# Deep Learning Models for Paraphrases Identification

*Master Thesis*


By

SADDAM ABDULWAHAB

Supervised by:

DR. ANTONIO MORENO
MOHAMMED JABREEL



UNIVERSITAT ROVIRA i VIRGILI

Department of Computer Engineering and Mathematics
School of Engineering
UNIVERSITAT ROVIRA I VIRGILI


A dissertation submitted to the UNIVERSITAT ROVIRA I VIRGILI in accordance with the requirements of the degree of MASTER in COMPUTER SECURITY AND ARTIFICIAL INTELLIGENCE.

SEPTEMBER 2017

# Abstract

Paraphrase identification is the task of identifying automatically whether a pair of sentences carries the same meaning. In this dissertation, we propose a deep learning system for paraphrase identification of tweets. The proposed system integrates the state-of-the-art features and Gated Recurrent Units to extract high level features of two tweets and identify whether they are identical. The effectiveness of the proposed system has been evaluated by using it in the supervised task of paraphrase in Twitter that presented in SemEval 2015, obtaining results which show its superiority over the state-of-the-art systems.

## ACKNOWLEDGEMENTS

First and foremost, I offer my sincerest gratitude to my supervisors, Dr. Antonio Moreno and Mohammed Jabreel, they have supported me throughout my thesis with their patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Master degree to their encouragement and effort and without them this thesis, too, would not have been completed or written. One simply could not wish for a best or friendlies supervisors.

I am grateful for the funding sources that allowed me to pursue my master degree: UNIVERSITAT ROVIRA I VIRGILI.

Finally, I must express my very profound gratitude to my parents, my brothers and my wife for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

# TABLE OF CONTENTS

# List of Tables

# List of Figures

**Figure** **Page**

CHAPTER 1

# INTRODUCTION

Paraphrases are the alternative expressions of the same (or similar) meaning. For example, "forget" is a paraphrase of "fail to remember". The criteria of semantically equivalence (i.e the same or almost the same meaning) are difficult to define exactly and can vary from task to task. Paraphrase Identification (PI) is the task of identifying automatically whether a pair of sentences carries the same meaning. It is normally a binary classification problem. Likewise, Semantic Similarity determination is another Natural Language Processing (NLP) task in which the system needs to examine the degree of semantic similarity (in a predefined semantic scale) of a given pair of texts, varying in different levels such as word, phrase, sentence, or paragraph.

Identifying paraphrases and their degree of semantic similarity has been proved to be useful for numerous NLP applications [43, 51]. For example, it can be used as feature to improve many other NLP tasks, e.g. Information Retrieval, Machine Translation Evaluation, Text Summarization, Question Answering, and others. Besides this, analysing social media data like tweets is a field of growing interest for different purposes. The study of these typical NLP tasks on Twitter data can be very interesting as social media data carries many surprises and unpredictable information.

Most of the current systems of PI use supervised Machine Learning approaches based on a basic set of features, including bag of words (BoW), part of speech tags (POS), clusters mapping, machine translation metrics and some text or POS overlap features. The most common techniques employed in supervised approaches are Support Vector Machines (SVMs), K-Nearest Neighbour, Naive Bayesian, Maximum Entropy classifiers and Neural Networks [61, 63].

For instance, The work presented in [12] used a supervised learning approach using SVM to learn a classifier based on simple lexical and character n-gram overlap features. They showed that overlap of character bigrams was more informative than that of character unigrams. Their

system is called ASOBEK and it was ranked in the first position in SemEval 2015.

The authors of [63] trained three different classifiers (Random Forest, SVM and Gradient Boost) with a set of common features. They categorized their features into five groups: (1) string-based, which measures the sequence similarities of original strings with others, e.g., n-gram Overlap, cosine similarity; (2) corpus-based, which measures word or sentence similarities using word distributional vectors learned from large corpora using distributional models, like Latent Semantic Analysis; (3) knowledge-based, which estimates similarities with the aid of external resources, such as WordNet; (4) syntactic-based, which utilizes syntax information to measure similarities; (5) other features such as using Named Entity similarity. Their system, called ECNU, was ranked in the third position in SemEval 2015.

The main drawback of those systems is that they rely heavily on a set of hand-crafted features, whose definition is very time consuming. Recently, deep learning models such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks have been utilized to extract automatically high-level features in many tasks such as [17], text classification [27], image classification [20], etc.

One interesting possibility to overcome this drawback is the work presented in [62]. They proposed a system called MITRE, in which a recurrent neural network was used to model the semantic similarity between sentences using the sequence of symmetric word alignments that maximizes the cosine similarity between word embeddings. Sets of features from local similarity of characters, random projection, matching word sequences, pooling of word embeddings and alignment quality metrics were included. The resulting ensemble uses both semantic and string matching at many levels of granularity.

Following this approach of using deep learning models, in this thesis we propose a deep learning system for PI of tweets. The proposed system integrates the state-of-the-art features of PI and Gated Recurrent Units (GRUs) to extract high level features of two tweets and identify whether they are identical.

## 1.1 Objectives

The main goal of this work is to develop a deep learning framework to identify whether a pair of tweets is identical. In order to achieve this main goal, the work will focus on the following specific goals:

1. To study the state-of-the-art on paraphrase analysis in Twitter.

2. To develop a new model to determine if two given sentences have the same (or a very similar) meaning.

3. To design a new method to determine a numerical score between 0 (no relation) and 1 (semantic equivalence) to indicate the semantic similarity between two sentences.

4. To evaluate the resulting system by testing it on some publicly available datasets.

The contributions of this work are the following:

- We have developed a novel deep learning model to identify whether a pair of tweets is identical.
- The effectiveness of the proposed model has been evaluated by using it in the supervised task of paraphrase detection in Twitter that was presented in SemEval 2015 (PIT-2015).
- The obtained results show the superiority of our model over the state-of-the-art models.

The results of these works have been presented in the following conference:

- Mohammed Jabreel, **Saddam Abdulwahab** and Antonio Moreno: Deep Learning Model for Paraphrases Identification, in the *21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, Marseille, France, 6-8 September 2017.

## 1.2 Document Structure

The rest of this document is organized as follows:

- Chapter 2: presents a background overview of the basic concepts, techniques, and tools used to support this work. It contains a brief discussion about Twitter, the knowledge structure WordNet and deep learning.
- Chapter 3: presents the state of art.
- Chapter 4: explains the methodology followed in this work.
- Chapter 5: includes the experimental results and their analysis.
- Chapter 6: presents a list of conclusions of this work and comments some lines of future work.

# CHAPTER 2

# BACKGROUND

In this chapter, the basic concepts and tools used to support this work are presented. The first section introduces Twitter, which was considered as the source of the dataset that used in this work. Later, we describe the knowledge structures used in this work, in particular, WordNet, the most used English lexical database. In the last sections we describe the following concepts: Deep Learning and its techniques such as Recurrent Neural Networks, and Vector Representations of Words.

