UNIVERSITAT ROVIRA I VIRGILI

MASTER THESIS

# Target-Dependent Sentiment Analysis of Tweets

*Author:*
Fadi HASSAN

*Supervisors:*
Dr. Antonio MORENO
Mohammed JABREEL

*A thesis submitted in partial fulfillment of the requirements for the degree of MASTER IN COMPUTER SECURITY AND ARTIFICIAL INTELLIGENCE*

*in the*

Intellegent Technologies for Advanced Knowledge Acquisition (iTAKA)
Department of Computer Engineering and Mathematics (DEIM), School of Engineering (ETSE)

UNIVERSITAT ROVIRA I VIRGILI

August 27, 2017

# Declaration of Authorship

I, Fadi HASSAN, declare that this thesis titled, "Target-Dependent Sentiment Analysis of Tweets" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

---

Date:

---

"*I would like to express my special thanks of gratitude to my teacher* ***Mohammed Jabreel*** *who gave me the golden opportunity to do this wonderful project, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to him.*"

Universitat Rovira i Virgili

# *Abstract*

Department of Computer Engineering and Mathematics (DEIM), School of Engineering (ETSE)

MASTER IN COMPUTER SECURITY AND ARTIFICIAL INTELLIGENCE

**Target-Dependent Sentiment Analysis of Tweets**

by Fadi HASSAN

The task of target-dependent sentiment analysis aims to identify the sentiment polarity towards a certain target in a given text. All the existing models of this task assume that the target is known. This fact has motivated us to develop an end-to-end target-dependent sentiment analysis system. To the extent of our knowledge, this is the first system that identifies and extract the target of the tweets. The proposed system is composed of two main steps. First, the targets of the tweet to be analysed are extracted. Afterwards, the system identifies the polarities of the tweet towards each extracted target. We have evaluated the effectiveness of the proposed model on a benchmark dataset from Twitter. The experiments show that our proposed system outperforms the state-of-the-art methods for target-dependent sentiment analysis.

# *Acknowledgements*

First and foremost I offer my sincerest gratitude to my supervisor, Dr Antonio Moreno, who has supported me thoughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Masters degree to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

To all my friends, thank you for your understanding and encouragement in my many moments of crisis, Your friendship makes my life a wonderful experience. I cannot list all the names here, but you are always on my mind.

Finally, I wish to thank my parents for their love and encouragement, without whom I would never have enjoyed so many opportunities.

# Contents

**Declaration of Authorship** **3**

**Acknowledgements** **9**

**1 Introduction** **21**
1.1 Target-Dependent Sentiment Analysis . . . 22
1.2 Target Identification . . . 23
1.3 Objectives and Contributions . . . 24
1.4 Document structure . . . 25

**2 Background** **27**
2.1 Machine Learning . . . 27
2.2 Deep Learning . . . 28
2.3 Neural Networks . . . 28
2.4 Recurrent Neural Networks . . . 29
2.5 Gated Recurrent Unit . . . 30
2.6 Bidirectional RNNs . . . 30
2.7 Softmax Classifier . . . 31
2.8 Word Representation . . . 31

**3 State of the Art** **33**
3.1 Sentiment Analysis . . . 33
3.2 Target-Dependent Sentiment Analysis . . . 33
3.3 Target Identification . . . 34

**4 Methodology** **37**
4.1 Target Identification . . . 37
4.2 Target-Dependent Sentiment Analysis . . . 39

**5 Experimental and Results** **41**
5.1 Datasets . . . 41
5.2 Evaluation Metrics . . . 41
5.3 Results and Discussion . . . 42
5.3.1 Target Identification . . . 42
5.3.2 Effects of Word Embeddings . . . 42
5.4 Target-Dependent Sentiment Analysis . . . 42

**6 Conclusion and Future Work** **47**

**Bibliography** **49**

# List of Figures

1.1 Example of the two steps. The filled rectangle represents the target extraction step, where the rounded rectangle represents the sentiment analysis step and it receives two inputs the extracted target and its context. . . . . 22

2.1 Supervised machine learning life cycle. . . . . 27
2.2 Recurrent Neural Network. . . . . 29
2.3 Gated Recurrent Unit (GRU). Figure source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ . . . . . 30
2.4 Bidirectional Recurrent Neural Network. . . . . 31
2.5 Example of one-hot encoding. . . . . 32
2.6 Example of word projection in the embedding space. . . . . 32

4.1 TI-RNC model for target identification of tweets. . . . . 38
4.2 TD-biGRU model for target-dependent sentiment classification. . . . . 39

5.1 F1 score accuracy of TI-RNC with different word embeddings. . . . . 43
5.2 Confusion Matrix . . . . . 45

# List of Tables

5.1 Comparison of our model to the baselines on target identification. Best scores are shown in bold. . . . . 42
5.2 Pre-trained word embedding models. . . . . 43
5.3 Comparison of different methods on target-dependent sentiment classification. Evaluation metrics are accuracy and macro-F1. Best scores are shown in bold. . . . . 44

# List of Abbreviations

| | |
|---|---|
| **SA** | **S**entiment **A**nalysis |
| **NLP** | **N**atural **L**anguage **P**rocessing |
| **RNNs** | **R**ecurrent **N**eural **N**etworks |
| **GRU** | **G**ated **R**ecurrent **U**nit |
| **BiRNN** | **Bi**directional **R**ecurrent **N**ural **N**etworks |
| **BiGRU** | **Bi**directional **G**ated **R**ecurrent **U**nit |
| **TD-biGRU** | **T**arget-**D**ependent **Bi**directional **G**ated **R**ecurrent **U**nit |

*This thesis is dedicated to my parents.*
*For their endless love, support and encouragement*

# Chapter 1

# Introduction

Sentiment analysis (SA) (also known as opinion mining) is the problem of identifying people's opinions, sentiments or attitudes expressed in text. It normally involves the classification of text into categories such as "positive", "negative" and "neutral".

Due to the rapid growth of social networks on the Internet, SA has been applied to analyse opinions on Twitter, Facebook and other digital communities in real time. Sentiment analysis has now a wide range of applications in fields like marketing, management, e-health, politics and tourism (Jabreel and Moreno, 2016; Jabreel, Moreno, and Huertas, 2017; Liu, 2011). For instance, it can enhance the capabilities of customer relationship management systems and recommenders by finding out which features customers are particularly interested in or avoiding the recommendation of items that have received unfavourable feedbacks.

SA can be done at different levels. Coarse-grained analysis attempt to extract the overall polarity on a document or sentence level, whereas, in a fine-grained level of analysis, the problem is to identify the sentiment polarity towards a certain target in a given text (*Target-dependent sentiment analysis*) (Jiang et al., 2011; Dong et al., 2014; Vo and Zhang, 2015). In this problem it is necessary to determine the target and its context, which can be defined as follows:

**Target** A *target* is an entity (person, organisation, product, object, etc.) referred to in a text, about which an opinion is expressed.

**Context** The *context* of the target is the text surrounding it, that provides information about the polarity of the sentiment towards it.

It is quite usual to give several opinions on different aspects of an object in a single sentence. For example, the text *"I have got a new mobile. Its camera is wonderful but the battery life is too short."*, gives both positive and negative remarks about a mobile phone. It may be seen that the example contains three targets (*"mobile"*, *"camera"* and *"battery life"*) and the sentiment polarities towards them can be seen as "neutral", "positive" and "negative", respectively.

As shown in the example, although the text expresses an overall positive opinion about the phone, it also contains conflicting opinions associated with different aspects of the phone. The opinions toward the phone itself and its camera are positive, but the opinion towards its battery life is negative. Thus, the overall opinion of a product is often not enough for decision making and such fine-grained opinions are important for both producers and customers. They want to understand which positive or negative attributes or aspects contribute to the final rating of the product. (Liu et al., 2016).

