. We also discussed syntactic as well as semantic linguistic regularities encoded by these word embedding vectors. To evaluate the performance of constructed word embeddings, we introduced semantic and syntactic evaluation experiments, and reported the accuracy scores in intrinsic evaluation section of Results and Implementation. We observed that CBOW architecture successfully preserved the semantic and syntactic relations with an overall accuracy score of 64.7% in analogy tasks. Note that intrinsic analogy questions are considered very hard tasks in machine learning. For reference, the score of the model for English language trained by Mikolov and et al. in semantic analogy task is 57.3 %. And for syntactic analogy task, the score for English language is 68.9%. Thus overall score for English is 63.7%. For Azerbaijani, we have achieved 64.7% accuracy.

The analysis of comparison of accuracy scores with different choice of word vector dimensionality indicates that increasing vector dimensionality generally increases the accuracy score. This can be explained by the fact that vectors with higher dimensionality can preserve more information and therefore have more

capability to encode the meaning of words. Having extra dimensions for storing information allows the architecture to learn even more subtle semantic and syntactic relationships of words. Therefore, word vectors with high dimensions have more capability to capture the meaning of words successfully.

The fact that language models require massive amount of textual data is a well-known fact in research community. Having comprehensive and large scale data corpora plays a huge rule in achieving high accuracies, as language models need to learn the vector representations of words from a lot of different contexts. To measure the effect of dataset size, we conducted the experiments with datasets of varying sizes and provided analysis of the impact of the dataset size. For the experiments, 3 data corpora with different sizes have been created. News dataset is of size 55 million tokens, while Books and Combined datasets have 108 million and 164 million tokens respectively. We observed that the accuracy score increases as the dataset size is increasing, syntactic score peaking at 83.9% accuracy on the dataset consisting of 164 million tokens. The explanation is that in larger datasets model sees words in a lot more contexts allowing it to learn more accurate vector representation for the words.

5.2 Conclusion

In this thesis, very recent language representation architectures, word embedding generation techniques, contextualized word embeddings had been discussed. Word embeddings have been generated and their effectiveness have been measured empirically by means of intrinsic and extrinsic evaluation experiments. Different evaluation parameters for language representation learning architectures have been analyzed in this thesis paper. These techniques are at the root of many breakthroughs and the establishment of new state-of-art results in almost any natural language processing tasks. We showed and thoroughly analyze word embeddings for a morphologically rich, agglutinative language, namely, Azerbaijani. Achieving high accuracy scores in syntactic evaluation experiments shows that word embeddings are effective at preserving syntactic relationships for agglutinative languages which utilizes morphemes to create complex words. Generated word embeddings have also been applied to sentiment analysis task which is used as an extrinsic evaluation

experiment to assess the embeddings. Accuracy score on sentiment analysis task is 95.3%. We hope that this paper will be useful for researchers whose research is crossing with human natural language representation approaches.

Glossary

NLP	Natural Language Processing
BLUE	Bilingual Evaluation Understudy
GloVe	Global Vectors For Word Representation
CBOW	Continuous Bag of Words
BERT	Bidirectional Encoder Representations from Transformers
ELMo	Embeddings from Language Models
LSTM	Long Short-Term Memory
TF-IDF	Term Frequency Inverse Document Frequency
TF	Term Frequency
POS	Part of Speech Tagging
NER	Named Entity Recognition
LSA	Latent Semantic Analysis
MT-DNN	Multi-Task Deep Neural Networks
DL	Deep Learning
RNN	Recurrent Neural Networks
ML	Machine Learning
IDF	Inverse Document Frequency
CFG	Context Free Grammar
GPT	Generative Pre-trained Transformer
RoBERTa	A Robustly Optimized BERT Pretraining Approach
XLNet	Generalized Autoregressive Pretraining for Language Understanding
GPU	Graphics processing unit

TPU	Tensor processing unit
CNN	Convolutional neural network
CRFs	Conditional Random Fields
BRNN	Bidirectional Recurrent Neural Networks
GAP	Global Average Pooling
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
RNNLM	Recurrent Neural Network Language Model

6. References

[1] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

[2] Qing Cui, Bin Gao, Jiang Bian, Siyu Qiu, Hanjun Dai, and Tie-Yan Liu. (2015). KNET: A general framework for learning word embedding using morphological knowledge. *ACM Transactions on Information Systems*, 34(1):4:1–4:25.