## 2.1 Twitter

Twitter is social networking service that enables users to send and follow up messages called "tweets" and used to connect people with the same interests and share information in real time. And can be used to connect with your friends and other people. Also, get in the moment updates on the things that interest you. This process of connecting people who are complete strangers can be done with the use of hashtags.


Figure 2.1: A sample of a twee.

Hashtags, which are denoted with the “#” prefix, are added to Tweets so members of the community can share in the conversation. Also, it used when popular television shows or award

shows are on, or when significant events are unfolding or in elections, tourism and marketing in Business other companies including business to consumer can spread its content or product information through Twitter in the same way and used as in educational tool. Each tweet is a string of up to 140 characters (Figure 2.1) that may essentially include text, links (marked in black rectangle), user mentions (marked in red rectangle), symbols emoticons (marked in yellow rectangle) and hashtags (strings preceded by the # symbol with which users tag their messages; marked in green rectangle).

Twitterers (i.e., users of Twitter) may publish tweets to share news and opinions with the rest of the community. An important feature of Twitter that makes it different from other social networks is the fact that users do not need to give permission to the people that want to receive their messages and users can tweet about their experiences as soon as they happen making twitter a very powerful social tool such as facilitate authentic conversation with students and connect students with real-world problems. In fact, Twitter employs a social networking model called following, in which each twitterer can follow any other user without seeking any permission and, in consequence, he may also be followed by others without granting permission first. This is useful for users who want to receive tweets from users who they are following (i.e., followees) and to share their tweets with those that they are followed by (i.e. followers). Concerning the communication model, tweets, replys and retweets are the core of Twitter. Tweets are the messages published by twitterers. Any twitterer can reply to a tweet adding extra information or giving his impression about it creating a natural conversation among users. Finally, if a twitterer wants to only share with his followers a tweet that he has read he may retweet it and, automatically, it will be spread among his followers.

Twitter is a real-time environment, which means that tweets contain the most up-to-date and inclusive stream of information and commentary on current events, people's opinions, business trends, etc. In general, tweets are usually ungrammatical, and they contain many slang expressions, acronyms, abbreviations and symbols. These features motivate the need for systems that can identify the paraphrases in Twitter. All of these factors define an appealing research area of study for knowledge discovery and data mining.

## 2.2 Knowledge Repositories

Knowledge Repositories are online databases that systematically capture, organize, and categorize knowledge-based information. They are most often private databases that manage enterprise and proprietary information, but public repositories also exist to manage public domain intelligence. This section explains briefly one of the most popular knowledge repositories, WordNet, and explains how to use it to obtain the synonyms, hypernyms and hyponyms of a word.

### 2.2.1 WordNet

WordNet is a lexical database or semantic electronic repository for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus. In this section, an overview of its characteristics, structure and potential usefulness for our purposes is described.

WordNet is the most commonly used online lexical and semantic repository for the English language. WordNet was created in the Cognitive Science Laboratory of Princeton University. Many authors have contributed to it or used it to perform many knowledge acquisition tasks.

Concretely, it offers a lexicon, a thesaurus and semantic linkage between the majorities of English terms. It seeks to classify words into categories and to interrelate the meanings of those words. It is organized in synonym sets (synsets): a group of data elements that are considered semantically equivalent for the purposes of information retrieval. According to WordNet, a synset or synonym set is defined as a set of one or more synonyms that are interchangeable in some context without changing the truth-value of the proposition in which they are embedded. Each word in English may have many different senses in which it may be interpreted: each of these distinct senses points to a different synset. Every word in WordNet has a pointer to at least one synset.

Each synset, in turn, must point to at least one word. Thus, we have a many-to- many mapping between English words and synsets at the lowest level of WordNet. It is useful to think of synsets as nodes in a graph. At the next level, we have lexical and semantic pointers. A semantic pointer is simply a directed edge in the graph whose nodes are synsets. The pointer has one end we call a source and the other end we call a destination. All synsets are connected to other synsets by means of semantic relations. These relations, which are not shared by all lexical categories, are shown in Table 2.1:

Table 2.1: Wordnet's semantic relations.

| Nouns | |
|---|---|
| Hypernyms | Y is a hypernym of X if every X is a (kind of) Y |
| Hyponyms | Y is a hyponym of X if every Y is a,(kind of) X |
| Coordinate terms | Y is a coordinate term of X if X and Y,share a hypernym |
| Meronym | Y is a meronym of X if Y is a part of,X |
| Verbs | |
| Hypernym | the verb Y is a hypernym of the verb X,if the activity X is a (kind of) Y |
| Troponym | the verb Y is a<br>troponym of the verb X if the activity Y is doing X in some manner |
| Entailment | the verb Y is entailed by X if by,doing X you must be doing Y |
| Coordinate Terms | those verbs sharing a common hypernym |

Finally, each synset contains a description of its meaning, expressed in natural language as

a gloss. Example sentences of typical usage of that synset are also given. All this information summarises the meaning of a specific concept and models the knowledge available for a particular domain. Table 2.2 depicts the WordNet 3.0 database statistics (number of words, synsets and senses). In this work, WordNet will be particularly useful to extract semantic features to represent pair of tweets.

Table 2.2: WordNet 3.0 database statistics

| POS | Unique Strings | Synsets | Total Word-Sense Pairs |
|---|---|---|---|
| Noun | 117,798 | 82,115 | 146,312 |
| Verb | 11,529 | 13,767 | 25,047 |
| Adjective | 21,479 | 18,156 | 30,002 |
| Adverb | 4481 | 3621 | 5580 |
| *Total* | 155,287 | 117,659 | 206,941 |

## 2.3 Deep Learning

Deep learning refers to the branch of machine learning techniques, where many layers of information processing stages in hierarchical architectures are used for pattern classification and for feature or representation learning [3]. It lies in the intersections of several research areas, including neural networks, graphical modeling, optimization, pattern recognition, etc. [3]. Although machine learning based models can extract patterns from data, there is one main limitation is that they highly dependent on hand-crafted features which is time-consuming. To avoid this drawback, representation learning, particularly deep learning has shown great promise.

Representation learning can discover effective features as well as their mappings from data for given tasks. Furthermore, deep learning can learn complex features by combining simpler features learned from data. In other words, with artificial neural networks of multiple nonlinear layers, referred to as deep learning architectures, hierarchical representations of data can be discovered with increasing levels of abstraction [35].

To build a deep learning model, two main steps are required:construction and training of deep learning architectures. Deep learning architectures are basically artificial neural networks of multiple nonlinear layers and several types. In general, it can be categorized into four groups: Deep Neural Networks (DNNs) [22, 23, 56], Convolutional Neural Networks (CNNs) [30, 31, 34], Recurrent Neural Networks (RNNs) [18, 36], and emergent architectures [45].