The importance of target information has been proven by previous studies. It has been shown (Jiang et al., 2011) that about 40% of the errors of sentiment analysis systems are caused by the lack of information about the target. Thus, the target-dependent SA problem can be addressed by designing a system with two steps, shown in Fig.1.1. The first step called *Target Identification*, aims to extract

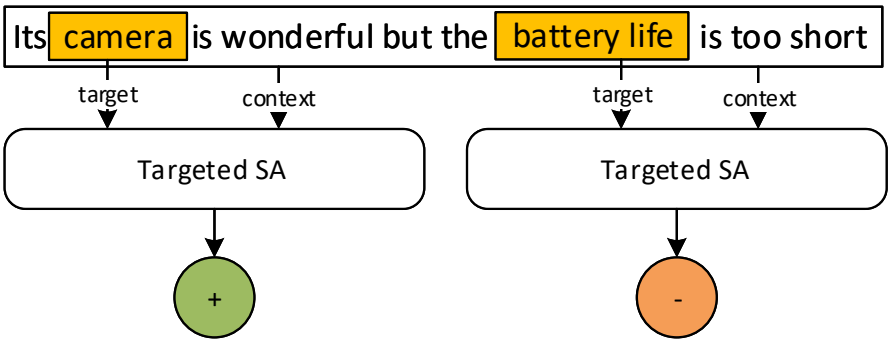


FIGURE 1.1: Example of the two steps. The filled rectangle represents the target extraction step, where the rounded rectangle represents the sentiment analysis step and it receives two inputs the extracted target and its context.

automatically the target in a given text, while the objective of the second step, *Target-Dependent SA*, is to identify the opinion expressed in the text towards the extracted target.

The next sections provides the describtions of these two problems, the most recent systems proposed to address them, their main drawbacks and our contributions to avoid theose drawbacks.

## 1.1 Target-Dependent Sentiment Analysis

*Target-dependent sentiment analysis on Twitter* is the problem of identifying the sentiment polarity towards a certain target in a given tweet. Most of the current studies on sentiment analysis are inspired by the work presented in (Pang, Lee, and Vaithyanathan, 2002). Machine learning techniques have been used to build a classifier from a set of sentences with a manually annotated sentiment polarity. The success of these models is based on two main facts: the availability of a large amount of labeled data and the intelligent manual design of a set of features that can be used to differentiate the samples.

Their performance basically depends on defining an appropriate set of efficient classifying features (Feldman, 2013; Liu, 2012; Pang and Lee, 2008; Perikos and Hatzilygeroudis, 2016). For instance, the authors in (Mohammad, Kiritchenko, and Zhu, 2013) and (Jabreel and Moreno, 2016) used diverse sentiment lexicons and a variety of hand-crafted features in their sentiment analysis systems.

*Target-dependent sentiment analysis* is also regarded as a text classification problem in the literature. Standard text classification approaches can be used to build a sentiment classifier. Extracting syntactic, semantic and sentimental information to represent the relatedness between targets and their contexts in a given text is the key step of targeted sentiment analysis systems. Due to the difficulty of dealing with this step, designing a powerful and robust targeted sentiment analysis system remains a challenge.

This problem can be addressed manually by designing a set of target-dependent features and passing them into feature-based classifiers such as Support Vector Machines (SVM). For instance, the work presented in Jiang et al., 2011 uses a rich set of features over part-of-speech (POS) tags and dependency links of a given text to extract target sentiment polarities. However, this approach has many drawbacks. First, feature engineering is a very intensive and time-consuming task. Second, sparse and discrete features are not good enough in encoding information like the target-context relatedness.

Recently, *neural networks* and *deep learning* approaches have been used to build target-independent and target-dependent sentiment analysis systems. Such systems have the capability of learning automatically a set of features to overcome the drawbacks of the handcrafted approaches (Deriu et al., 2016; Tang et al., 2014a; Tang et al., 2014b).

The most successful targeted sentiment analysis systems that use neural networks rely on the idea of splitting the sentence into three parts (*target*, *left context* and *right context*) with the aim of modeling the interaction between the targets and their contexts. For example, Vo and Zhang, 2015 divided the enclosing sentence into three segments and then they used pooling functions on each part to extract features for the left context, the target and the right context, respectively. These features were then passed through a linear classifier for sentiment classification.

This idea helps to improve modeling the relatedness between the targets and their contexts. However, disconnecting the three parts may cause the loss of some necessary information. For example, let us consider the entity "Facebook" as a target in the following sentence *"Before I used Twitter, I liked Facebook but now I hate it."* The left context ("Before I used Twitter, I liked") contains the word *like* which is positive, so it reflects a positive opinion. The right context ("but now I hate it.") contains the negative word *hate*, so it expresses a negative opinion. Thus, there are two contradictory opinions on the same target in a single sentence.

We believe that it is very important to consider the full sentence when representing the contextual knowledge about the target. This intuition motivated us to investigate a powerful neural network model, which is capable of representing the interaction between the targets and their contexts without losing the connection between the tokens of the text.

*Recurrent neural networks* (RNNs) have been proved to be a very useful technique to represent sequential inputs such as text in the literature. A special extension of recurrent neural networks called *bi-directional recurrent neural network* (BRNN) can capture both the preceding and the following contextual information in a text.

Unlike previous studies, in this work we propose a neural network model based on *gated recurrent units* (GRUs) and a bi-directional recurrent neural network. We have developed a model called *target-dependent bi-directional gated recurrent unit* (TD-biGRU) to deal with the problem of target-dependent sentiment analysis. TD-biGRU models the relatedness between target words and their contexts by concatenating an embedded vector that represents the target word(s) with two vectors that capture both the preceding and the following contextual information.

## 1.2 Target Identification

Extracting the targets from the tweets is the key task in the problem of target dependent SA. However, all the existing studies of this task assume that the target is known. Thus, we have developed a system to identify automatically the explicit targets of the tweets.

Recently, a similar problem to the target identification, known as aspect term extraction, has been studied extensively. There are two main kinds of approaches: *supervised* and *unsupervised*. In the supervised approaches machine-learning systems are trained on manually annotated data to extract targets in the reviews. The most common techniques employed in supervised approaches are decision trees, SVMs, K-nearest neighbour, Naive Bayesian classifiers and neural networks (Toh and Su, 2016; Kessler and Nicolov, 2009). On the other hand, unsupervised approaches aim to automatically extract product features using syntactic and contextual patterns without the need of annotated data (Liu et al., 2015; Liu et al., 2016).

There is one particularly interesting supervised approach, which conceptualizes the aspect extraction problem as a sequence labeling problem (Jebbara and Cimiano, 2016). The most successful sequence labeling systems are probabilistic graphical models such as Hidden Markov Models and Conditional Random Fields (CRFs) (Lafferty, McCallum, and Pereira, 2001; Ratinov and Roth, 2009). However, their main drawback is that they rely heavily on a set of hand-crafted features, whose definition is very time consuming. Recently, deep learning has been utilized to extract automatically high-level features in many tasks such as speech recognition (Graves, Mohamed, and Hinton, 2013a), text classification (Kim, 2014), image classification (He et al., 2016), etc. RNNs have been proved to be a very useful technique to represent sequential data such as text. These models have also shown great success in solving sequence labeling tasks, e.g. Named Entity Recognition (NER) and POS tagging (Lample et al., 2016; Ling et al., 2015). However, these models have their own weakness in solving the sequence labeling problems, as they predict each word label independently and not jointly as part of a sequence.

Recent works on sequence labeling tasks such as NER, POS tagging and others have combined RNNs with the CRFs to leverage their strength and overcome the limitations stated above (Huang, Xu, and Yu, 2015; Wang et al., 2016). Following this approach we propose a model based on Bidirectional Gated Recurrent Units and Conditional Random Fields to identify automatically the targets from the tweets. Specifically, the proposed model consists of two components. The first one is a bidirectional gated recurrent unit which learns automatically a high level feature representation for each word. The second one is a conditional random field that models the whole sequence jointly.

## 1.3 Objectives and Contributions

The main goal of this work is to develop a full fledged target dependent SA system of Twitter that can be used to automatically extract the targets from tweets and identify the sentiments expressed in those tweets toward them. To achieve this goal, the work focuses on the following specific goals:

- Developing a target identification system that can automatically extract the targets from the tweets.
- Developing a target dependent sentiment analysis system that can identify the opinion that expressed in the tweets toward a set of targets.
- Integrating these two systems in one end-to-end target dependent sentiment analysis system.
- Testing the proposed systems on publicly available datasets for the problem of target dependent sentiment analysis.
- Using the proposed systems in real case studies.