[3] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. (2012). Improving Word representations via global context and multiple word prototypes. In Proceedings of *Association for Computational Linguistics*, pages 873–882.

[4] John Duchi, Elad Hazan, and Yoram Singer. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* 12, (2/1/2011), 2121–2159.

[5] D. W. Otter, J. R. Medina and J. K. Kalita, (2020). "A Survey of the Usages of Deep Learning for Natural Language Processing," in *IEEE Transactions on Neural Networks and Learning Systems*, doi: 10.1109/TNNLS.2020.2979670.

[6] Jose Camacho-Collados & Mohammad Taher Pilehvar. (2018). From word to sense embeddings: a survey on vector representations of meaning. *J. Artif. Int. Res. 63, 1 (September 2018)*, 743–788. DOI:https://doi.org/10.1613/jair.1.11259

[7] Joseph Turian, Lev Ratinov, and Yoshua Bengio. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

[8] Landauer, T., & Dooley, S. (2002). Latent semantic analysis: theory, method and application. *In Proceedings of CSCL*, pp. 742–743. Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers, 28 (2)*, 203–208.

[9] Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, pp. 2177–2185.

[10] R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neuralnetworks. *CoRR*, abs/1412.1058, 2014.

[11] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*: Human Language Technologies, pages 912–921.

[12] Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 51–61, Berlin, Germany.
[13] Melamud, O., McClosky, D., Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1030-1040.
[14] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR, abs/1301.3781*.
[15] Mikolov, T., Le, Q. V., & Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.
[16] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. *In Advances in neural information processing systems*, pp. 3111–3119.
[17] Mikolov, T., Yih, W.-t., & Zweig, G. (2013d). Linguistic regularities in continuous space word representations. *In HLT-NAACL,* pp. 746–751.
[18] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *In Proceedings of EMNLP*, pp. 1532–1543.
[19] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL*, New Orleans, LA, USA.
[20] Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18 (11)*, 613–620.
[21] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," 2019, arXiv:1901.11504.
[22] Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In Proceedings of *Empirical Methods in Natural Language Processing.*
[23] Turney Peter D. & Pantel Patrick (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research, 37,* 141-188.
[24] Searle, J., 1980, 'Minds, Brains and Programs', Behavioral and Brain Sciences, 3: 417–57

[25] Grady, J. S., Her, M., Moreno, G., Perez, C., & Yelinek, J. (2019). Emotions in storybooks: A comparison of storybooks that represent ethnic and racial groups in the United States. Psychology of Popular Media Culture, 8(3), 207–217.
[26] Preston, John and Mark Bishop (eds.). (2002), Views into the Chinese Room: New Essays on Searle and Artificial Intelligence. Oxford/New York: *Oxford University Press*, xvi + 410.
[27] Turing, A.M. (1950), 'Computing Machinery and Intelligence', *Mind 59*, pp. 433–460.
[28] Searle, J. (1980b), 'Intrinsic Intentionality: Replies to Criticism of "Minds, Brains and Programs", Behavioral and Brain Sciences 3, pp.450–4566
[29] Searle, J. R. Is the Brain's Mind a Computer Program?, *The Scientific American*, January 1990.
[30] Searle, J. R. The Myth of the Computer, *The New York Review of Books*. April 29, 1982.
[31] Dennett, D. (1984), 'Self-made Selves', in Daniel Dennett (ed.), Elbow Room, *Cambridge, MA: The MIT Press*, pp. 74–100.
[32] Searle, J. (1993), 'The Failures of Computationalism', Think 2, pp. 68–7l.
[33] Brown, Tom & Mann, Benjamin & Ryder, Nick & Subbiah, Melanie & Kaplan, Jared & Dhariwal, Prafulla & Neelakantan, Arvind & Shyam, Pranav & Sastry, Girish & Askell, Amanda & Agarwal, Sandhini & Herbert-Voss, Ariel & Krueger, Gretchen & Henighan, Tom & Child, Rewon & Ramesh, Aditya & Ziegler, Daniel & Wu, Jeffrey & Winter, Clemens & Amodei, Dario. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems (33) (NeurIPS 2020)*, 1877-1901.
[34] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Adv. NIPS*
[35] Miguel Ballesteros, Chris Dyer, and Noah A. Smith. (2015). Improved transition-based parsing by modeling characters instead of words with LSTMs. *In EMNLP*, pp. 349–359.
[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.
[37] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.
[38] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. (2018.) QANet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

[39] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.