The goal of training deep learning architectures is to find the best values of the model's parameters to satisfy an objective function. It is usually known as an optimization problem. A single cycle of the optimization process is organized as follows: first, given a training dataset, the forward pass sequentially computes the output in each layer and propagates the function signals forward through the network. In the final output layer, an objective loss function measures

error between the predicted values and the desired values. To minimize the training error, the backward pass uses the chain rule to back-propagate error signals and compute gradients with respect to all weights throughout the neural network. Finally, the weight parameters are updated using optimization algorithms based on stochastic gradient descent (SGD) [6, 21, 33]. Whereas batch gradient descent performs parameter updates for each complete dataset, SGD provides stochastic approximations by performing the updates for each small set of data examples. Several optimization algorithms are adapted from SGD. For instance, Adagrad [10] and Adam [28] perform SGD while adaptively modifying learning rates based on update frequency and moments of the gradients for each parameter, respectively.

RNNs are designed to utilize sequential information of input data with cyclic connections among building blocks like perceptrons, long short-term memory units (LSTMs) [24], or GRUs [7]. The models proposed in this work are RNNs based, so, next section explains briefly their main concepts.

## 2.4 Recurrent Neural Networks

Recurrent Neural Networks(RNNs) are designed to represent sequences, e.g. sentences. A recurrent neural network has basic structure with a cyclic connection (Figure 2.2). Since input data are processed sequentially, recurrent computation is performed in the hidden units where cyclic connection exists. Therefore, past information is implicitly stored in the hidden units called state or internal memory. Thus, the output for the input at the time step $t$ is computed considering all previous inputs using these hidden states.

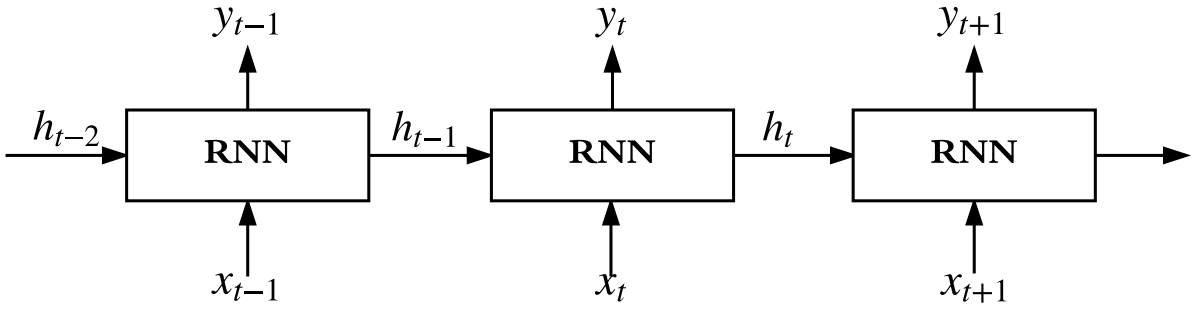


Figure 2.2: Recurrent Neural Network.

As shown in Fig.2, at each time step $t$, it takes the input vector $x \in \mathbb{R}^d$ and the hidden state vector $h_{t-1} \in \mathbb{R}^{d_h}$ and outputs the next hidden state $h_t$ by applying the following equation:

$$h_t = \phi(x_t, h_{t-1}) \tag{2.1}$$

Usually, $h_0$ is initialized to a zero vector in order to calculate the first hidden state. The most common approach is to use the affine transformation operation followed by an element-wise

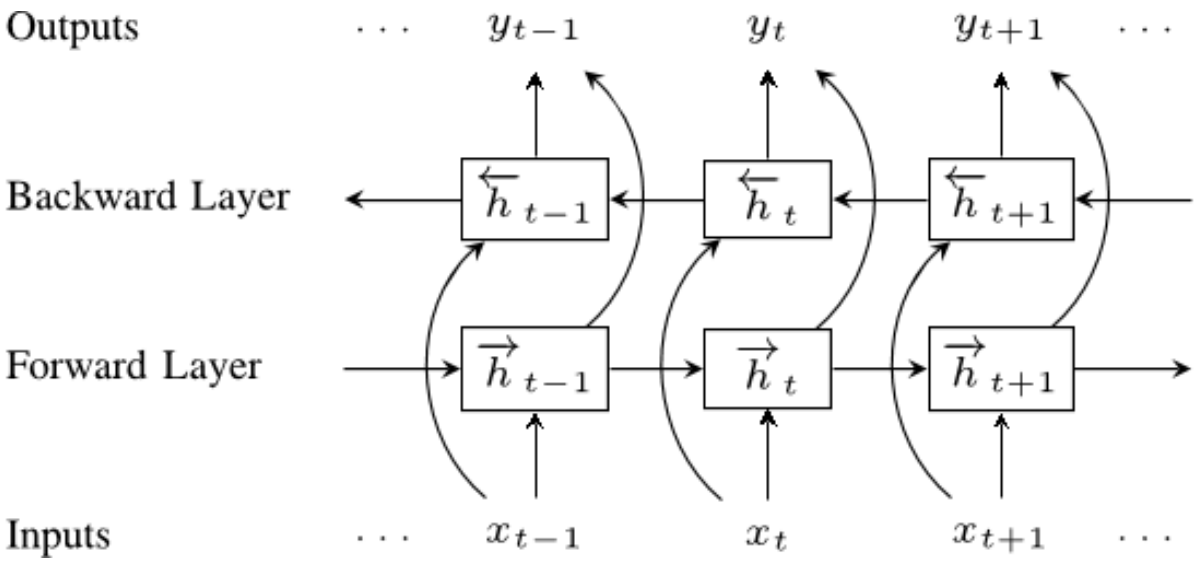


Figure 2.3: Bidirectional Recurrent Neural Network.

non-linearity, e.g. Rectified Linear Unit (ReLU), as the function $\phi$ that produces the next hidden state vector $h_t$. In this formula, $W \in \mathbb{R}^{d \times d_h}$, $V \in \mathbb{R}^{d_h \times d_h}$ and $b \in \mathbb{R}^{d_h}$ are the parameters of the model, and $f$ is an element-wise non-linearity.

$$\phi(x_t, h_{t-1}) = f(Wx_t + Vh_{t-1} + b) \tag{2.2}$$

In practice, the major issue of RNNs using these transition functions is the difficulty of learning long-term dependencies due to vanishing/exploding gradients [4]. LSTM units and GRU have been specifically designed to address this problem. In this work we use a GRU as $\phi$, and we explain how it is used to produce the hidden state vector $h_t$ in the next subsection.

## 2.5 Bidirectional RNNs

The standard RNN, described above, reads an input sequence $X = (x_1, ..., x_n)$ in a forward direction (left-to-right) starting from the first symbol $x_1$ and ending in the last one $x_n$. Thus, it processes sequences in temporal order, ignoring the future context. For many tasks on sequences it is beneficial to have access to future as well as to past information. For example, in text processing, decisions are usually made after the whole sentence is known. The Bidirectional BiRNN architecture [16] proposed a solution for making predictions based on both past and future information.

Figure 2.3 illustrates the architecture of a BiRNN, it consists of forward $\overrightarrow{\phi}$ and backward $\overleftarrow{\phi}$ RNNs. The first one reads the input sequence in a forward direction $(x_1, ..., x_n)$ and produces

a sequence of forward hidden states $(\overrightarrow{h_1},...,\overrightarrow{h_n})$, whereas the former reads the sequence in the reverse order $(x_n,...,x_1)$ resulting in a sequence of backward hidden states $(\overleftarrow{h_n},...,\overleftarrow{h_1})$.