The contributions of this work are the following:

- First, we have developed a Bidirectional Gated Recurrent Units and Conditional Random Fields based model to identify automatically the targets from the tweets.
- Second, We have developed a model called *target-dependent bi-directional gated recurrent unit* (TD-biGRU) to deal with the problem of target-dependent sentiment analysis. TD-biGRU models the relatedness between target words and their contexts by concatenating an embedded vector that represents the target word(s) with two vectors that capture both the preceding and the following contextual information.

- Third, we have integrated these two systems into an end-to-end targeted SA system.
- Finally, we have tested the proposed systems on two publicly available datatsets.

The results of these works have been published in the following conferences/books:

- Mohammed Jabreel, **Fadi Hassan** and Antonio Moreno: on Target-Dependent Sentiment Analysis of Tweets using Bi-directional Gated Recurrent Neural Networks, in CIMA-16 Post workshop.
- Mohammed Jabreel, **Fadi Hassan**, Saddam Abdulwahab and Antonio Moreno: on the Bidirectional LSTM-CRF for Target Identification of Tweets, in CCIA

## 1.4 Document structure

The structure of the dissertation is as follows: this chapter has introduced our work. Chapter 2 explains the background of the work. Chapter 3 describes the state of the art. Chapter 4 describes the methodology we use it. Chapter 5 shows the experimental and result. Finally, in chapter 6 we conclude the work, and highlight the future work.

# Chapter 2

# Background

This chapter explains briefly the basic concepts used in this work. We start by explaining Machine Learning, Deep Learning, Neural Networks, and then we describe recurrent neural networks, gated recurrent units, bidirectional recurrent neural networks and the softmax classifier. Finally, we introduce the concept of word representation.

## 2.1 Machine Learning

Machine learning is the subfield of computer science that, according to Arthur Samuel in 1959, gives "computers the ability to learn without being explicitly programmed."(Munoz, 1959) Evolved from the study of pattern recognition and computational learning theory in artificial intelligence.

Machine learning models can solve various type of problems such as classification, regression, and clustering problems, Those models can be categorized to supervised and unsupervised approaches. Supervised machine learning models need labeled datasets in which experts annotate manually set of examples and usually split them into two or sometimes three data sets: training set, development sets and testing set.

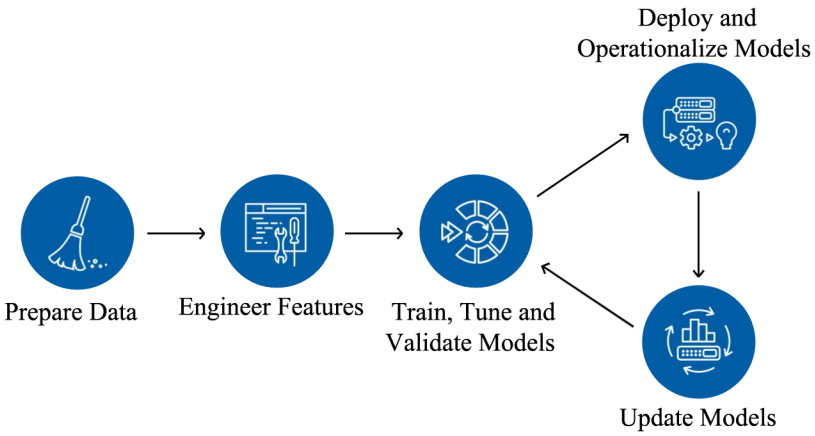


FIGURE 2.1: Supervised machine learning life cycle.

To develop supervised a machine learning model (e.g. a classifier), the following steps are needed: first, a set of features must be designed and used to represent the examples as vectors. Afterward, the computed vectors must be used to train the classifier, the most popular classifiers that have been used recently, especially for sentiment analysis problem, are SVMs, Naive Bayes, Decision Trees, Random Forest and Logistic Regression Jabreel and Moreno, 2016; Pang, Lee, and Vaithyanathan, 2002; Feldman, 2013; Mohammad, Kiritchenko, and Zhu, 2013. Usually, the classifier has a set of

parameters that must be tuned; thus, some techniques like grid-search and k-fold cross validation can be used to select the parameters that give the best performance based on the development set. Later, the effectiveness of the model can evaluated by applying it on the testing set. Figure 2.1 illustrates the life cycle of building a supervised machine learning model.

In unsupervised learning techniques the end goal less clear-cut than predicting an output based on a corresponding input. There are three fundamental problems of unsupervised learning: data clustering, matrix factorization, and sequential models for order-dependent data. Some applications of these models include object recommendation and topic modeling.

## 2.2 Deep Learning

Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised or unsupervised.

Some representations are loosely based on interpretation of information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain. Research attempts to create efficient systems to learn these representations from large-scale, unlabeled data sets.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to many fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation and bioinformatics where they produced results comparable to and in some cases superior to human experts Goodfellow, Bengio, and Courville, 2016.

Unlike the traditional supervised machine learning models, deep leaning models do not require to design and define manually the set of features that will be used to train the models. They can automatically learn high level features from the row data by stacking many neural layers, the lowest layers learn low level features while the highest layers learn high level once. One interesting example of this approach is the CNN LeCun et al., 1998 which shows impressive results on many fields such as computer vision Krizhevsky, Sutskever, and Hinton, 2012 and natural language processing Conneau et al., 2016; Kim, 2014. RNN is another useful example of deep learning models which has been proposed model the sequential data (e.g. text) effectively Choi, Cho, and Bengio, 2017. RNN is the key of the models that proposed in this work. Next sections explain the main deep learning concepts that are used in this work.

## 2.3 Neural Networks

Neural Networks or Artificial neural networks (ANNs) are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve performance) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They have found most use in applications difficult to express in a traditional computer algorithm using rule-based programming.

An ANN is based on a collection of connected units called artificial neurons, (analogous to axons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically

between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Further, they may have a threshold such that only if the aggregate signal is below (or above) that level is the downstream signal sent.

Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times.

The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as backpropagation, or passing information in the reverse direction and adjusting the network to reflect that information.

Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnosis and in many other domains.

## 2.4 Recurrent Neural Networks

A recurrent neural network is a type of neural network architecture specifically designed for modeling sequential inputs of varying lengths such as text.

As shown in Figure 2.2, 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:

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

Usually, $h_0$ is initialized to a zero vector in order to calculate the first hidden state. The most common

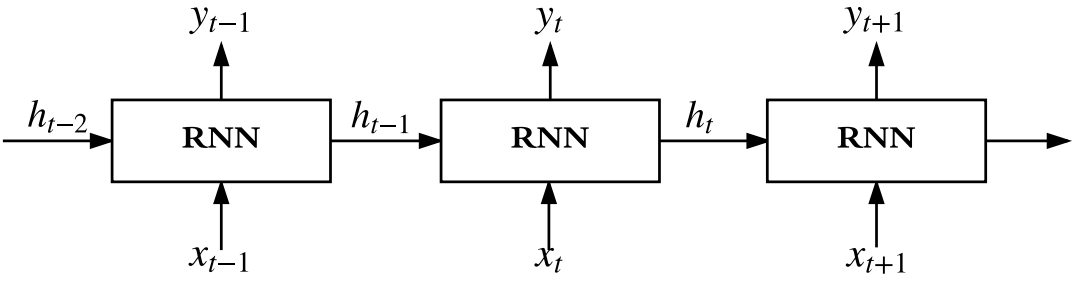


FIGURE 2.2: Recurrent Neural Network.

approach is to use the affine transformation operation followed by an element-wise non-linearity, e.g. Rectified Linear Unit (ReLU), as the function $\phi$ that produces the next hidden state vector $h_t$.

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

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.

In practice, the major issue of RNNs using these transition functions is the difficulty of learning long-term dependencies due to vanishing/exploding gradients (Bengio, Simard, and Frasconi, 1994). Long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997) and GRU (Cho et al., 2014) 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.

