Simplified Natural Language Processing and Discrimination in Antonyms-Synonyms Word Structure in a Neural Network Approach to Exploit Lexico-Semantic Pattern

Maksuda Sultana[1] and Dr. Ana Romina A, Miguel[2]

[1]DIT Student, AMA University Philippines (Member PSITE)

[2]Dean SGS and CCS, AMA University Philippines (Member PSITE, ICpep)

[1]mak.sultana@gmail.com and [2]aramiguel@amaes.edu.ph

Abstract— To get high efficiency or high yielding in NLP system the vital challenge is to distinguish between antonym and synonym. By using lexico-syntactic patterns in pattern-based models, which are string-matching patterns based on lexical and syntactic structure, we exploiting to represent the distinguish between antonyms and synonyms word pairs as vector representation in Arabic word structure. It is very difficult to make automatic distinguish between antonymy-synonymy in NLP system because of they have a tendency to occur in similar contexts. I intend a 2-step novel process that exploit lexico-semantic pattern to distinguish antonymy-synonymy from syntactic parse tress. The experiment result shows the improvement of the performance over prior pattern-based method.

Keywords— Neural language processing, Semantic relation classification, Antonyms-synonyms Distinction.

I. INTRODUCTION

From the last few years, NLP gathered excellent fulfillment in human language modeling for the cease – to cease computer system. Thus this system can easily understand and recognize everything from very short meaning of any type of human language. It's a big enormous expectation that how to full use power of cease to cease of any computer system which can be made in a sense of less meaning of the language of humans. So to get better performance from the computer system and to understand the meaning of different words and their semantic similarity is pivotal for many NLP system such as machine learning algorithm, machine translation named entity recognition, question answering, document summarization, predictive typing, co-reference context and so on.

There are two basic components can be classified in NLP system that is natural language understanding and another ne is natural language Generation. By using NLP system, we can do phonological analysis, morphological analysis, Lexical Analysis, Syntactic Analysis, Pragmatic Analysis and so on. The structure of words analyzing and identifying is under lexical analysis and very common and important lexicon normalization practice are stemming and lemmatization. Any machine can understand and process any word which we call a token that is the minimal unit in this system and the first step in NLP system is tokenization. Word tokenization, stopword removal, part of speech tagging, chunking, named entity recognition is crucial part in the NLP techniques that help us to improve our communication, our goal reaching and outcomes that we receive from every interaction.

NLP or Natural language processing is a branch of Artificial Intelligence (AI) that support computers understand, interpret and manipulate human language and there are some computational model which distinguishes semantic relations by either representing semantically related words for representations as vector in the vector space model or using neural networks to classify the semantic relation. Actually our main concern is to classify antonym-synonymy by using lexico-syntactic pattern because in linguistic structures, many aspects of semantics can exploits such as lexical semantic, semantic type, semantic relation, semantic similarity and also semantic relatedness. In semantic relation our concern is to point to task of distinguishing antonym (example small/big) and synonymy (example: help/support). Different kinds of large amount of data such as raw data, parallel data, unleveled or leveled data which accessed by NLP application Actually in the organization of the lexical database the antonymy-synonymy are lexical semantic relation play most important role [2,3]. Between two words the synonymy is a semantic relation which have the same meaning and in the otherwise between two words which gives the opposite meaning we can say the semantic relation of antonymy [1].

In this research paper basically there are two parts. First part author simplified Neural Processing language and describe some important task. In the second part author exploit lexico-semantic pattern to distinguish antonymy-synonymy from syntactic parse tress by proposed a 2-spet novel process.

There are two strategies to distinguish antonymy and synonymy in NLP system. The first strategy is applying to both distributional and distribute word vector representation by consolidating lexical contrast which is distributional to antoynmy-synonymy into word vector representation. The second strategy is distinguishing antonymy from synonymy by exploits syntactic pattern between antonymous-synonymous word pair. We applying new lexico syntactic pattern structure from the syntactic parse tree of any sentence which carry new word pairs of antonymous-synonymous. Thus we can easily handle the sparsity of standard lexico syntactic patterns and this new lexico syntactic pattern of neural network easily encode these pattern as vector representation for discrimination of antonymy from synonymy. So under this novel two step proposed model, at first this pattern based neural network encodes lexico syntactic pattern as a vector representation that help the classifier to discriminate antonymy and synonymy and in addition this model takes into another account to concatenation both vector representation of the words and thus three vector representation is used to classify antonym and synonymy. Based on lexico syntactic pattern this 2-layer neural network model improves the distinguish antonymy from synonymy to prior existing model and relatedness in a low resource language.

II. SIGNIFICANCE OF THE STUDY

There are four approaches can be classified on this problem in the previous studies such as vector semantic, unsupervised measure, syntactic structure based and pattern based. Basically Neural networks (NNs) are computational models which might be inspired by using biology of the human mind or brain and that large number of neurons (in computer system we call computational nodes) by using mapping function that are trained to map the inputs to the outputs. Under the NNs there are different types of layer such as input layer, hidden layers and outputs layer and each layer contains multiple computational neurons. The main purpose of input layer to receive the input signals from the training data, the main purpose of hidden layer for computing and transforming inputs signals into representation of training data and the output layer then transforms representation of hidden layers to the particular output format. In current years, NNs have obtain dramatic achievement in solving task in NLP. So it is clear that natural language Processing is a branch of Artificial Intelligence which focus around measuring human language to convert it intelligible to machine and also combination the power of linguistics and computer to make the intelligent system for comprehension, breaking down and separating significance from text and speech. Distinguishing between antonyms-synonymy is crucial and main task to get highest performance in NLP system which focused on lexical semantic extraction. Huge contextual overlap and highly similar context occur in antonym and synonym pair, so it very hard task to distinguish them. So the proposed 2-step novel neural model not only using embedding representations of words but also co-occurrence and patterns of Arabic word structure. I achieved significant performance improvement over prior pattern based methods.