We obtain a representation for each word $x_t$ by concatenating the corresponding forward hidden state $\overrightarrow{h_t}$ and the backward one $\overleftarrow{h_t}$. The following equations illustrate the main ideas:

$$\overrightarrow{h_t} = \overrightarrow{\phi}(x_t, \overrightarrow{h_{t-1}}) \tag{2.3}$$

$$\overleftarrow{h_t} = \overleftarrow{\phi}(x_t, \overleftarrow{h_{t-1}}) \tag{2.4}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{2.5}$$

In this work we use two GRUs, one as $\overrightarrow{\phi}$ and the other as $\overleftarrow{\phi}$. We call this model BiGRU and we explain how to use it to represent a pair of tweets in chapter 4.

## 2.6 Vector Representations of Words

*Word embeddings* are an approach for distributional semantics which represents words as vectors of real numbers. Such representation has useful clustering properties, since the words that are semantically and syntactically related are represented by similar vectors [47]. For example, the words "coffee" and "tea" will be very close in the created space.

When a text has to be analysed, the first step is to map each word into a continuous, low dimensional and real-valued vector, which can later be processed by a neural network model. All the word vectors are stacked into a matrix $E \in \mathbb{R}^{d \times N}$, where $N$ is the vocabulary size and $d$ is the vector dimension. This matrix is called the *embedding layer* or the *lookup table layer*. The embedding matrix can be initialized using a pre-trained model like $word2vec$ or $Glove$ [47, 50].

## 2.7 Summary

In this chapter, we presented a background of the techniques that have been used in this work. We explained briefly the following concepts: Twitter and its features the challenges that exist in Twitter as information and knowledge source, the knowledge structure used in this work (i.e WordNet), deep learning and its characteristics and types, how to build and train a deep learning model specifically an RNN based model, and finally we showed how to represent words in the embedding space. The goal of this overview was to make the understanding of the methodology of this work easier and simpler. The next chapter, we will provide a review and summary of the state of the art on the methods that have been used to analyze and evaluate paraphrase identification of two sentences through social media.

CHAPTER 3

# RELATED WORKS

This chapter describes related work on the PI task. The existing systems and models of PI can be categorized, based on the techniques that have been used, into two approaches: supervised approaches and unsupervised approaches. Next subsections provide a brief survey of these two kinds of methods that have been used.

## 3.1 Supervised Approaches

As we stated, most of the current systems of PI used supervised machine learning approaches in which a classifier is trained on manually annotated (i.e. labeled) data. Set of hand-crafted features, including BoWs, POS tags, clusters mapping, machine translation metrics and some text or POS overlap features have been developed and used to train classifiers such as SVM, Maximum Entropy (MaxEnt), logistic regression, K-Nearest Neighbor (KNN), ... etc. For instance, the authors of the work presented in [14] proposed a SVM classifier. The authors assumed that machine translation is closely related to the task of sentence-level semantic equivalence classification. Thus, they leveraged a set of machine translation features like NIST score, position-independent word error rate,word error rate and BLEU score. They also used POS features.and semantic similarity distance measure computed based on WordNet-based lexical relationship measures.

Qiu et al. (2006) [54] proposed a framework of two-phase for PI. First, they identified the common content information of the pair of sentences using similarity detection. Then, this information was paired using a pairing module. Using a simple matching technique, the predicate arguments were compared. This approach is different from other approaches because it focused on dissimilarities between the pair of sentences. It achieved 72.0% accuracy on MRPC test data using SVM classifier.

Kozareva and Montoyo (2006) [29] used set of features extracted from a combination of lexical and semantic attributes to train three machine-learning classifiers (SVM, KNN and MaxEnt). The lexical attributes were the cardinal number, the proper name, the longest common sub-sequence and n-grams, wheres the semantic similarity features were based on WordNet. The experiments showed that the classifier that used the lexical feature set independently gave better performance than the one that used the similarity feature set, while combining the two features sets enhanced the performance by 1%. The best result between all classifier obtained with SVM.

Ul-Qayyum and Altaf (2012) [57] trained a logistic regression classifier based on two sets of features: monotonic and no-monotonic alignment and semantic heuristics. In their approach, the monotonicity was regarded and implemented as longest common sub-sequence. The approach achieved good accuracy according to state-of-the-art PI systems.

Eyecioglu and Keller (2015) [11] developed a model, called ASOBEK, based on SVM, this model ranked 1st in the PI task of SemEval-2015. In their approach the classifier were trained on simple lexical word overlap and character n-grams features. The experimental results showed the importance of such lexical features and indicated the role they could play in enhancing the results of PI [8].

One of the most interested work of PI is the model presented in [1], the authors of that work proposed a model based on two techniques. The first one was a lexical-semantic net. The second technique was a deep learning model, they proposed a model upon the Deep Structured Semantic Model (DSSM) [25] which is a deep learning based technique that was developed for semantic understanding of textual data. DSSM maps short textual strings, such as sentences, to feature vectors in a low-dimensional semantic space. Then the vector representations were utilized for document retrieval by comparing the similarity between documents and queries. It was reported to outperform other semantic models applying to document retrieval [25]. However, the performance has not been evaluated to measure the degree of similarity in the underlying semantics of paired snippets of text. After obtaining the semantic feature vectors for each paired snippets of text, cosine similarity was utilized to measure the semantic similarity between the pair.

Similar to the work reported above, Potash et al., (2016) [53] developed a deep ensemble system for semantic textual similarity based on four systems: a small feature-based system that leverages word alignment and machine translation quality evaluation metrics, two end-to-end LSTM-based systems, and an ensemble system. The LSTM based systems used either a simple LSTM architecture or a Tree-LSTM structure. The experimental results showed that out of the three base systems, the feature-based model obtained the best results, outperforming each LSTM-based model. Whoever, the ensemble system was able to outperform the base systems substantially.

## 3.2 Unsupervised Approaches

Unlike the supervised approaches, the unsupervised approaches do not need annotated (i.e labeled data) and they mostly depend on external resources such as knowledge repositories. This sections describes some unsupervised models.

Mihalcea, Corley, and Strapparava (2006) [46] proposed a method for measuring the semantic similarity of texts, using corpus-based and knowledge-based measures of similarity. Specifically, they used two corpus-based measures, point-wise mutual information [44] and latent semantic analysis [9], and six types of knowledge-based measures that include: Leacock and Chodorow [32] , Lesk [37] , Wu and Palmer [59] , Resnik [55] , Jiang and Conrath [26] and Lin [38]. These measures of word semantic similarity used to define text-to-text similarity. The approach achieved a good performance with accuracy of 71.5% on the Microsoft paraphrase corpus standard dataset with a threshold prediction value of 0.5.

Fernando and Stevenson (2008) [13] defined semantic similarity matrix between all words pairs from both sentences. The proposed system used WordNet similarity package to compute the similarity degree with a threshold of 0.8 for the similarity decision. The approach achieved good performance with accuracy of 74% on the Microsoft paraphrase corpus standard dataset.