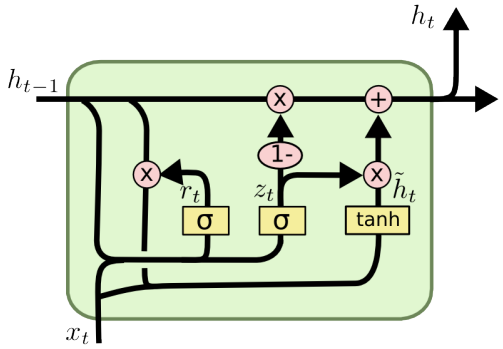


FIGURE 2.3: Gated Recurrent Unit (GRU). Figure source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

## 2.5 Gated Recurrent Unit

*Gated recurrent units* were designed to have more persistent memory, making them very useful to capture long-term dependencies between the elements of a sequence. GRUs are the basic components of the models proposed in this work. Figure 2.3 shows a graphical depiction of a gated recurrent unit.

This kind of units have *reset* ($r_t$) and *update* ($z_t$) gates. The former has the ability to completely reduce the past hidden state $h_{t-1}$ if it considers that it is irrelevant to the computation of the new state, whereas the later is responsible for determining how much of $h_{t-1}$ should be carried forward to the next state $h_t$.

The output $h_t$ of a GRU depends on the input $x_t$ and the previous state $h_{t-1}$, and it is computed as follows:

$$r_t = \sigma\left(W_r \cdot [h_{t-1}; x_t] + b_r\right) \tag{2.3}$$

$$z_t = \sigma\left(W_z \cdot [h_{t-1}; x_t] + b_z\right) \tag{2.4}$$

$$\widetilde{h}_t = tanh\left(W_h \cdot [(r_t \odot h_{t-1}); x_t] + b_h\right) \tag{2.5}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \tag{2.6}$$

In these expressions $r_t$ and $z_t$ denote 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.

## 2.6 Bidirectional RNNs

The standard RNN, described in subsection 2.4, 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 (Graves, Mohamed, and Hinton, 2013b) proposed a solution for making predictions based on both past and future information.

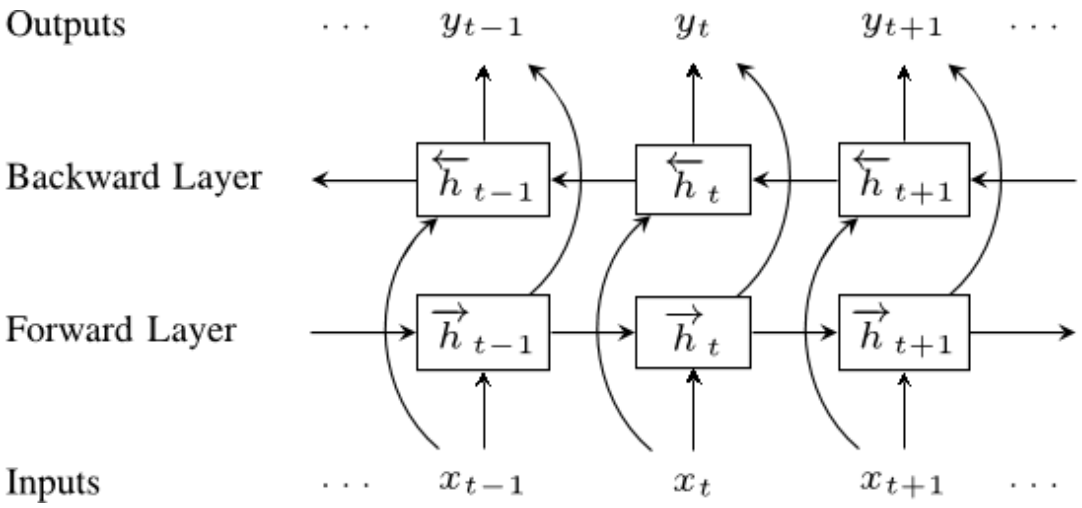


FIGURE 2.4: Bidirectional Recurrent Neural Network.

Figure 2.4 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:

$$\overrightarrow{h_t} = \overrightarrow{\phi}(x_t, \overrightarrow{h_{t-1}}) \tag{2.7}$$

$$\overleftarrow{h_t} = \overleftarrow{\phi}(x_t, \overleftarrow{h_{t-1}}) \tag{2.8}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{2.9}$$

In this work we use two GRUs, one as $\overrightarrow{\phi}$ and the other as $\overleftarrow{\phi}$. We call this model biGRU.

## 2.7 Softmax Classifier

The softmax classifier is a feed-forward neural network followed by the softmax function, which is used for multi-class classification (under the assumption that the classes are mutually exclusive). It takes as input a vector $v \in \mathbb{R}^m$ and produces the probabilities for each class as follows:

$$p(y = i|v;W,b) = \frac{exp(w_i^T v + b_i)}{\sum_{j=1}^{C} exp(w_j^T v + b_j)}, i = 1,2,...,C \tag{2.10}$$

This can be interpreted as the (normalized) probability assigned to each class $i$ given the input vector $v$, and parameterized by $W \in \mathbb{R}^{m \times C}$ and $b \in \mathbb{R}^C$, where $C$ is the number of classes, $w_i$ is the $i$-th column of $W$ and $b_i$ is a bias term.

## 2.8 Word Representation

Word Representation is the process of representing each word as a vector. The most simple method of encoding words is called one-hot or 1-of-N vector representation. In this method each word is represented as an $\mathbb{R}^{|V| \times 1}$ vector with all 0s and one 1 at the index of that word in the sorted vocabulary.

In this notation, $|V|$ is the size of the vocabulary. Word vectors in this type of encoding for the vocabulary {King, Queen, Man, Woman, Child} would appear as shown in Figure 2.5.

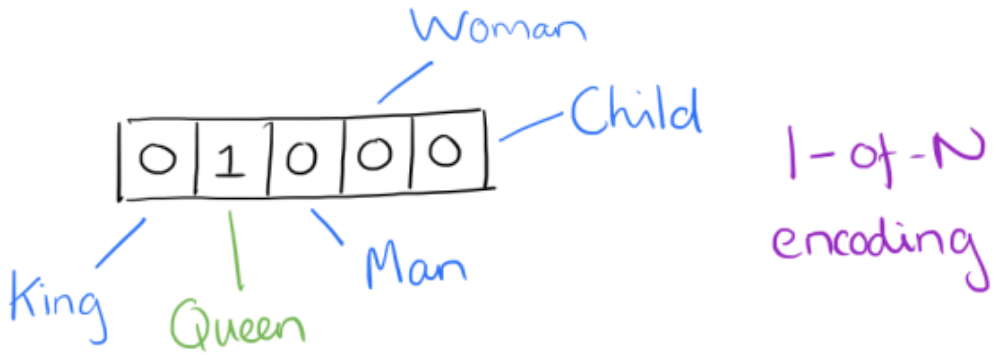


FIGURE 2.5: Example of one-hot encoding.

In the one-hot vector representation method, every word is equidistant from every other. However, it lacks to preserve any relationship among them and leads to data sparsity. Using word embeddings can overcome some of these drawbacks. Word embeddings are an approach for distributional semantics which represents words as vectors of real numbers.

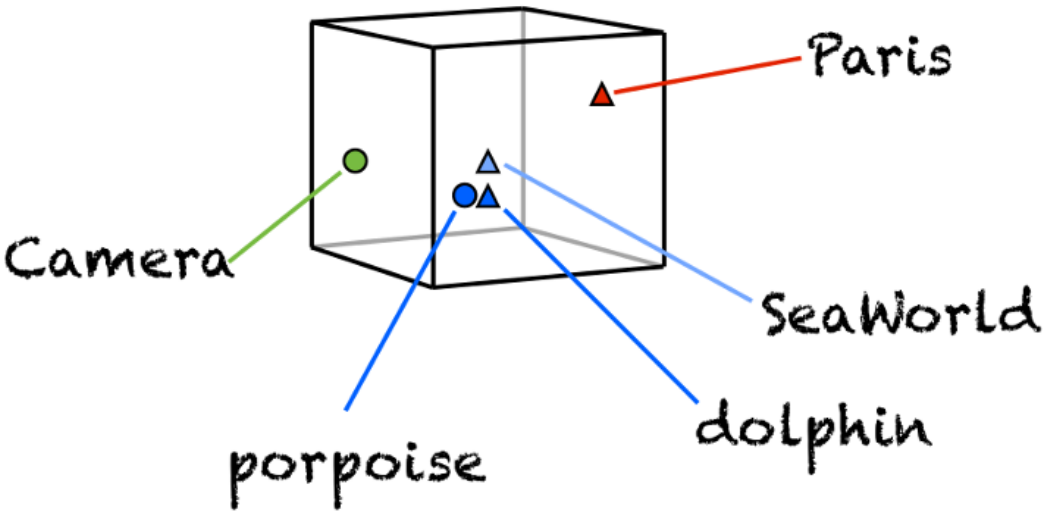


FIGURE 2.6: Example of word projection in the embedding space.