Under pattern based model the vector representations always use the lexico-semantic surface form or pattern to discriminate between the relations of word pairs and it has the ability to alleviate antonymy-synonymy interchangeable substitution. So always face a problem the sparsity of pattern and can't cover all antonymous-synonymous pairs by using typical lexico-semantic pattern. For example, in the sentence "He is wearing two different color of socks but his shoes are the same", the antonymous pair "different-same" can't possible to derive from any typical pattern and for handling this sparsity of pattern to distinguish antonyms from synonyms I proposed 2-step neural model. The antonymous work pairs co-occur with each other where we hypothesize that within a sentence in lexico-semantic patterns more often then should be anticipated of synonymous pairs. The corpus-based research on antinomy and synonymy is the hypothesis inspired by. It is very difficult to make automatic distinguish between antonymy-synonymy in NLP system because of they have a tendency to occur in similar contexts. The purpose of the study or the main objectives of my research is the improvement of computational models which distinguish the antonymy-synonymy and also measure the semantic similarity. I intend a 2-step novel process that exploit lexico-semantic pattern to distinguish antonymy-synonymy from syntactic parse tress. The experiment result shows the improvement of the performance over prior pattern-based method and also focus to evaluate a computational model with the low resource language rather than English.

III. PREVIOUS STDIES

Synonymy and antonymy are without doubt of the most well-known semantic relations between words, and can be widely described as phrases which might be 'similar' in which means (synonyms), and words which might be 'opposite' in that means (antonyms). The charming difficulty about antonymy is that even though antonymous phrases or word are said to be opposites, they are however semantically very closer, all most similar. There are two approaches are classified on this problem distinguish between antonymy from synonymy in the previous studies that is (i)embedding approach in vector representation and (ii) pattern based approach in vector representation.

A. Embedding based approach

Turney and pantel 2010 [4] describe that this approach depends on distributional hypothesis. That means the meaning of every word which is similar or opposite emerge in same context. In 2013 Mikolov – 2013a, 2013b [5, 6] explained using neural network these must be trained. In their proposed method where they were using Word2Vec technique and this algorithm graps word from big corpora. After training classifier this method can easily calculate the synonymous word and trained classifier distinguish the word in near to far words. Their proposed method do everything in vector space model with word embedding system. In 2014 Pennington and co-author [7] describe this embedding approach always depends on embedding trained vector which have various properties and these properties always take out from big scale text corpora where they are using Glove for word representation. Co-occurrence matrix are using in this model to calculate how many words are visible in context in frequently.

In 2014 to distinguish antonyms Adel and Schutz are explained co-reference chains for training their skip-gram model [9]. In 2016 Nguyen, Schulte im Walde and Vu [10] are using skip-gram model with integrated distributional lexical contrast for distinction of antonym-synonym purpose. Their presented word embedding model name is dLCH with skip-gram which can use lexical resources externally. Using this model, they can easily find the degree of similarity and it's very effective to classify antonymy-synonymy.

In 2015 Pham, Laza and Baroni are proposed a multitasking distributional lexical contrast in skip-gram model which is called multitask Lexical Contrast Model (mLCM) with supervised information from WordNet for this distinction task [11]. This model actually extended version in skip-gram model which was very effective to convert in semantic vectors to estimate contexts.

In 2015 Miwa, Ono and Saski[12] developed a model for detect antonyms in distributional information by using WE-T and WE-TD, two types of word embedding model. Sometimes its very difficult to distinguish antonymy from synonymy of some infrequent word or some rare word. To solve this problem in 2017 Bojanowski and co-author [24] represent a method to produce greater word embedding system. This model use skip-gram model with Word2Vec to create new word related vector by the summation of the n-gram vector and then every word like as an actual valued vector. In 2017 N. Mrksic, I. Vulic, D. S´eaghdha [24] represent another model named ATTRACT-REPL for word embedding which is pre-trained and can be create vector space with unified cross lingual. It's very helpful for lower resources languages and produce high quality vectors to distinguish antonymy-synonymy.

But in every proposed model faced some limitation in this embedding approach that there are some impotence or some incompetence to discriminate between various relation in lexico semantic. For example: In globe pinnacle comparable phrases for the phrase small yield a combination of synonyms.

In 2013 Scheibla and co-authors [8] represent a method which was unsupervised with embedded in a vector space model. This method can measure distributionally and can distinguish all antonyms - synonyms those are adjectival. Their expected result was adjectival antonyms-synonyms does not have any distributionally similarity. In 2014 another unsupervised distributional measure method proposed by Enrico Santus [25]. The proposed method name was APAnt to distinguish antonyms – synonyms. They observe some antonyms are similar besides in one dimension of meaning such as tall and midget both are man/women with two hand, one head and two eyes but except the size. Average idiomatic expressions for measure in this APAnt method. The hypothesis of this model is the variety of salient contexts shared by way of synonyms are drastically higher than the quantity of the ones shared through antonyms.

B. Pattern based Approach

From big scale textual content corpora this technique depend on capture the pattern which is called lexico syntactic pattern. For the distinction of synonym from antonym there are different type of approaches is exist where they are depend on pattern. In 2003 Lin and co-author [12] developed a model to extract antonyms from distributionally same words pattern. This model depend on two pattern which is antonym pattern denoted as "from X to Y and either X or Y". The main theorem of this model is if there is two words which denotes X and Y are present in one of these pattern, they are very unlikely to present in synonymous pair. In 2013 Mohammad and co-author [14] examined that there is very low coverage in the Lin's method for the anatomy pattern. In 2008 Turney and co-author [15] developed a model which is applicable for supper vector machine. This model had a feature in removal algorithm with supervised classification. This model also contain different types of pair class. In 2013 Sabina Schelte im Walde and co-author [16] proposed a pattern based approach for German words for distinction paradigmatic relations in antonymy, synonymy and hypernymy, where at first they extract a lexico-syntactic pattern between a word pair. After then they calculate the frequency of pattern vector and finally they applied the algorithm to distinguish antonyms from synonyms. In 2014 Roth and Schult [17] proposed a modle for this purpose in both German and English language and this model can indicate the same relation in antinomy-synonymy in general lexico-syntactic pattern. They are using raw corpus and its very easy expand in other languages. In 2015 Schwatrz proposed a very good model for this task which is symmetric pattern based vector representation model.