Hassan and Mihalcea (2011) [19] produced a Salient Semantic Analysis model for measuring the semantic relatedness of words. They used salient encyclopedic features taken from encyclopedic knowledge to construct a semantic profile for these words. This method builds on the idea that the meaning of a word can be represented in a salient concept found in its immediate context. It has outstanding performance in comparison with corpus-based and knowledge-based semantic relatedness models.

Milajevs, Kartsaklis, Sadrzadeh, and Purver (2014) [48] compared the neural word embeddings with co-occurrence based word representations in compositional models. They choose the tensor-based compositional models to be implemented. They performed a couple of tasks on a small scale (sentence similarity and verb disambiguation) and on a large scale (paraphrase identification and dialogue act tagging). On the small-scale tasks, the neural vectors gave a result better than or similar to the count based vectors, whereas on the large-scale tasks the neural word embedding gave a result better than the co-occurrence based.

## 3.3 Summary

This chapter summarized the state of the art of the most related work. Two approaches of PI methods are commented: supervised and unsupervised approaches. The methods that use labeled data to train a classifier based on set of designed features or automatically extracted features are categorized as supervised methods, wheres the methods that depend on external resources and do not use annotated data are unsupervised methods.

In this work we combined the two approaches of the features extraction methods: "hand-crafted features" that depend and "embedding features" to design a deep learning model for the PI problem. The next chapters (Chapter 4 and Chapter 5) explain how these methods were used and discuss the results that were obtained.

CHAPTER 4

# METHODOLOGY

This chapter explains the main steps of the proposed system, the tools and the resources that have been used in this work, the features used to describe a pair of tweets and the classification method. Figure 4.1 shows a graphical depiction of the system.

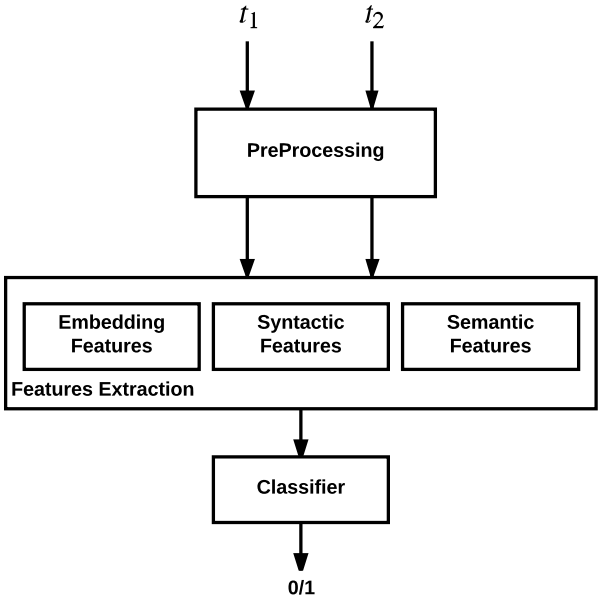


Figure 4.1: Architecture of the Paraphrase Identification System.

First, we pre-process the tweets (section 4.1). Afterwards, three sets of features are used to encode a pair of tweets into a real-valued and fixed length vector (section 4.2). Finally, this vector is passed through a binary classifier to determine whether the pair of tweets is identical (section 4.3).

## 4.1 Pre-Processing

Some standard pre-processing methods are applied on the tweets:

- *Normalization*: Each tweet is converted to the lowercase. URLs, usernames hashtags and emoticons are omitted.
- *Tokenization and POS tagging*: All tweets are tokenized and tagged using the Ark Tweet NLP [15].

## 4.2 Features Extraction

This task receives a pair of tweets $T = (t_1, t_2)$ as an input, where $t_1 = \{w_1^1, w_2^1, ..., w_n^1\}$, $t_2 = \{w_1^2, w_2^2, ..., w_m^2\}$ and $w_j^i$ denotes the $j$th word of the $i$th tweet. In this work, we propose three kind of features : *embedding features*, *syntactic features* and *semantic features* to encode $T$ into a real-valued and fixed length vector.

### 4.2.1 Embedding Features

As we stated, RNNs have the ability to represent sequences, e.g. sentences [40, 41]. However, in practice learning long-term dependencies with a vanilla RNN is difficult due to vanishing/exploding gradients [4]. *Gated Recurrent Units* [7] were designed to have more persistent memory, making them very useful to capture long-term dependencies between the elements of a sequence.

We explain in this section how we used a shared-parameter bidirectional GRU model to represent $T$. We start by mapping each word $w_t^i$ in the input tweet with a vector $x_j^i \in \mathbb{R}^d$. This technique is called *word embedding*. It is an approach for distributional semantics which represents words as vectors of real numbers. Such representation has useful clustering properties, since the words that are semantically and syntactically related are represented by similar vectors [47].

All the word vectors are stacked into a matrix $E \in \mathbb{R}^{d \times N}$, where $N$ is the vocabulary size and $d$ is the vector dimension. This matrix is called the *embedding layer* or the *lookup table layer*. The embedding matrix can be initialized randomly or using a pre-trained model like $word2vec$ or $Glove$ [47, 50]. In this work, we use available pre-trained embeddings which were trained on a large data set. The following modules were used:

- **Glove**: a word embedding model trained on 2 billion tweets from Twitter [50], vectors of 25, 50, 100 and 200 dimensions are provided as part of the pre-trained model. For this work, we use the 200 dimensional vectors.
- **Edinburgh Embeddings**: trained on 10 million tweets for sentiment classification, they provide 400 dimensional vectors [52].

Let $x_1, x_2, ..., x_n$ be the sequence of word vectors of a tweet obtained in the previous step, where $n$ is the length of the tweet. We use two GRU (Figure 4.2 shows the graphical depiction of GRU) neural networks: a *forward-GRU*, which processes the sentence from left to right, and a *backward-GRU*, which processes the sentence in reverse order. Each of the GRU units processes the word vectors sequentially. Starting with an initial state $h_0$, they compute the sequence $h_1, h_2, ...h_n$ as follows:

$$r_t = \sigma(W_r \cdot [h_{t-1}; x_t] + b_r) \tag{4.1}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}; x_t] + b_z) \tag{4.2}$$

$$\widetilde{h}_t = tanh(W_h \cdot [(r_t \odot h_{t-1}); x_t] + b_h) \tag{4.3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \tag{4.4}$$

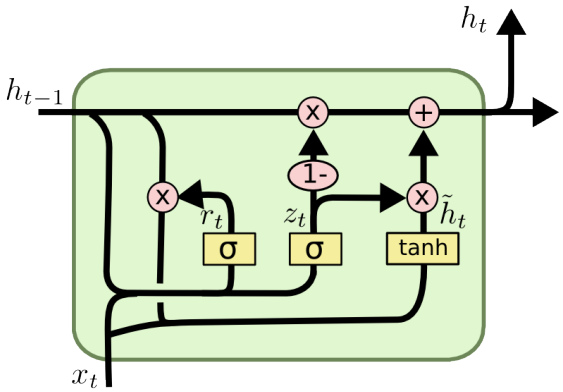


Figure 4.2: Gated Recurrent Unit (GRU)