Such representation has useful clustering properties, since it groups together words that are semantically and syntactically similar Mikolov et al., 2013. For example, the words "seaworld" and "dolphin" will be very close in the embedding space (Figure 2.6). The main aim of this step is to map each word into a continuous, low dimensional and real valued vector and use it as input to a model such as a RNN, a CNN , etc.

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* (Mikolov et al., 2013; Pennington, Socher, and Manning, 2014).

# Chapter 3

# State of the Art

This chapter explains briefly the state-of-the-art studies related to this work. We start by reviewing the approaches used in *sentiment analysis* and then we summarize the existing models on *target-dependent sentiment analysis*, Finally, we review the most related works to the *target identification*.

## 3.1 Sentiment Analysis

Most of the current studies on sentiment analysis are inspired by the work presented in (Pang, Lee, and Vaithyanathan, 2002). Machine learning techniques have been used to build a classifier from a set of sentences with a manually annotated sentiment polarity. The success of the machine learning models is based on two main facts: the availability of a large amount of labeled data and the intelligent manual design of a set of features that can be used to differentiate the samples.

In this approach, most studies have focused on designing a set of efficient features to obtain a good classification performance (Feldman, 2013; Liu, 2012; Pang and Lee, 2008). For instance, the authors in (Mohammad, Kiritchenko, and Zhu, 2013) and (Jabreel and Moreno, 2016) used diverse sentiment lexicons and a variety of hand-crafted features in their sentiment analysis systems.

Neural network and deep learning approaches have recently been used to build supervised, unsupervised and semi-supervised methods to analyze the sentiment of texts and to build efficient opinion lexicons (Severyn and Moschitti, 2015; Tang et al., 2014a; Tang et al., 2014b). The main advantage of neural models is their capability to learn a continuous text representation from data without any feature engineering. For example, the work presented in (Severyn and Moschitti, 2015) trained a CNN to learn the best features and used it to classify the sentiment of the tweets. The work in (Tang et al., 2014b) proposed a model to learn sentiment-specific word embeddings, which were combined with a set of state-of-the-art hand-crafted features to learn a deep model system.

Most of the previous studies on sentiment analysis have two main steps. First, they use continuous and real-valued vectors learned from scratch to represent the words (Bengio et al., 2003; Mikolov et al., 2013; Pennington, Socher, and Manning, 2014; Tang et al., 2014b; Liu, Joty, and Meng, 2015). Then, they learn a sentence representation by using a compositional approach like *recursive networks* (Socher et al., 2013), *convolutional neural networks* (Kim, 2014), and *recurrent neural networks* (Liu, Joty, and Meng, 2015).

## 3.2 Target-Dependent Sentiment Analysis

As we stated, *Target-dependent sentiment analysis* is also regarded as a text classification problem in the literature. Standard text classification approaches such as feature-based Support Vector Machines (Pang, Lee, and Vaithyanathan, 2002; Jiang et al., 2011) can be used to build a sentiment classifier.

For instance, (Jiang et al., 2011) manually designed target-independent features and target-dependent features with expert knowledge, a syntactic parser and external resources.

Recent studies, such as the works proposed by Dong et al., 2014 , Vo and Zhang, 2015, Tang et al., 2015 and Zhang, Zhang, and Vo, 2016, use neural network methods and encode each sentence in a continuous and low-dimensional vector space without feature engineering. Dong et al., 2014 transformed a sentence dependency tree into a target-specific recursive structure, and used an *Adaptive Recursive Neural Network* to learn a higher level representation. Vo and Zhang, 2015 used rich features including sentiment-specific word embedding and sentiment lexicons. The work presented by Zhang, Zhang, and Vo, 2016 modeled the interaction between the target and the surrounding context using a gated neural network. Tang et al., 2015 developed long short-term memory models to capture the relatedness of a target word with its context words when composing the continuous representation of a sentence. Most of those studies rely on the idea of splitting the sentence/text into target, left context and right context.

Unlike previous studies, we propose a *target-dependent bi-directional gated recurrent unit* (TD-biGRU), which is capable of modeling the relatedness between target words and their contexts by concatenating an embedded vector that represents the target word(s) with two vectors that capture both the preceding and following contextual information. The next chapter describes the proposed models in detail.

## 3.3 Target Identification

Aspect extraction from reviews is the most similar problem to the task of target identification of tweets. This task has been proved to be an important step in opinion mining to generate list of objects and the opinions that are expressed toward them. Aspect extraction from opinionated text was first introduced by Hu and Liu, 2004. Although, the authors introduced the distinction between explicit and implicit aspects. they only dealt with the explicit aspects by adopting set of rules based on statistical observations. This method has been improved by Popescu and Etzioni, 2007. The authors assumed the product class to be known as priori. Their algorithm detects whether a noun or noun phrase is a product feature or not by computing the point-wise mutual information between the noun phrase and the product class. The work presented by Scaffidi et al., 2007 also tried to improve Hu and Liu's idea by proposing a method that uses a language model to identify product features, assuming that product features are the more frequent in product reviews than in general natural language text. Recent studies on aspect extraction have shown that the syntactical approach, which employs rules about grammar dependency relations between opinion words and aspects, performs quite well. This approach is highly desirable in practice because it is unsupervised and domain independent. However, the rules need to be carefully selected and tuned manually so as not to produce too many errors. Although it is easy to evaluate the accuracy of each rule automatically, it is not easy to select a set of rules that produces the best overall result due to the overlapping coverage of the rules. Liu et al., 2015 proposed a novel method to select an effective set of rules.

Aspect extraction can be treated as a sequential labeling problem. The most popular methods in this context, in particular, are Hidden Markov Models and Conditional Random Fields. Jin, Ho, and Srihari, 2009 used a lexicalized HMM for joint extraction of opinions along with their explicit aspects. Jakob and Gurevych, 2010 used CRF to extract explicit aspects in a custom corpus with data of different domains. Li et al., 2010 and Choi and Cardie, 2010 also used CRF for extraction of explicit aspects.

Recently, deep learning methods have been proposed for aspect extraction. For example, Liu, Joty, and Meng, 2015 proposed a recurrent neural network on top of pre-trained word embeddings. Yin et al., 2016 developed an unsupervised embedding method to encode dependency path into a recurrent neural network to learn high-level features for words. Those feature were taken as input features for CRFs to extract automatically the aspects from reviews. Wang et al., 2017 proposed a coupled multi-Layer attentions for co-extraction of aspect and opinion terms. Their model provides an end-to-end solution for this task and does not require any parsers or other linguistic resources for preprocessing.

Similarly, target identification problem can also be regarded as a sequential labeling problem. Thus, we propose a model based on Bidirectional Gated Recurrent Units and Conditional Random Fields to identify automatically the targets from the tweets. Next chapter explains in details our models for target identification and targeted SA.

# Chapter 4

# Methodology

We describe in this chapter the proposed model to tackle the problem of target-dependent SA. It is composed of two main steps. First, the target of the tweet to be analysed is identified as described in next section. Once the target has been obtained, it is passed together with the tweet as input to the model described in section 4.2 to determine the sentiment polarity.

## 4.1 Target Identification

We describe in this section the proposed model of the problem of target identification in Twitter. Target identification can be typically regarded as a kind of sequence labeling problem in which the text (i.e. a sequence of words) can be represented using the IOB2 tagging scheme Sang and Veenstra, 1999. The idea is that each word in a given text is labeled by one of the tags *I*, *O*, or *B*, which indicate if the word is inside, outside, or at the beginning of a target respectively.

In this work we propose a model based on Bidirectional Recurrent Neural Networks, Gated Recurrent Units and Conditional Random Fields to extract the targets from a given tweet. This model, called TI-RNC, reads a sequence of words and predicts a sequence of corresponding IOB2 tags. Once we have the predicted sequence of IOB2 tags for a tweet, we can interpret it and extract the targets.