Now a day's different authors have been used neural Network approach to distinction antonymy-synonymy in syntactic structure [25, 26]. This model can identify co-occurrence contexts of word pairs in dependency form. If (y, z) the word pair then this y and z can be considered to connect the lexico syntactic path to persuade the semantic relation in their proposed model. Their research demonstrates the effectiveness of lexico-syntactic information. K. Fundal and co-authors [28] represented the model using shortest dependency path (SDP). They were using SDP in co-occurrence context between two words to recognize the relations. Some authors are using recurrent neural network (RNN) to handle the dependency parse tree between word pairs in end to end relation. In 2015 Y.Xu and co-authors [29] proposed long short term memory network to handle dependency parse relation. They achieved highest performance in their research by using deep neural network technique because from syntactic parse tree they found lexico syntactic pattern automatically and their model had the ability to automatically learning.

In the Arabic language J.Sadek, and co-author [30] presented a method which is based on Rhetorical structure theory for question-answering purpose in extracting relation from text. A. Ibrahim [31] presented method using the same RST approach but for Arabic text summarization they were using pattern based for the rhetorical purpose. M.A Hearst proposed method [32] is pattern based extraction method for hyponym relation for extraction in Arabic. But this method is processed by hand crafting by manually which is time consuming though author achieved good result. It's also very hard to compare all pattern. P. Pantel and co-author [33] represent the model to overcome this problem and using a approach named bootstrapping for extraction in semantic relation. To recognize generic pattern automatically author was using an algorithm named bootstrapping algorithm containing metonymy and hyponymy. W. Wang and co-authors [34] presented a method in verb antonyms-synonyms extraction from English newspaper corpus. Using WordNet at first analyze the pattern corpus and then compute the confidence value for each pattern. After then new antonymy-synonymy pairs are extracted.

The rest of this paper is organized as follows: Section IV present summery or overview of the work. Some important features are described in section V. In section VI present the methodology and description of the model architecture of the proposed model. After then section VII present the experimental setting and section VIII shows the result. The last section presents the conclusion.

IV. SUMMERY OF THE WORK

There are two strategies to distinguish antonymy and synonymy in NLP system. The first strategy is applying to both distributional and distribute word vector representation by consolidating lexical contrast which is distributional to antoynmy-synonymy into word vector representation. The second strategy is distinguishing antonymy from synonymy by exploits syntactic pattern between antonymous-synonymous word pair. We applying new lexico syntactic pattern structure from the syntactic parse tree of any sentence which carry new word pairs of antonymous-synonymous. Thus we can easily handle the sparsity of standard lexico syntactic patterns and this new lexico syntactic pattern of neural network easily encode these pattern as vector representation for discrimination of antonymy from synonymy. So under this novel two step proposed model, at first this pattern based neural network encodes lexico syntactic pattern as a vector representation that help the classifier to discriminate antonymy and synonymy and in addition this model takes into another account to concatenation both vector representation of the words and thus three vector representation is used to classify antonym and synonymy. Based on lexico syntactic pattern this 2-layer neural network model improves the distinguish antonymy from synonymy to prior existing model and relatedness in a low resource language.

Now a day's neural processing network achieved high performance rather than other method based on lexico syntactic co-occurrence context. This method depends on parsed corpora and can easily undergo the syntactic parser error. Can and co-author's proposed model[35] could ignore some adverb, negation which were also valuable information by using dependency but Mandar joshi's [36] proposed model were used an alternative structure to exploiting surface from word context named pair2vec model. The analysis from the previous studies in section 2 there are two types of complication we found such as:

- In neural network system sometimes without depending on syntactic parse tree its exploit the context of their pair of words. But for some datasets with high resources syntactic parsers are not enough to achieve an accuracy result than English.
- Using neural network models to exploit co-occurrence contexts of word pairs without relying on syntactic parse trees: For low-resource languages like Vietnamese, syntactic parsers can only achieve an accuracy much lower than those for English [23].

The Qur'an as a corpus and is made up of seventy-seven, 430 words. It is split into 114 chapters which consist of 6,243 verses. The Qur'anic WordNet services everybody who seeks to make bigger his knowledge of Qur'anic Arabic vocabulary and increases knowledge of the Qur'an and of Islam. In Qur'an, we discover many phrases that are conceptually synonyms however if we check out their dictionary meanings, then variations will surface such as Ahmded (SAW) : أَحمَد , Al-Muzzammel : المزَمِّل , Al-Muddasshir : المدَّثِّر , Yasin : يس , Ar-Rasul : الرَّسول all are the synonymous of Muhammed (SAW). In another war example such as Sabili : سَبِيلِ , and wazhe : وَجْهِ are the synonym of two words where the meaning is spend their wealth in that way. In table 1 shows the example of sabili and wazhe in different verse 'ayah in same chapter.

Example :1 – chapter/Para : 2,verse'Ayah : 272	
وَمَا تُنْفِقُونَ إِلَّا ابْتِغَاءَ وَجْهِ اللَّهِ	And do not spend expect seeking the countenance of ALLAH
Example:2 – chapter/Para : 2, verse'Ayah : 261	
مَثَلُ الَّذِينَ يُنْفِقُونَ أَمْوَالَهُمْ فِي سَبِيلِ اللَّهِ كَمَثَلِ حَبَّةٍ أَنْبَتَتْ سَبْعَ سَنَابِلَ	The example of those who spend their wealth in the way of ALLAH is like a seed of (of grain) which grows seven spikes

Table: 1

- because of textual context and complex morphological structure in Quranic Arabic text it is not easy to deal and find strong, accurate parts-of-speech tagging system in the Quranic Arabic text process.

By the virtue of description the corpus in Quran for all, in 2010 Dukes and Habash [37] proposed the Quranic Arabic Corpus which is an combined and authentic linguistic resources that is contain the total 77,430 words of Qur'anic Arabic and 114 documents which we call Sura. Every word in indicate with its POS together with multiple morphological feather. These feathers rely on the traditional Arabic grammar.

All patterns and clauses are very rich in Arabic grammar which serve the special function inside the sentences. In Arabic out of nine conjunctions only six of them play role in the holy Quran which have a conjunctive role. These conjunctions repeated in several times [38] in holy Quran show in the table 2 below:

The Arabic conjunctions introduced in the Holy Quran:

ثم	أو	و	بل	الفاء	أم
Then	Or	And	But	Then	Or

Table: 2

We are using Qur'anic WordNet for Arabic language distinguish between antonym-synonym and by modeling Qur'anic WordNet for better understanding of the meaning of Qur'an and for computational linguistic theory in Arabic language using new technology in NLP. To develop Qur'anic WordNet, we will make use of Arabic WordNet [39], Ontology of Qur'an and classical Arabic dictionary. Qur'anic WordNet is also useful for Arabic linguistics and Lexicography.