In these expressions $r_t, z_t$ denote to the *reset* and *update* gates, $\widetilde{h}_t$ is the candidate output state and $h_t$ is the actual output state at time $t$. The symbol $\odot$ stands for element-wise multiplication, $\sigma$ is a sigmoid function and ; stands for the vector-concatenation operation. $W_r, W_z, W_h \in \mathbb{R}^{d_h \times (d+d_h)}$ and $b_r, b_z, b_h \in \mathbb{R}^{d_h}$ are the parameters of the *reset* and *update* gates, where $d_h$ is the dimension of the hidden state. The final states from the forward-GRU and backward-GRU units are denoted by $h_n^f$ and $h_n^b$, respectively. Finally, the input tweet is represented by the concatenation of the vectors $h_n^f$ and $h_n^b$, formally:

$$v = [h_n^f; h_n^b] \tag{4.5}$$

We denote to this model as $BiGRU$. It takes as input a tweet and returns a vector $v$ which is its representation. Thus, the final representation of a pair of tweets $T = (t_1, t_2)$ is given as follows:

$$v_1 = BiGRU(t_1) \tag{4.6}$$

$$v_2 = BiGRU(t_2) \tag{4.7}$$

$$X = [v_1; v_2] \tag{4.8}$$

### 4.2.2 Syntactic Features

The set of syntactic features is extracted from the text and it includes the overlap features based on *n-grams* of the two tweets. For each pair of tweets $T = (t_1, t_2)$, we first compute the uni-grams, bi-grams and tri-grams for $t_1$ and $t_2$, then for each of the three cases we compute the percentage of n-grams' overlapping for $t_1$ ($PerOver1$), and $t_2$ ($PerOver1$) and their combination ($CombOver$), afterwards we use them as features to represent $T$. Let $G$, $G_1$ and $G_2$ denote the number of overlapped n-grams, the number of n-grams in $t_1$ and the number of n-grams in $t_2$ respectively, where $n \in \{uni, bi, tri\}$. We can compute the features as follows:

$$PerOver1(T,n) = \frac{G}{G_1} \tag{4.9}$$

$$PerOver2(T,n) = \frac{G}{G_2} \tag{4.10}$$

$$CombOver(T,n) = 2 \times \frac{PerOver1(T,n) \times PerOver2(T,n)}{PerOver1(T,n) + PerOver2(T,n)} \tag{4.11}$$

### 4.2.3 Semantic Features

This set of features includes the semantic similarity between the two tweets of $T$ (i.e. $t_1$ and $t_2$). It is usually computed with the support of an external knowledge like WordNet [49].

WordNet is a large lexical database of English where the nouns, the verbs, the adjectives and the adverbs are divided into sets of cognitive synonyms called synsets. Each synonym expresses a distinct concept. In this system, we use WordNet APIs provided by NLTK toolkit [5] to calculate the similarity of two tweets using different measures. Table 1 shows the set of measures used in this system.

Table 4.1: Semantic similarity measures.

| Semantic Similarity Measure | Equation | Notes |
|---|---|---|
| Path similarity | $SimPath(c_1, c_2) = 2 * deep\_max - len(c1, c2)$ | c1 and c2 are concepts, deep max is a fixed value len(c1,c2) is the shortest path of concepts c1 an c2 in WordNet. |
| Lch similarity | $SimLch(c_1, c_2) = -log(\frac{len(c1,c2)}{2*deep\_max})$ | It measures two words similarity by using the depth of concepts in the WordNet hierarchy tree. |
| Wup similarity | $SimWup(c_1, c_2) = \frac{2*N3}{N1+N2+2*N3}$ | N1 and N2 are the number of hypernym links from the terms c1 and c2 to their least common subsumer (LCS) in WordNet, respectively, N3 is the number of hypernym links from the LCS to the root of WordNet |

The computation of the semantic features, for each semantic semlarity measure, involves the following steps:

- Find out all the senses of each word according to its POS-tag; put the results into two lists $L1$ and $L2$.
- For each sense $s$ in $L1$, find out the sense in $L2$ that has the maximum similarity with $s$. Add all of the similarity values together, and then average this value with the length of $L1$.
- For each sense $s$ in $L2$, find out the sense in $L1$ that has the maximum similarity with $s$. Add all of the similarity values together, and then average this value with the length of $L2$.
- Compute the harmonic mean of the two average values, and the result is the value of this feature.

## 4.3 Classifier

Once the final vector has been obtained, it is passed into a Multi Layer Perceptron (MLP) binary classifier with one hidden layer to identify whether the pair of tweets is identical. Let $x \in \mathbb{R}^{2d+12}$ be the vector obtained from the previous step, the next equations illustrate our MLP model.

$$z = tanh(x * W_1 + b_1) \tag{4.12}$$

$$y = \sigma(z * W_2 + b_2) \tag{4.13}$$

Where $W_1 \in \mathbb{R}^{(2d+12)\times k}$, $b_1 \in \mathbb{R}^k$, $W_2 \in \mathbb{R}^{k\times 1}$, $b_2 \in \mathbb{R}$ are the MLP parameters and $k$ is the dimensionality of the hidden layer.

## 4.4 Model Training

We trained the model to minimize the following binary cross-entropy:

$$J = -(yt * log(y) + (1 - yt) * log(1 - y)) \tag{4.14}$$

In this expression $yt$ is the desired value and y is the predicted value which is computed by Eq. 4.13. The derivative of the objective function is taken through back-propagation with respect to the whole set of parameters of the model, and these parameters are updated with the stochastic gradient descent. The learning rate is initially set to 0.01 and the parameters are initialized randomly over a uniform distribution in $[-0.03, 0.03]$. For the regularization, dropout [**? ?** ] is used with probability 0.5 on the embedding output to the GRU input and on the concatenation output to the classifier input.

CHAPTER 5

# EXPERIMENTS AND RESULTS

This chapter describes the experiments that were done to evaluate the proposed model. Section 5.1 describes the dataset that has been used in this experiments. In Section 5.2, the evaluation metrics, the results obtained and their analysis are presented.

## 5.1 Dataset

We evaluated the effectiveness of our method by using it in the supervised task of paraphrase detection in Twitter that was presented in SemEval 2015 (PIT-2015) [61]. The statistic description of the dataset is shown in table 5.1.

## 5.2 Results

We used the $F1$ score, precision and recall as evaluation metrics in all the experiments. We compared our system with the top three systems of SemEval 2015. The rows under "A" in Table 5.2 show the results obtained by applying the proposed method with different embedding models, whereas the rows under "B" show the results of our system when we remove the syntactic features, the semantic features and both of them in order to study their effect on model's performance.