Figure 4.1 shows an example of the application of the proposed model for the problem of target identification. It consists of two main sub-models. The first one, called BiGRU, which learns a high-level representation of the words in a tweet. The second one is a CRF model, which takes the produced sequence states to model whole tag sequence.

The main steps of the BiGRU sub-model are the following. First, the words of the input sentence are mapped to vectors of real numbers resulting in a sequence of vectors $x_1, x_2, ...x_n$. Afterwards, the resulting sequence is passed to a bidirectional recurrent neural network to produce a sequence of recurrent states $h_1, h_2, ..., h_n$. These states are the high-level representation of the input words.

Sequence labeling (tagging) can be simply modeled by using the $h_t$'s as features to make an independent tagging decision at each time $t$ (Ling et al., 2015). However, in such tasks it is beneficial to consider the correlation between labels in neighborhoods, specifically when there are strong dependencies across the output tags. For example, in the problem of target identification, the tag **B** is more likely to be followed by the tag **I**. Thus, instead of modeling tagging decisions independently, we model them jointly using a CRF (Lafferty, McCallum, and Pereira, 2001).

Formally, let $H = \{h_1, h_2, ..., h_n\}$ be the sequence of vectors to be labeled, which is produced by the BiGRU sub-model, and $y = \{y_1, y_2, ..., y_n\}$ is the corresponding tag sequence. Each element $y_i$ of $y$ is one of the *B*, *I* or *O* tags. Both $H$ and $y$ are assumed to be random variables and they are jointly modeled. The entire model can be represented as an undirected graph $G = (V, E)$ with cliques $C$.

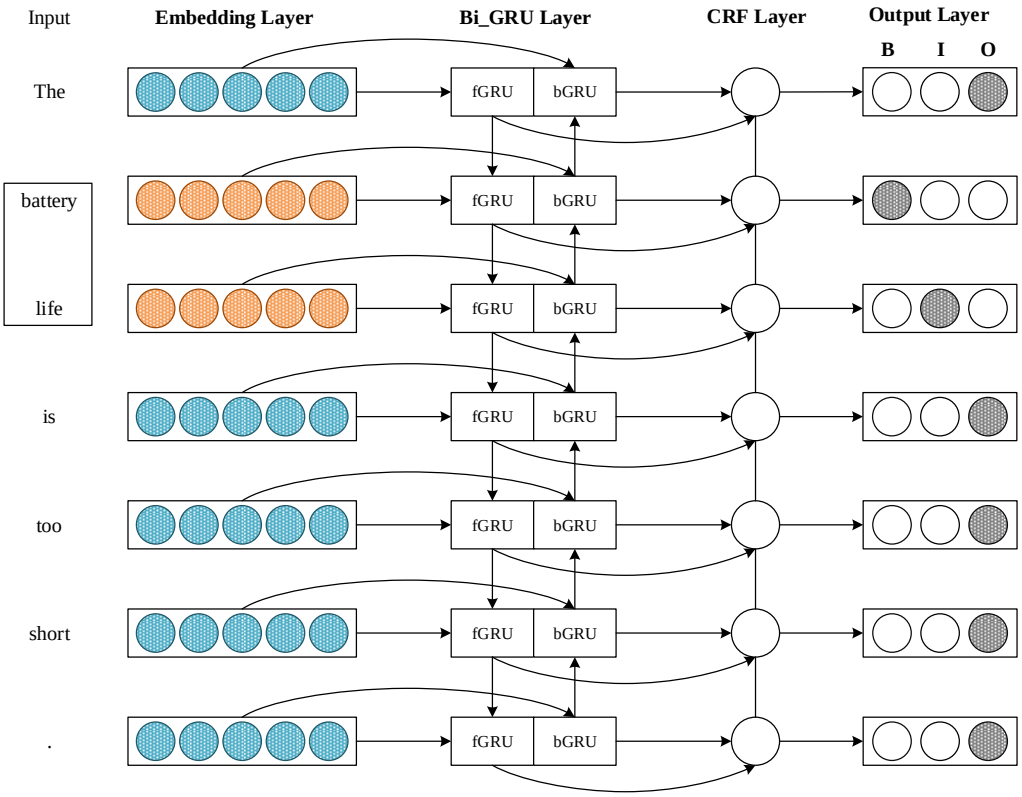


FIGURE 4.1: TI-RNC model for target identification of tweets.

In this work we employed a linear-chain CRF, where $G$ is a simple chain or line: $G = (V = \{1,2,...,n\}, E = \{(i,i+1)\})$. It has two different cliques (i.e. $C = \{P,M\}$): a unary clique ($P$) representing the input-output connection, and a pairwise clique ($M$) representing the adjacent output connection. We consider $P$ to be the matrix of output scores, where $P_{i,j}$ corresponds to the score of the $j^{th}$ tag of the $i^{th}$ word in a sentence and it is computed as follows:

$$P_{i,j} = W_{i,j} h_i + b_j \; ; \; j = 1,2,3 \tag{4.1}$$

In this equation, the parameters are $W_{i,j} \in \mathbb{R}^{2*d_h}$ and $b_j \in \mathbb{R}^1$, where $d_h$ is the dimensionality size of the hidden state.

The clique $M$ is considered to be the matrix of transition scores such that $M_{i,j}$ represents the score of a transition from the tag $i$ to the tag $j$. Given that, we define the score function of the sequence of predictions as follows:

$$s(H,y) = \sum_{i=1}^{n} P_{i,y_i} + \sum_{i=0}^{n} M_{y_i,y_{i+1}} \tag{4.2}$$

In this expression $y_0$ and $y_{n+1}$ denotes the start and the end tags of the sentence, that we add to the set of possible tags.

A softmax over all possible tag sequences ($Y^*$) on a sequence $H$ yields a probability for the sequence $y$ as follows:

$$p(y|H) = \frac{e^{s(H,y)}}{\sum_{\tilde{y} \in Y} e^{s(H,\tilde{y})}} \tag{4.3}$$

During training, we minimize the negative log-probability of the correct tag sequence:

$$J = -log\,(p(y|H)) = -s(H,y) + log\left(\sum_{\tilde{y} \in Y^*} e^{s(H,\tilde{y})}\right) \tag{4.4}$$

The derivative of the objective function $J$ is taken through back-propagation with respect to the whole set of parameters of the model, which are the transition matrix $M$, the parameters of the BiGRU model and the parameters of the matrix $P$ that defined in Eq. 4.1. The parameters are optimized using the stochastic gradient descent (SGD) with a learning rate of 0.005. To reduce the effects of gradient exploding, we set the clipping threshold of the gradient as 5. We apply a dropout Hinton et al., 2012 between the embedding layer and the recurrent layer with probability of 0.5 to reduce the over-fitting.

During inference, we search for the output sequence $y^*$ that obtains the highest probability given by:

$$y^* = \arg\max_{\tilde{y} \in Y^*} p(\tilde{y}|H) \tag{4.5}$$

In this model, Eq. 4.4 and Eq. 4.5 can be solved efficiently using dynamic programming.

## 4.2 Target-Dependent Sentiment Analysis

Figure 4.2 shows the proposed model for the problem of target-dependent sentiment classification. Its main steps are the following. First, the words of the input sentence are mapped to vectors of real numbers. Then, the input sentence is represented by a real-valued vector using the TD-biGRU encoder by concatenating the vectors $\overrightarrow{h_n}, \overleftarrow{h_n}$ and $x_v$, formally:

$$X = [\overrightarrow{h_n}; x_v; \overleftarrow{h_n}] \tag{4.6}$$

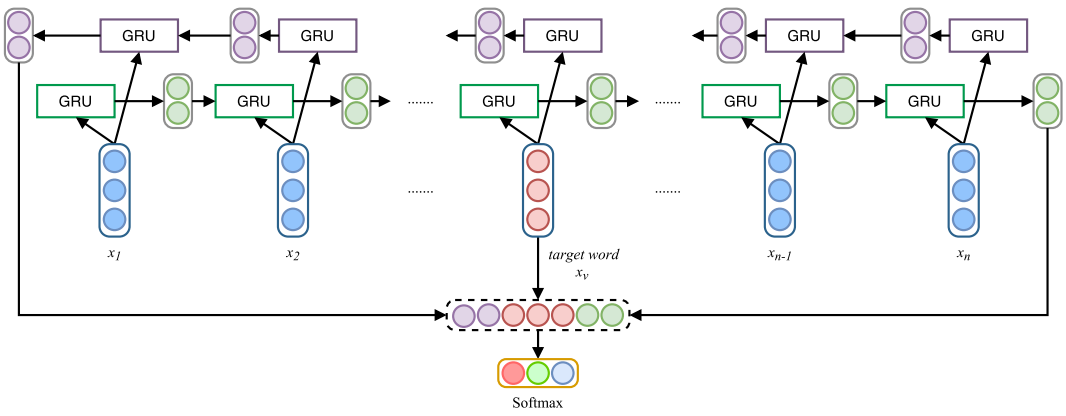


FIGURE 4.2: TD-biGRU model for target-dependent sentiment classification.