After then for English language we describe 2-step model using LSTM which are pattern based for distinguish antonyms from synonyms. At first we will describe the process or action of the pattern. After complete this we will present how to encode the all pattern as vector presentation by using recurrent neural network with long short term memory. And finally we describe the method to classify for antonym and synonym. Nguyen and co-author's [40] proposed model is presented for distinguish between antonomy from synonymy where they are using dependency path from the syntactic parse trees to exploits lexico semantics pattern. But in this paper 2 step model we are using a surface form rather than dependency path and measuring a semantic similarity between words are using for classify antonomy and synonymy.

V. PROGRAMMING FEATURES FOR NLP

Some methods are very important for data processing in NLP. Such as a) Tokenizing words b) stop words c) stemming words d) Part-Of-Speech tagging for each word e) chunking f) lemmatizing g) named entry recognition

A) Tokenizing words: The processes of breaking the raw data into small pieces or chunks is call tokenization and the small chunks are called tokens. So we can say after tokenization breaking words in a sentence are called tokens. For better context understanding or for developing the NLP model these tokens are very helpful. The tokenization helps in decoding which means of the text by using reading the collection of the words. Based on the language and purpose of modeling there are different kind of tokenization technique are applied in NLP system such as white space, dictionary based, rule based, moses tokenizer and so on.

B) Stop words: The process of changing statistics or raw data to something a computer can apprehend is called pre-processing. To remove useless data is most important part of the pre-processing system and theses useless data or words are mention stop words in NLP system. a lot of times people might use words that are typically used sarcastically as stop words because they don't want to continued attempting to analyze something when it may or may not be the actual like opposite meaning. So that's notion but another notion of stop words these words that we just pull out and we just don't care about them. They are fluff and we don't need them. So this should be like a, the, and for the most part these kinds of things don't really have any meaning to our text. Therefore, we can just remove them because they are filler words basically and they make our language to us make a lot of sense but as far as data analysis is concerned they are useless.

C) Stemming words: The idea of stemming is kind of it's a form of data pre-processing and it's a form of not really normalization but it's the best word if compare it to and take the root stem of the word. That means the process where any one can reducing a word to its word stem is call stemming. So for the example if we have got a writing the stem of writing would be rid basically. So we get rid of the "ing" and you have a stem of our ID. Our ID is applicable to ride, riding, ridden that kind of stuff. We need to understand why we are even doing this and the reason why is a lot of times we are going to have different variations of words based on their stems and at the end really the actual meaning of that is unchanged. So for example let's say we have two sentences such as sentence 1: they were taking a ride in the bus sentence 2: they were riding in the bus. In both sentence all are same including the tense but the difference is the ride and ridding. And we can say in this stemming process from the word stem the reducing word that affixes to suffixes and prefixes are known as a lemma.

D) Part-Of Speech tagging: Part of tagging one of the most powerful features in the NLTK module. By using this we can make labeling words in a sentence as nouns, adjectives, verbs and so on. Not only has the word had it also labeled by tense. It's very much useful for building parse tree and also extracting the relation between words. Parts Of speech tagging are essential when we want to build lemmatizers for reduce a word to its root form. Another name of parts of speech tagging is grammatical tagging because we can divide every word in a text (called corpus) as equivalent to a parts of speech.

E) Chunking: The process of pull out the phrases from unstructured textual content or any text is called chunking. It is being any one can analyze a sentence to recognize the noun groups or verbs or verb groups and so on. But their internal structure is not identifying nor is their main role in the sentence. This process work on POS tag as a input and provide chunks as output.

F) Lemmatizing: Lemmatizing is very identical as stemming operation. the main different is stemming can often nonexistent words and lemmas are treated as actual words. So lemmatizing actually considerate the analysis of morphological of the word. If we consider two words such as studies/ studies then the lemma should be 'study' in both words but in stemming the Stem should be "studi/study". So we can get perfect word from lemmatizing process which is very important in NLP.

VI. METHODOLOGY

In Arabic Quranic WordNet the following steps are included for implementation and for preprocessing Quranic text we have to complete some process such as tokenization, stop words, stemming words, Part-Of-Speech tagging for each word, chunking, and lemmatizing.

- The synonym sets are called in synsets in Arabic WordNet are generated with the aid of words which words are grouping words and part-of-speech. For example, the word (absara) and the words (nazoro) both share the same meaning 'see' but in a synset those are grouped together.
- Between different types of synsets are explained in semantic relation and following are included are included in Qur'anic WordNet are:

1.Synonymy: The words that are similar in meaning are called synonymy and synonymy are determined using synsets. Such as the words (haol), (sanahh) and (aim) all three words are same meaning 'year' and these are synonyms.

2.Antinomy: The words which provides opposite meaning are called antinomy and they are marked For example: (Al hayaat) means "life" and (al mowot) means "life" and these two words are totally opposite and indicate as antonym.

3.Glossary: Glossary is using for the store an explanation, definition and example of every sentences and gloss is a small part to store a specific sentences information for every synset. Actually gloss is stored by glossary and glossary is active after compilation. Some words that are using for same sense or share the same sense but is using in different contexts. For example: the two words (al-mator) and (al-goyat) both two words are the same meaning that is 'rain ' but they are using in the different contexts.

Word	Semantics	Example	Translation
al-maṭar المطر	torment	وَأَمْطَرْنَا عَلَيْهِمْ مَطَرًا فَانْظُرْ كَيْفَ كَانَ عَاقِبَةُ الْمُجْرِمِينَ (الأعراف: 84)	And We rained upon them a rain [of stones]. Then see how was the end of the criminals.
al-gayṯ الغيث	goodness and grace	وَهُوَ الَّذِي يُنَزِّلُ الْغَيْثَ مِنْ بَعْدِ مَا قَنَطُوا وَيَنْشُرُ رَحْمَتَهُ ۚ وَهُوَ الْوَلِيُّ الْحَمِيدُ (الشورى: 28)	And it is He who sends down the rain after they had despaired and spreads His mercy. And He is the Protector, the Praiseworthy.