Table 5.1: Statistic of PIT-2015 Twitter Paraphrase Corpus. Debatable cases are ignored in this work.

| | # Sent Pair | # Paraphrase | # Non-Paraphrase | # Debatable |
|---|---|---|---|---|
| Train | 13063 | 3996 (30.6%) | 7534 (57.7%) | 1533 (11.7%) |
| Dev | 4727 | 1470 (31.1%) | 2672 (56.5%) | 585 (12.4%) |
| Test | 972 | 175 (18.0%) | 663 (68.2%) | 134 (13.8%) |

Finally, the rows under "C" show the results of the compared systems that reported in the work presented by [61].

Table 5.2: Comparison of our model to the state-of-the–art on Paraphrases identification. Best scores are shown in bold.

| | Precision | Recall | F1 |
|---|---|---|---|
| A. Our system + | | | |
| Glove | 0.710 | 0.724 | 0.717 |
| Edinburgh | 0.732 | 0.720 | **0.726** |
| Random | 0.666 | 0.645 | 0.656 |
| B. All features - | | | |
| Syntactic features | 0.656 | 0.672 | 0.665 |
| Semantic features | 0.630 | 0.633 | 0.634 |
| Syntactic and Semntci features | 0.633 | 0.650 | 0.641 |
| C. State of the art systems | | | |
| ASOBEK | 0.680 | 0.669 | 0.674 |
| MITRE | 0.569 | **0.806** | 0.667 |
| ECNU | **0.767** | 0.583 | 0.662 |
| BASELINE (random) | 0.192 | 0.434 | 0.266 |

It is clearly shown that our system outperforms the state-of-the-art systems of PI in Twitter in terms of $F1$ measure. In terms of $Precision$ ECNU yields the best performance whereas MITRE obtains the best in terms of $Recall$. However, none of them obtains high scores both in precision and recall.

Comparing the different versions of our system, the one with Glove embedding gives the best performance in terms of $Recall$ and the one with Edinburgh embedding gives the best performance in terms of $Precision$. Using the random initialization of the embedding gives results lower than the compared systems. This can be attributed to the fact that deep learning models do not learn well with a low number of training samples, thus using pre-trained models such as the embedding models helps to improve the model's performance.

It can also be observed that our model gives results comparable to those of the state of the art systems when we remove the syntactic and the semantic features. This shows the importance of integrating these features and shows the strength of GRUs in modeling the text.

CHAPTER 6

# CONCLUSION AND FUTURE WORK

We have developed a system that automatically identifies whether a pair of tweets is identical. It contains three main steps. First, some standard pre-processing methods are applied to clean the tweets. Afterwards, three sets of features are used to encode a pair of tweets into a real-valued and fixed length vector. Finally, this vector is passed through a fully connected binary classifier to determine whether the pair of tweets is identical.

The effectiveness of the proposed system has been evaluated by using it in the supervised task of paraphrase identification in Twitter presented in SemEval 2015, obtaining results which show its superiority over the state-of-the-art systems. Recently, neural attention models have been proposed and integrated with recurrent neural networks to be used in many natural language processing tasks such as neural translation [2, 42], question answering [60], sentiment analysis [39, 58], etc, showing great improvement. Thus, we will consider this point in our future work, by designing a system that integrates neural attention models with recurrent neural networks to solve the problem of paraphrase identification in Twitter.

# BIBLIOGRAPHY


[1] N. AFZAL, Y. WANG, AND H. LIU, *Mayonlp at semeval-2016 task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model.*, in SemEval@ NAACL-HLT, 2016, pp. 674–679.

[2] D. BAHDANAU, K. CHO, AND Y. BENGIO, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, (2014).

[3] Y. BENGIO, I. J. GOODFELLOW, AND A. COURVILLE, *Deep learning*, Nature, 521 (2015), pp. 436–444.

[4] Y. BENGIO, P. SIMARD, AND P. FRASCONI, *Learning long-term dependencies with gradient descent is difficult*, IEEE transactions on neural networks, 5 (1994), pp. 157–166.

[5] S. BIRD, E. KLEIN, AND E. LOPER, *Natural language processing with Python: analyzing text with the natural language toolkit*, " O'Reilly Media, Inc.", 2009.

[6] L. BOTTOU, *Stochastic gradient learning in neural networks*, Proceedings of Neuro-Nımes, 91 (1991).

[7] K. CHO, B. VAN MERRIËNBOER, C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK, AND Y. BENGIO, *Learning phrase representations using RNN encoder-decoder for statistical machine translation*, arXiv preprint arXiv:1406.1078, (2014).

[8] D. DAS AND N. A. SMITH, *Paraphrase identification as probabilistic quasi-synchronous recognition*, in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1, Association for Computational Linguistics, 2009, pp. 468–476.

[9] S. DENNIS, T. LANDAUER, W. KINTSCH, AND J. QUESADA, *Introduction to latent semantic analysis*, in Slides from the tutorial given at the 25th Annual Meeting of the Cognitive Science Society, Boston, 2003.

[10] J. DUCHI, E. HAZAN, AND Y. SINGER, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12 (2011), pp. 2121–2159.

[11] A. EYECIOGLU AND B. KELLER, *Asobek: Twitter paraphrase identification with simple overlap features and svms*, Proceedings of SemEval, (2015).

[12] ——, *Twitter paraphrase identification with simple overlap features and svms*, in Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 2015, Association for Computational Linguistics, pp. 64–69.

[13] S. FERNANDO AND M. STEVENSON, *A semantic similarity approach to paraphrase detection*, in Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics, 2008, pp. 45–52.

[14] A. FINCH, Y.-S. HWANG, AND E. SUMITA, *Using machine translation evaluation techniques to determine sentence-level semantic equivalence*, in Proceedings of the Third International Workshop on Paraphrasing (IWP2005), 2005, pp. 17–24.

[15] K. GIMPEL, N. SCHNEIDER, B. O'CONNOR, D. DAS, D. MILLS, J. EISENSTEIN, M. HEILMAN, D. YOGATAMA, J. FLANIGAN, AND N. A. SMITH, *Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments*, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, HLT '11, Stroudsburg, PA, USA, 2011, Association for Computational Linguistics, pp. 42–47.

[16] A. GRAVES, A.-R. MOHAMED, AND G. HINTON, *Speech recognition with deep recurrent neural networks*, in Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, IEEE, 2013, pp. 6645–6649.

[17] A. GRAVES, A. R. MOHAMED, AND G. HINTON, *Speech recognition with deep recurrent neural networks*, in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May 2013, pp. 6645–6649.

[18] A. GRAVES AND J. SCHMIDHUBER, *Offline handwriting recognition with multidimensional recurrent neural networks*, in Advances in neural information processing systems, 2009, pp. 545–552.

[19] S. HASSAN AND R. MIHALCEA, *Semantic relatedness using salient semantic analysis.*, in Aaai, 2011.

[20] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[21] R. HECHT-NIELSEN ET AL., *Theory of the backpropagation neural network.*, Neural Networks, 1 (1988), pp. 445–448.

[22] G. E. HINTON, S. OSINDERO, AND Y.-W. TEH, *A fast learning algorithm for deep belief nets*, Neural computation, 18 (2006), pp. 1527–1554.