Here $\overrightarrow{h_n}$ and $\overleftarrow{h_n}$ are the last forward hidden state and the last backward one obtained by TD-biGRU. The $x_v$ is the vector representation of the target word(s). If the target is a single word, its representation is the embedding vector of that word. If the target is composed of multiple words, such as "battery life", its representation is the average of the embedding vectors of the words (Sun et al., 2015).

In this way, the obtained vector summarizes the input sentence and contains semantic, syntactic and/or sentimental information based on the word vectors. Finally, this vector is passed through a softmax classifier to classify the sentence into positive, negative or neutral.

We trained the system to minimize the following categorical cross-entropy:

$$J = -\sum_{s \in S} \sum_{c=1}^{3} G_c(s) log(P(y = c|s)) \tag{4.7}$$

In this expression $S$ is the training set and $G_c(s) \in \{0,1\}$ is the ground-truth function which indicates whether class $c$ is the correct sentiment category for sentence $s$.

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 0.1 and the parameters are initialized randomly over a uniform distribution in $[-0.03, 0.03]$. For the regularization, dropout layers (Hinton et al., 2012; Srivastava et al., 2014) are used with probability 0.5 on the lookup-table output to the GRU input and on the concatenation output to the softmax input.

# Chapter 5

# Experimental and Results

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

## 5.1 Datasets

We evaluated the effectiveness of the proposed model of target identification problem (i.e. TI-RNC) by applying it on two benchmarks of tweets. The first one is the dataset used in Dong et al., 2014, called "T-Dataset" in this work. It contains 6248 training examples and 692 examples in the testing set. The second benchmark, provided by Zhang, Zhang, and Vo in (Zhang, Zhang, and Vo, 2016), contains 9489 training examples, 1036 development examples and 1170 testing examples. This dataset is called "Z-Dataset" in this work. Each example in these datasets contains the tweet and the target. Both datasets were used in the problem of target dependent sentiment analysis of tweets in the previous works. Whereas the effectiveness of the proposed models of targeted SA have been evaluated by using them on the T-Dataset.

## 5.2 Evaluation Metrics

The evaluation metrics of the target identification problem are the precision (the number of correct targets divided by the number of all returned targets), recall (the number of correct targets divided by the number of targets that should have been returned) and $F_1$ (the harmonic mean of precision and recall), which can be defined as follows:

$$Precision = \frac{|S \cap G|}{|S|} \tag{5.1}$$

$$Recall = \frac{|S \cap G|}{|G|} \tag{5.2}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{5.3}$$

Here $S$ is the set of the predicted targets that the system returned for all the test examples, and $G$ is the set of the gold (correct) targets.

The evaluation metrics of the target-dependent sentiment analysis system are the classification accuracy (the percentage of examples that are correctly classified) and the Macro-F1 measure (the averaged F1 measure over the three sentiment classes).

## 5.3 Results and Discussion

### 5.3.1 Target Identification

We investigated the effectiveness of TI-RNC, which is used to automatically identify the target from a tweet, by comparing it with the baseline models listed below.

Following the approach used in (Ling et al., 2015) for sequence tagging, in which the tagging model uses the hidden state $h_t$ produced by a RNN as a feature to make an independent tagging decision for each word $w_t$ in the sentence, we compare the proposed system with the following baseline methods:

- **RNN**: is the standard recurrent neural network.
- **Bi-RNN**: is a bidirectional version of the RNN model.
- **GRU**: is a RNN based on gated recurrent units.
- **Bi-GRU**: a bidirectional version of the GRU model.

Experimental results of the baseline models and the proposed model are given in Table 5.1. It is clearly shown that TI-RNC outperforms the other models. Another interesting observation from the reported results is that among the models described in this paper, the basic RNN approach has the worst performance. This is not surprising and such conclusion confirms the effectiveness of BiRNNs in this kind of tasks.

TABLE 5.1: Comparison of our model to the baselines on target identification. Best scores are shown in bold.

| Model | T-Dataset | | | Z-Dataset | | |
|---|---|---|---|---|---|---|
| | Precision ($P$) | Recall ($R$) | $F_1$ | Precision ($P$) | Recall ($R$) | $F_1$ |
| RNN | 77.90 | 87.57 | 82.44 | 73.93 | 52.65 | 61.50 |
| BiRNN | 79.76 | 90.17 | 84.67 | 81.00 | 50.43 | 62.16 |
| GRU | 81.18 | 90.89 | 86.10 | 73.98 | 54.19 | 62.56 |
| BiGRU | 87.39 | 91.18 | 89.25 | 79.82 | 60.51 | 68.84 |
| TI-RNC | **95.30** | **94.02** | **94.66** | **82.46** | **64.27** | **72.24** |

### 5.3.2 Effects of Word Embeddings

Since it is well known that the efficiency of the word embedding helps to improve the composition of a powerful text representation at a high level, in this subsection we study the effects of different word embeddings on the models' performance. We compare four publicly available pre-trained word embeddings, as well as a randomly initialized one (see Table 5.2). Figure 5.2 provides a graphical report of the performance of our model with each embedding method. According to these results, the variations of our model that used a pre-trained embedding model obtain a significant improvement in compare to the random embedding. We can find that Glove-100 dimensional word embeddings perform better than 50-dimensional word vectors, while Glove-200 and Word2Vec do not show significant improvements.

## 5.4 Target-Dependent Sentiment Analysis

We compared the proposed model with the state-of-the-art methods used in the task of target-dependent sentiment classification, including:

TABLE 5.2: Pre-trained word embedding models.

| Embedding | Dimension | Algorithm | Refrence |
|---|---|---|---|
| Glove-50 | 50 | Glove | Pennington, Socher, and Manning, 2014 |
| Glove-100 | 100 | Glove | Pennington, Socher, and Manning, 2014 |
| Glove-200 | 200 | Glove | Pennington, Socher, and Manning, 2014 |
| Word2Vec | 300 | SkipGram | Mikolov et al., 2013 |
| Random | 100 | | |

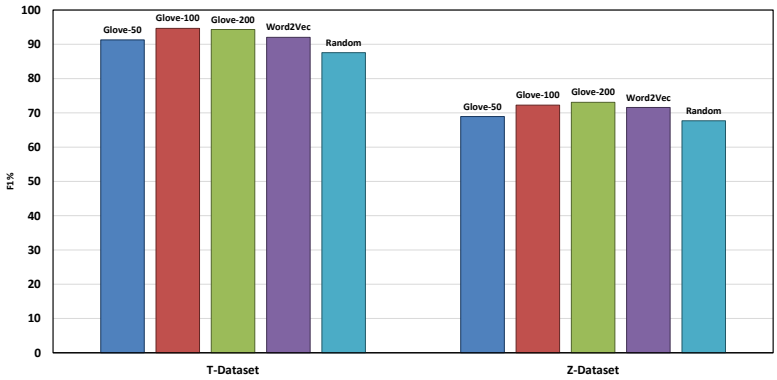


FIGURE 5.1: F1 score accuracy of TI-RNC with different word embeddings.

- **SVM-indep**: SVM classifier built with target-independent features, such as unigram, bigram, punctuations, emoticons, hashtags and the numbers of positive or negative words in the General Inquirer sentiment lexicon (Jiang et al., 2011).

- **SVM-dep**: SVM-indep model extended by adding a set of features that represent the target (Jiang et al., 2011).

- **Recursive RNN**: a recursive neural network is employed to learn the feature representation of the examples over a transferred target-dependent dependency tree (Dong et al., 2014).