Table: 3

4.Similarity: Similarity means different word but share the sense same meaning. In qur'anic WordNet it's differentiated after connecting the synsets. Such as the words خشيه (khoshiniyh) خوف (khowf), الروع ar-rowu, and الرهب (ar-rohb) all words are different words but provide the meaning 'fear' and 'fright'. Table 4.

Word	Semantic	Example	Translation
خشيَه Khoshiya	As(they) fear	إِذَا فَرِيقٌ مِنْهُمْ يَخْشَوْنَ النَّاسَ كَخَشْيَةِ اللَّهِ (النساء: 77)	At once a party of them feared men as they fear Allah or with [even] greater fear
خوف Khaowf	fear	فَمَا آمَنَ لِمُوسَىٰ إِلَّا ذُرِّيَّةٌ مِنْ قَوْمِهِ عَلَىٰ خَوْفٍ مِنْ فِرْعَوْنَ وَمَلَئِهِمْ أَنْ يَفْتِنَهُمْ (يونس: 83)	But no one believed Moses, except[some] youths among his people, for fear of Pharaoh persecute them

Table 2a : Example of similar meaning { خشيَه Khoshiya, خوف Khaowf }

Word	Semantic	Example	Translation
الروع l-row u	fright	فَلَمَّا ذَهَبَ عَنْ إِبْرَاهِيمَ الرَّوْعُ وَجَاءَتْهُ الْبُشْرَىٰ يُجَادِلُنَا فِي قَوْمِ لُوطٍ (هود:74)	And When the fright had left Abraham and the good tidings had reached him, he began to argue with Us concerning the people pf Lot
الرهب l-Rohb	fear	وَاضْمُمْ إِلَيْكَ جَنَاحَكَ مِنَ الرَّهْبِ (القصص:32)	And draw in your arm close to you [as prevention] from fear

Table 2b : Example of similar meaning { الروع l-row u , الرهب l-Rohb }

Table: 4

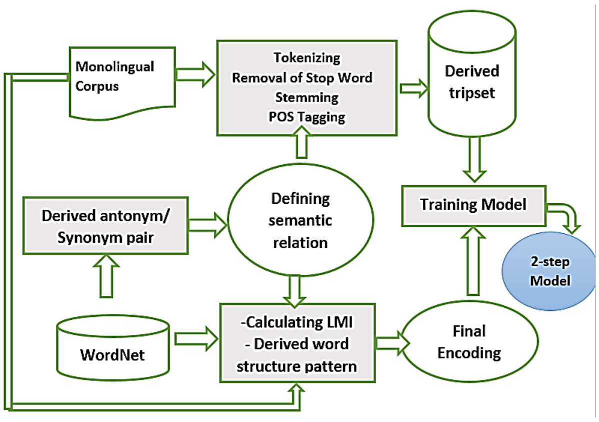

Figure 1: Methodology of the proposed model

Under the 2-step novel model I present a framework which can exploit to distinguish antonyms from synonyms. First step approach a bidirectional Long Short-Term Memory Architecture (LSTM) to classify antonyms and synonyms to encode co-occurrence context or patter as a vector representation which vector will convert a logistic regression. The second step a hard code will combined vector representation as a word structure patterns which concatenate the patterns structure and the vector of the words.

There is two variable assign in pattern based technology Y and Z. Two words of an antonym or synonym word pair is denoted by Y and Z. The pattern is stratifying in a syntactic parse tree between Y and Z as a simple path. Every node like as a simple path which combined the lexical and syntactic information.

All pattern feature nourished into LSTM to encode the patterns as a vector representation. After that this patterns of vector representation can be used as a classifier to discarnate between antonyms-antonyms (section 3.3). In this methodology we extracted a set of pair antonym- synonym from Arabic WordNet [41] and using Arabic lexicon (AL : http://compling.hss.ntu.edu.sg/omw/). After getting the tripset (m, n and contextual word) is extracting from this set where m denoted synonyms, n is denoted antonyms and other words in a sentence is denoted as a contextual word.

A) Long Short-Term Memory (LSTM) Architecture:

A Recurrent Neural Network (RNN) can recognize data's sequential characteristics and knowing the next latter in a word or the next word in a sentence which use the pattern to predicated on the data that comes before it. So RNN is very useful for modeling sequential data to covert by a vector presentation by nature. Here we use LSTM in our system because to overcome the drawback of standard RNN is the vanishing gradient problem. In 1997 S. Hochreiter and co-author [42] first proposed LSTM architecture for solving recurrent Neural Network problems. In 2015 Y. Xu, L. Mou, and co-authors [43] proposed model is more impressive for the programmer because this architecture apply POS tagging, pattern of words, dependency intensity and WordNet hypernyms together with path. And in 2017 K.A. Nguyen, S.S. im Walde and co- author [44] proposed a Recurrent Neural Network with bidirectional long short term memory architecture(biLSTM) which is very impressive to encode the lexico syntactic patterns as a vector representation.

There are four components in the LSTM based recurrent neural network such as input sequence it, a forget sequence ft, an output sequence ot, and a memory cell m_t where t stated as time. The three adaptive sequences, ft, and ot always rely on the earlier state $pt-1$ and the current input c_t. An extracted feature vector et is also computed as the candidate memory cell. Each time step t is formulated in the following ways:

$it = (W_i.c_t + Ui \cdot p_{t-1} + bi)$

$ft = (W_f \cdot c_t + Uf.p_{t-1} + bf)$……………(i)

$ot = (Wo.\ c_t + Uo \cdot p_{t-1} + bo)$

$gt = tanh\ (Wg \cdot ct + Ug \cdot pt-1 + bg)$…….(ii)

The recurrent neural network (RNN) always sustain with hidden state vector z, where z always changes with inputs data. The current memory cell mt is a composite of the earlier cell content $mt-1$ and the candidate content gt,

$m_t = i_t \otimes g_t + f_t \otimes m_{t-1}$…………(iii)

The output of LSTM units is the recurrent network's hidden state, which is computed by Eq. (iv) as follows.

$z_t = ot \otimes tanh(m_t)$……….(iv)

In the above equations, σ denotes a sigmoid function; $\otimes$ denotes element-wise multiplication.