[23] G. E. HINTON AND R. R. SALAKHUTDINOV, *Reducing the dimensionality of data with neural networks*, science, 313 (2006), pp. 504–507.

[24] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[25] P.-S. HUANG, X. HE, J. GAO, L. DENG, A. ACERO, AND L. HECK, *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data*, in Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, ACM, 2013, pp. 2333–2338.

[26] J. J. JIANG AND D. W. CONRATH, *Semantic similarity based on corpus statistics and lexical taxonomy*, arXiv preprint cmp-lg/9709008, (1997).

[27] Y. KIM, *Convolutional neural networks for sentence classification*, arXiv preprint arXiv:1408.5882, (2014).

[28] D. KINGA AND J. B. ADAM, *A method for stochastic optimization*, in International Conference on Learning Representations (ICLR), 2015.

[29] Z. KOZAREVA AND A. MONTOYO, *Paraphrase identification on the basis of supervised machine learning techniques*, in FinTAL, Springer, 2006, pp. 524–533.

[30] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *Imagenet classification with deep convolutional neural networks*, in Advances in neural information processing systems, 2012, pp. 1097–1105.

[31] S. LAWRENCE, C. L. GILES, A. C. TSOI, AND A. D. BACK, *Face recognition: A convolutional neural-network approach*, IEEE transactions on neural networks, 8 (1997), pp. 98–113.

[32] C. LEACOCK AND M. CHODOROW, *Combining local context and wordnet sense similarity for word sense identification. wordnet, an electronic lexical database*, 1998.

[33] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, Nature, 521 (2015), pp. 436–444.

[34] Y. LECUN, B. E. BOSER, J. S. DENKER, D. HENDERSON, R. E. HOWARD, W. E. HUBBARD, AND L. D. JACKEL, *Handwritten digit recognition with a back-propagation network*, in Advances in neural information processing systems, 1990, pp. 396–404.

[35] Y. LECUN AND M. RANZATO, *Deep learning tutorial*, in Tutorials in International Conference on Machine Learning (ICML'13), Citeseer, 2013.

[36] P. D. LENA, K. NAGATA, AND P. F. BALDI, *Deep spatio-temporal architectures and learning for protein structure prediction*, in Advances in neural information processing systems, 2012, pp. 512–520.

[37] M. LESK, *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*, in Proceedings of the 5th annual international conference on Systems documentation, ACM, 1986, pp. 24–26.

[38] D. LIN ET AL., *An information-theoretic definition of similarity.*, in Icml, vol. 98, 1998, pp. 296–304.

[39] J. LIU AND Y. ZHANG, *Attention modeling for targeted sentiment*, EACL 2017, (2017), p. 572.

[40] P. LIU, X. QIU, J. CHEN, AND X. HUANG, *Deep fusion lstms for text semantic matching.*, in ACL (1), 2016.

[41] P. LIU, X. QIU, AND X. HUANG, *Recurrent neural network for text classification with multi-task learning*, arXiv preprint arXiv:1605.05101, (2016).

[42] M.-T. LUONG, H. PHAM, AND C. D. MANNING, *Effective approaches to attention-based neural machine translation*, arXiv preprint arXiv:1508.04025, (2015).

[43] N. MADNANI AND B. J. DORR, *Generating phrasal and sentential paraphrases: A survey of data-driven methods*, Computational Linguistics, 36 (2010), pp. 341–387.

[44] C. D. MANNING, H. SCHÜTZE, ET AL., *Foundations of statistical natural language processing*, vol. 999, MIT Press, 1999.

[45] J. MASCI, U. MEIER, D. CIREŞAN, AND J. SCHMIDHUBER, *Stacked convolutional auto-encoders for hierarchical feature extraction*, Artificial Neural Networks and Machine Learning–ICANN 2011, (2011), pp. 52–59.

[46] R. MIHALCEA, C. CORLEY, C. STRAPPARAVA, ET AL., *Corpus-based and knowledge-based measures of text semantic similarity*, in AAAI, vol. 6, 2006, pp. 775–780.

[47] T. MIKOLOV, K. CHEN, G. CORRADO, AND J. DEAN, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, (2013).

[48] D. MILAJEVS, D. KARTSAKLIS, M. SADRZADEH, AND M. PURVER, *Evaluating neural word representations in tensor-based compositional settings*, arXiv preprint arXiv:1408.6179, (2014).

[49] G. A. MILLER, *Wordnet: a lexical database for english*, Communications of the ACM, 38 (1995), pp. 39–41.

[50] J. PENNINGTON, R. SOCHER, AND C. D. MANNING, *Glove: Global vectors for word representation.*, in EMNLP, vol. 14, 2014, pp. 1532–1543.

[51] I. PERIKOS AND I. HATZILYGEROUDIS, *A methodology for generating natural language paraphrases*, in Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference on, IEEE, 2016, pp. 1–5.

[52] S. PETROVIC, M. OSBORNE, AND V. LAVRENKO, *The edinburgh twitter corpus*, in Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, 2010, pp. 25–26.

[53] P. POTASH, W. BOAG, A. ROMANOV, V. RAMANISHKA, AND A. RUMSHISKY, *Simihawk at semeval-2016 task 1: A deep ensemble system for semantic textual similarity*, Proceedings of SemEval, (2016), pp. 741–748.

[54] L. QIU, M.-Y. KAN, AND T.-S. CHUA, *Paraphrase recognition via dissimilarity significance classification*, in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2006, pp. 18–26.

[55] P. RESNIK, *Using information content to evaluate semantic similarity in a taxonomy*, arXiv preprint cmp-lg/9511007, (1995).

[56] D. SVOZIL, V. KVASNICKA, AND J. POSPICHAL, *Introduction to multi-layer feed-forward neural networks*, Chemometrics and intelligent laboratory systems, 39 (1997), pp. 43–62.

[57] Z. UL-QAYYUM AND W. ALTAF, *Paraphrase identification using semantic heuristic features*, Research Journal of Applied Sciences, Engineering and Technology, 4 (2012), pp. 4894–4904.

[58] Y. WANG, M. HUANG, X. ZHU, AND L. ZHAO, *Attention-based lstm for aspect-level sentiment classification.*, in EMNLP, 2016, pp. 606–615.

[59] Z. WU AND M. PALMER, *Verbs semantics and lexical selection*, in Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, 1994, pp. 133–138.

[60] H. XU AND K. SAENKO, *Ask, attend and answer: Exploring question-guided spatial attention for visual question answering*, in European Conference on Computer Vision, Springer, 2016, pp. 451–466.

[61] W. XU, C. CALLISON-BURCH, AND B. DOLAN, *Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit)*, in Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 2015, Association for Computational Linguistics, pp. 1–11.

[62] G. ZARRELLA, J. HENDERSON, E. M. MERKHOFER, AND L. STRICKHART, *Mitre: Seven systems for semantic similarity in tweets*, in Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 2015, Association for Computational Linguistics, pp. 12–17.

[63] J. ZHAO AND M. LAN, *Ecnu: Leveraging word embeddings to boost performance for paraphrase in twitter*, in Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, June 2015, Association for Computational Linguistics, pp. 34–39.