B) Model Architecture

In this 2-step architecture we are using recurrent neural network including Long-Short Term Memory (LSTM) structure to complete the encoding system in the context of every word pairs. And there are a tripset denoted by m and n for antonomy and synonymy respectively and contextual words denoted by cw where every word is presented or calculated as a vector by the concatenation of 1st text and POS embedding. So every contextual word $cw_{1:z}$ are nourish from the bidirectional Long Short term Memory (biLSTM) system. After then the contextual vector which is denoted by v vector is defined as:

$\vec{V}\text{biLSTM} = [\vec{V}\text{leLSTM}(wc_{1:z}), \vec{V}\text{rhLSTM}(wc_{z:1})]$…...(v)

In the above equation leLATM indicate left-right word embedding and rhLSTM indicate right-left word embedding of any contextual words. Now we need to apply non-linear function to find the Multi-Layer Perception and after multiplying the left context and right context we can easily find that Multilayer perception which is given bellow:

$\text{MPLE}(\vec{V}\text{biLSTM}) = \text{L2}(\text{ReLUA}(\text{L1}(\vec{V}\text{biLSTM})))$…… (vi)

Here MLPU indicate Multi-Layer Perception, ReLUA stands for Rectified Linear Unit Activation Function. And $(z) = W_{iz+bi}$ represent the linear operation function. Actually the output of the Bidirectional Long Short Term Memory is given by the MLPU vector. Now we can calculate the Contextual words vector:

$\vec{V}_{cw} = \text{MLPU}(\vec{V}\text{biLSTM})$……………(vii)

After finding the contextual vector we can easily calculate the first encoded vector which will help us to finalize the final encoding vector. The concatenation of the $\vec{V}_{cw}$, $\vec{V}_m$ and $\vec{V}_n$ we can easily calculate the first encoded vector which is donated V_{first}. This first encoded vector features which is learned through by training the neural network.

$\vec{V}_{first} = [\vec{V}_{cw} . \vec{V}_m . \vec{V}_n]$……………………………(viii)

After that we need to calculate the final encoded function. For calculating this we need a vector which we call vector construction function (VECF) that is a x-dimensions vector. For generating this we need a sigmoid function (SignFunc) by using dot product of two embedding vectors and encoded value of word-structure pattern (WSP):

$\vec{V}_{final} = \text{VECF}(\text{SignFunc. WSP})$…………………..(ix)

At last we can calculate the logistic regression tripset vector to classify the antonyms-synonyms from the word and the concatenation of the two vector $\vec{V}_{first}$ and $\vec{V}_{final}$ we can find the tripset vector:

$\vec{V}_{tripset} = [\vec{V}_{first} . \vec{V}_{final}]$…………………………(x)

C) Final encoding:

1: Word structure: Different types of pattern and clauses are very rich in Arabic grammar which solve the various purposes in any sentences. For encoding lexico-syntactic patterns we uses a conjunctive pattern. The structure of conjunctive or compound synonyms-antonyms are very exciting to analyze. In Arabic there are nine conjunctives. Among this two combined terms must have a type of association between each other. In the Holy Quran only six conjunctive words have active role and are separated in several times [45] which in shows in Table1. The semantic relationship between syllables is one of the strong indication of the antonymy-synonymy is proven by the linguistic analyzes of conjunctive antonymy-synonymy.

***2. Measuring Word similarity*:** From the word relatedness sometime times we need to distinguish word similarity. Similar word which is same as near synonyms. For example, car and bicycle both are similar. But in related words which can be related anyways. For example, car and gasoline both are related words, not the similar words. The score of similarity word pair can be very effective to exploit this score to calculate the word pair's semantic relation. If we denote word pair as two embedded vectors, then the distance of this vectors can be classified to calculate the word similarity. We are using dot product of the embedded vectors to calculate the similarity of word pair and the using the sigmoid function to formalize the dot product score.

VII. EXPERIMENT

1: Dataset: For the experiment at fist should trained the model and for this training large amount of data needed by the neural network system. So for training we are using large scale dataset which is previously used. Some researcher such as in 2015 named Schwarts and co-authors and in 2014 named Ruth and co-authors and also in 2017-17 another researchers name Walde and co-authors used this dataset. From various sources we are trying to accumulate the WordNet and WordNik1 for different surround. In this research also using multilingual WordNet for testing some purpose.(OMWEdit_The_Integrated_Open_Multilingual_Wordnet). For training we select three different categories such as noun, adjectives and verbs of antonym-synonym word pair. From the dataset at first we have to persuade the pattern in order to word pair, we recognize the specific sentence from the corpus. This corpus must be carry on the word pair. After then we have to filter all word pattern because we need the word pair. For this experiment we need to train the data, validate the data and test the data. In table 3 shows the three-word category of antonym-synonym word pair's size. It shows not only train data set, also shows the train and developed data. The trained data used to train the model in the first stage and then the developed data is using for the parameter in our model. But the performance is measures in the testing data

	Trained	Developed	Testing	Total
Verb	2625	192	800	3617
Noun	2945	214	1128	4287
Adjective	4863	298	1878	7039

TABLE: 5

In the final novel 2 dataset antonym-synonym word pair are extracted from the parts-of-speech (POS) group such as adjective, noun and verb. Total number of Antonym-synonyms word pair are strictly selected those have the same POS. In Arabic Language Synonymy word is more than the English language synonymy word.

***2. Experimental Setting**:* For experimental setting we are using the baseline model to express to idiomatic expressions of the model and performance only for the embedding vectors witch is pre-trained and can easily distinguishing the pair of antonym-synonym word. For getting the of vector difference of antonym-synonym word pair form the direct baseline model we are using the k-means clustering. From this clustering method we get the k-pivot vector which the representative of the active antonym-synonym word pair. Besides, using final encoded features is likewise evaluated against baselines. For parsing the random vector of corpus we depend on [2]spaCy and using 300-dimension random vector. For lemma embedding we depend on dLCE [46] and dimensionality is 100d. This is used for customized embedding for antonymy-synonymy distinguish task. Pre-trained 300d Glove [47] embedding are totally unsupervised embedding vectors. All of this models are general purpose model and specialized purpose models fastText [48] for logistic regression method for relation classification propose. For word embedding in POS tagging the we are using 5D. In our model we also use fastText embedded vector for representing a concatenating vector which dimension is 300d. Extracting triplet form our trained embedding model monolingual and multilingual corpus both are used.

In 2014 Rooth and co-authors were using the discourse markers for computing the performance which is very useful for design vector space model in lexico-symantic pattern. For comparison with our data we are using the same score from published papers. We also make a comparison with a very popular model Nguyen, Schulte im Walde model named AntSynNET model in 2017 [49]. Their proposed model had two types of architecture that AntSynNET and combination of AntSynNET. The combined AntSynNET model are differentiate between two models that AntSynNET_Glove and AntSynNET_dLCE. We comprise or result with their score in the published papers.

VIII. EXPERIMENTAL RESULT

The experimental final result in our model are shown in table-5. With the same setting in the baseline model our result are compared between two different pre-trained embedding that is Glove and dLCE for our model. For verifying the performance of our model for antonym-synonym distinguish task to another model at first we selected the Arabic language but Arabic language is very rich then other language and very rich in English language also.

The synonym of every word is five to six times double then English and its has very complex morphological structure. So we choose a low resource language like Urdu and our model give very good performance in low resources. We also test it by using open multigual WordNet and get good performance. For Urdu language we created a dataset manually using linguistic resource. It contains 850 instances with an identical share of antonyms-synonyms word pair. We divided the data set into trained 65%, test 30% and 5% developed set.

Table_6 shows the performance compared with the sp_baseline model and its shows the vast improvement and very potential for working with other language furnished with the availability of pre-trained embedding word pair with nominal dataset.

Embedding	Model	Adjective			Verb			Noun		
		P	R	F1	P	R	F1	P	R	F1
	Markers	0.716	0.716	0.716	0.298	0.298	0.298	0.822	0.830	0.830
	Symmetric Pattern	.0731	0.705	0.717	0.261	0.608	0.583	0.628	0.392	0.432
	AntSysNET	0.754	0.787	0.775	0.731	0.822	0.783	0.803	0.820	0.826
Glove	Baseline	0.701	0.618	0.655	0.635	0.631	0.633	0.981	0.045	0.663
	AntSynNet_Glove	0.751	0.789	0.774	0.716	0.825	0.765	0.804	0.817	0.812
	Novel_2 Model	**0.864**	**0.916**	**0.881**	**0.881**	**0.941**	**0.895**	**0.832**	**0.868**	**0.845**
dLCE	Baseline	0.896	0.896	0.898	0.858	0.824	0.835	0.870	0.857	0.871
	AntSynNet_dLCE	0.762	0.801	0.754	0.744	0.813	0.771	0.813	0.876	0.844
	Novel_2 Model	**0.914**	**0.933**	**0.924**	**0.896**	**0.848**	**0.920**	**0.916**	**0.914**	**0.922**

Table 6: Performance comparison some baseline model in antinomy-synonymy distinction

Model	P	R	F1
SP_model	0.622	0.633	0.578
Novle_2	0.887	0.876	0.891

Table 7: Final ratio in antinomy-synonymy distinction

For primary we are using-Score (F1) which is harmonic meaning of precision (P) and recall (R). Comparing the performance with Glove model we observed that for nouns the performance is relatively lower then as verbs and adjectives and it occurs specially in noun because of the effect of polysemy. The polysemy words is unable to handle by unsupervised word embedding vector. This score is almost align the same as Md. Asif Ali and co-authors published papers in 2019.

They also find the adjectives and verbs word pair are relatively high contextual clues then the noun word pairs same as our model. Comparing with the model dLCE the previous state- of –art the F1 score is higher than all three word classes.[50] It improves the F1 scores by 19%, 16% and 7% respectively. F1 score is also improved from baseline scores. So our novel_2 model exploits soft encoded features and hard encoded features simultaneously accomplished higher performance.

IX. Conclusion

The proposed model introduces a pattern based deep neural network that distinguish antonyms from synonyms. We hypothesized this model can have utilized lexico syntactic pattern from co-occurrence contexts of word pair in a corpus in the sentence and word structure pattern are very important to exploited as recognize the word pattern relation. Our proposed model outperformed baseline method of recall score and F1 score also. In the future we will extend to apply the new framework to other lexico semantic relation in other language ontologies by using open multilingual WordNet.

REFERENCES

[1] J. Lyons, "Semantics", Cambridge University Press, volume 1, 1977.

[2] W.G. Charles and G.A. Miller, "Contexts of antonymous adjectives", Applied Psycholinguistics, pp.357-375, 1989.

[3] C. Fellbaum, "Co-Occurrence and Antonymy", International Journal of Lexicography, pp.281-303, 1995.

[4] Turney, P. D., and Pantel, P. 2010. From frequency to meaning: Vector space models of semantics. J. Artif. Intell. Res. 37:141–188.

[5] Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. CoRR abs/1301.3781.

[6] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In NIPS, 3111–3119.

[7] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In EMNLP, 1532– 1543. ACL.

[8] Scheible, S.; Schulte im Walde, S.; and Springorum, S. 2013. Uncovering distributional differences between synonyms and antonyms in a word space model. In IJCNLP, 489–497. Asian Federation of Natural Language Processing / ACL.

[9] Adel, H., and Sch¨utze, H. 2014. Using mined coreference chains as a resource for a semantic task. In EMNLP, 1447– 1452. ACL.

[10] Nguyen, K. A.; Schulte im Walde, S.; and Vu, N. T. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In ACL (2). The Association for Computer Linguistics.

[11] Pham, N. T.; Lazaridou, A.; and Baroni, M. 2015. A multitask objective to inject lexical contrast into distributional semantics. In ACL (2), 21–26. The Association for Computer Linguistics.

[12] Ono, M.; Miwa, M.; and Sasaki, Y. 2015. Word embedding based antonym detection using thesauri and distributional information. In HLT-NAACL,

[13] D. Lin, S. Zhao, L. Qin, and M. Zhou, "Identifying synonyms among distributionally similar words", In: Proc. of IJCAI, pp.1492-1493, 2003.

[14] S. Mohammad, B.J Dorr, G. Hirst, and P.D. Turney, "Computing lexical contrast", Computational Linguistics, Vol. 39, No.3, pp.555-590, 2013.

[15] P.D. Turney, "A uniform approach to analogies, synonyms, antonyms, and associations", In: Proc. of COLING, pp.905-912, 2008.

[16] S. Scheible, S.S. im Walde, and S. Springorum, "Uncovering distributional differences between

synonyms and antonyms in a word space model", In: Proc. of IJCNLP, pp.489-497, 2013.

[17] [Roth and Schulte im Walde2014] Michael Roth and Sabine Schulte im Walde. 2014. Combining word patterns and discourse markers for paradigmatic relation classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), pages 524–530, Baltimore, MD.

[18] Scheible et al.2013] Silke Scheible, Sabine Schulte im Walde, and Sylvia Springorum. 2013. Uncovering distributional differences between synonyms and antonyms in a word space model. In Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP), pages 489–497, Nagoya, Japan.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", In: Proc. of ICLR, 2013.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", In: Proc. of ICLR, 2013.

[21] J. Pennington, R. Socher, and C.D. Manning, "Glove: Global vectors for word representation", In: Proc. of EMNLP, Vol. 14, pp.1532-1543, 2014.

[22] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information", In: Proc. of TACL 5, pp.135-146, 2017.

[23] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information", In: Proc. of TACL 5, pp.135-146, 2017.

[24] N. Mrksic, I. Vulic, D. S´eaghdha, I. Leviant, R. Reichart, M. Gasic, A. Korhonen, and S.J. Young, "Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints", Transactions of the ACL, 2017.

[25] E. Santus, Q. Lu, A. Lenci, and C.R. Huang, "Unsupervised antonym-synonym discrimination in vector space", In: Proc. of CLiC-it, 2014.

[26] M. Roth and S.S. im Walde, "Combining word patterns and discourse markers for paradigmatic relation classification", In: Proc. of NAACL, pp. 524-530, 2014.

[27] V.T. Bui, P.T, Nguyen, and M.T. Nguyen, "Enhancing performance of lexical entailment recognition for Vietnamese based on exploiting lexical structure features", In: Proc. of KSE, pp.341-346, 2018.

[28] K. Fundel, R. K¨uffner, and R. Zimmer, "Relex - relation extraction using dependency parse trees", Bioinformatics, Vol. 23, No. 3, pp.365-371, 2007.

[29] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths", In: Proc. of EMNLP, pp.1785-1794, 2015.

[30] J. Sadek, F. Chakkour, and F. Meziane, "Arabic Rhetorical Relations Extraction for Answering 'Why' and 'How to' Questions," in Proceedings of the 17th International Conference on Applications of Natural Language Processing and Informati Systems, Berlin, Heidelberg, 2012, pp. 385– 390.

[31] A. Ibrahim and T. Elghazaly, "Arabic text summarization using Rhetorical Structure Theory," in 2012 8th International Conference on Informatics and Systems (INFOS), 2012, p. NLP–34–NLP–38.

[32] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in Proceedings of the 14th conference on Computational linguistics - Volume 2, Stroudsburg, PA, USA, 1992, pp. 539–545.

[33] P. Pantel and M. Pennacchiotti, "Espresso: leveraging generic patterns for automatically harvesting semantic relations," in Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 113–120.

[34] W. Wang, C. Thomas, A. Sheth, and V.Chan, "Pattern-based synonym and antonym extraction," in Proceedings of the 48th Annual Southeast Regional Conference, New York, NY, USA, 2010, pp. 64:1–64:4.

[35] D.C. Can, H.Q. Le, Q.T. Ha, and N. Collier, "A richer-but-smarter shortest dependency path with attentive augmentation for relation extraction", In: Proc. of NAACL-HLT(1), pp.2902-2912, 2019.

[36] M. Joshi, E. Choi, O. Levy, D.S. Weld, and L. Zettlemoyer, "pair2vec: Compositional word-pair embeddings for cross-sentence inference", In: Proc. of NAACL, pp.3597- 3608, 2019.

[37] Dukes, K., Habash, N., 2010. Morphological annotation of Quranic Arabic. In: The 7th International Conference on Language Resources and Evaluation (LREC), Valletta, Malta, pp. 2530–2536.

[38] Adhima, M., 1972. Derasat li Osloob Al-Quraan Al-Karim. In: Arabic. Cairo, Egypt: Dar Al-Hadith.

[39] Elkateb, S., Black, W., Rodríguez, H., Alkhalifa, M., Vossen, P., Pease, A., & Fellbaum, C. (2006). Building a wordnet for Arabic. In Proceedings of The fifth international conference on Language Resources and Evaluation (LREC 2006).

[40] K.A. Nguyen, S.S. im Walde, and N.T. Vu, "Distinguishing antonyms and synonyms in a

pattern-based neural network", In: Proc. of EACL, pp.76-85, 2017.

[41] OMWEdit - The Integrated Open Multilingual WordNet Editing System

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural computation, pp.1735-1780, 1997.

[43] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths", In: Proc. of EMNLP, pp.1785-1794, 2015.

[44] K.A. Nguyen, S.S. im Walde, and N.T. Vu, "Distinguishing antonyms and synonyms in a pattern-based neural network", In: Proc. of EACL, pp.76-85, 2017.

[45] Adhima, M., 1972. Derasat li Osloob Al-Quraan Al-Karim. In: Arabic. Cairo, Egypt: Dar Al-Hadith.

[46] K.A. Nguyen, S.S. im Walde, and N.T. Vu, "Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction", In: Proc. of NAACL, p. 454-459, 2016.

[47] J. Pennington, R. Socher, and C.D. Manning, "Glove: Global vectors for word representation", In: Proc. of EMNLP, Vol. 14, pp.1532-1543, 2014.

[48] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information", In: Proc. of TACL 5,

[49] Nguyen, K. A.; Schulte im Walde, S.; and Vu, N. T. 2016. Distinguishing Antonyms and Synonyms in a pattern based Neural Network. In ACL (2). The Association for Computer Linguistics.

[50] Muhammad Asif Ali,1 Yifang Sun,1 Xiaoling Zhou,1 Wei Wang,1,2 Xiang Zhao3-2019 Antonym-Synonym Classification Based on New Sub-Space Embeddings The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)