- **AdaRNN**: extension of the recursive RNN which uses more than one composition function and adaptively selects them according to the input (Dong et al., 2014). AdaRNN has three variations: AdaRNN-w/oE, AdaRNN-w/E and AdaRNN-comb. Unlike AdaRNN-w/oE, AddRNN-w/E model uses the dependency type in the process of composition function selection. AddaRNN-comb combines the root vectors obtained by AdaRNN-w/E with the unigram and bigram features, and then they are fed into a SVM classifier.

- **Target-ind/Target-dep**: SVM classifiers based on a rich set of target-independent and target-dependent features (Vo and Zhang, 2015). This model has an extension, called **Target-dep+**, in which sentiment lexicon features have been incorporated.

- **LSTM**, **TD-LSTM**, **TC-LSTM**: these methods are based on the *long short-term memory* model (LSTM) proposed by (Tang et al., 2015). In the LSTM model the target is ignored. The idea behind TD-LSTM is to use two LSTM neural networks, so that the left one represents the preceding context plus the target and the right one represents the target plus the following context. TC-LSTM is an extension of TD-LSTM in which a vector that represents the target is concatenated to each context word.

The values under the section "A" in Table ?? represent the results of the baseline model (basic bi-directional gated recurrent units - biGRU - without incorporating target information), the new TD-biGRU model in case the targets are manually given and the results when we apply the two steps of our system to analyse the tweets. Each tweet is passed to the system to first extract the targets and then identify the sentiment polarities towards these targets. Section "B" contains the results of the compared models (obtained from their associated papers). With the exception of AdaRNN, each approach presented in Table ?? has a target-independent version (which does not incorporate any information about targets) and two or three target-dependent versions. For instance, in our case biGRU is the target-independent version.

TABLE 5.3: Comparison of different methods on target-dependent sentiment classification. Evaluation metrics are accuracy and macro-F1. Best scores are shown in bold.

| Model | Accuracy | Macro-F1 |
|---|---|---|
| A.Our model | | |
| biGRU | 69.94 | 68.40 |
| TD-biGRU | **72.25** | **70.47** |
| End-To-End-TD | 70.08 | 68.22 |
| B. State-of-the-art systems | | |
| SVM-indep | 62.70 | 60.20 |
| SVM-dep | 63.40 | 63.30 |
| Recursive NN | 63.00 | 62.80 |
| AdaRNN-w/oE | 64.90 | 64.44 |
| AdaRNN-w/E | 65.80 | 65.50 |
| AdaRNN-comb | 66.30 | 65.90 |
| Target-ind | 67.30 | 66.40 |
| Target-dep | 69.70 | 68.00 |
| Target-dep$^+$ | 71.10 | 69.90 |
| LSTM | 66.50 | 64.70 |
| TD-LSTM | 70.80 | 69.00 |
| TC-LSTM | 71.50 | 69.50 |

As it can be observed from the reported results, the target-independent models (SVM-indep, Target-indep, LSTM and biGRU) have a worst performance than the corresponding models that consider the target information (SVM-dep, Target-dep*, TD-LSTM, TC-LSTM and TD-biGRU). This conclusion confirms the fact that ignoring the target information causes about 40% of sentiment analysis errors (Jiang et al., 2011). It may also be notived that neural-based models perform better than the feature-based SVM classifiers.

The novel TD-biGRU model outperforms the state-of-the-art models both in terms of accuracy and Macro-F1. Our end-to-end approach gives a comparable results to those models, including our TD-biGRU model, that assume the target is known.

To get more insight on this result, we analyzed the confusion matrix given by the TD-biGRU model to figure out which are the most common incorrect cases. Figure 5.2 shows the confusion matrix obtained by applying TD-biGRU. As observed, the matching between the true and the predicted labels is quite high (matrix diagonal). Out of the 192 misclassified samples, 76 (39.6%) of them were misclassified between negative and neutral (i.e., either negative samples were misclassified as neutral or viceversa) and 31 (16.1%) samples were misclassified between negative and positive. The number of samples misclassified between positive and neutral is 85 (44.3%).

This analysis shows that most of the misclassified examples are related to the neutral category. We believe that this problem can be handled by adding more information (e.g. lexicon information). We

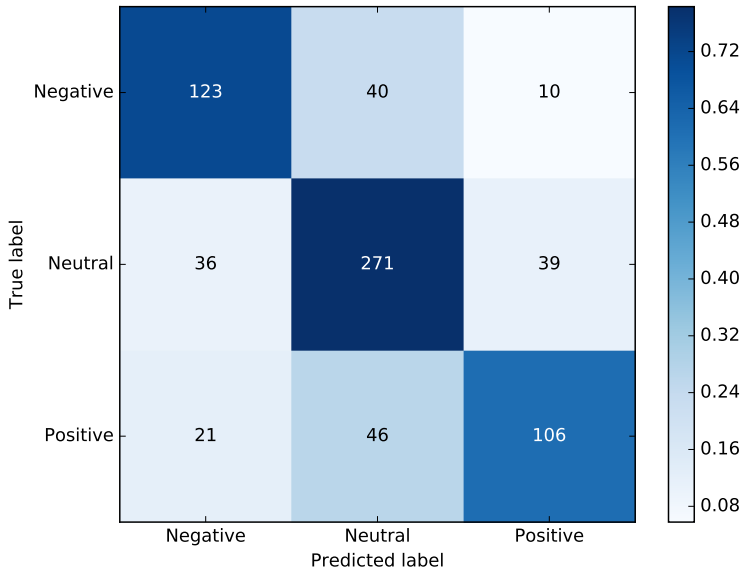


FIGURE 5.2: Confusion Matrix

leave the study of this hypothesis for the future work.

# Chapter 6

# Conclusion and Future Work

We have developed a system that automatically identifies the target of a tweet. It contains two main components. The first one is a bidirectional gated recurrent unit which learns automatically a high level feature representation for each word. The second one is a conditional random field that models the whole sequence jointly. The effectiveness of the proposed system has been evaluated on two benchmarks of tweets, obtaining results which show its superiority over several baseline methods. The reported results show that the proposed system could be a good first step in a targeted sentiment analysis system of tweets. Our system extracts only the targets that are mentioned explicitly in the tweets. However, it is sometimes recognized that targets are mentioned implicitly in tweets and they are detected from the context. Thus, we will consider this point in our future work, by designing a system that can detect both the explicit targets and the implicit targets that are not mentioned in the tweets.

We also have developed an end-to-end target-dependent Twitter sentiment analysis system. The proposed model has the ability of identifying and extracting the target of the tweets, representing the relatedness between the targets and its contexts and identifying the polarities of the tweets towards the targets. The effectiveness of the proposed system has been evaluated on a benchmark of tweets, obtaining results that outperform the state-of-the-art models. The confusion matrix of the results obtained by TD-biGRU shows that most of the misclassified examples are related to the neutral category.

In the future work we plan to extend our system to handle this weakness by integrating more information such as lexicon information and/or the dependency tree. Our system extracts only the targets that are mentioned explicitly in the tweets. However, it is sometimes recognized that targets are mentioned implicitly in tweets and they are detected from the context. Thus, we will consider this point in our future work, by designing a system that can detect both the explicit targets and the implicit targets that are not mentioned in the tweets. Although joint learning of all subsystems has been proved to be useful in natural language processing and text analysis tasks, in this work we have trained each subsystem (i.e. the target identification and the targeted SA) independently and we have combined them in the inference step. Thus, we plan to extend our system and apply this learning technique.

# Bibliography


Bengio, Yoshua, Patrice Simard, and Paolo Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2, pp. 157–166.

Bengio, Yoshua et al. (2003). "A neural probabilistic language model". In: *journal of machine learning research* 3.Feb, pp. 1137–1155.

Cho, Kyunghyun et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078*.

Choi, Heeyoul, Kyunghyun Cho, and Yoshua Bengio (2017). "Context-dependent word representation for neural machine translation". In: *Computer Speech & Language* 45, pp. 149–160.

Choi, Yejin and Claire Cardie (2010). "Hierarchical sequential learning for extracting opinions and their attributes". In: *Proceedings of the ACL 2010 conference short papers*. Association for Computational Linguistics, pp. 269–274.

Conneau, Alexis et al. (2016). "Very deep convolutional networks for text classification". In: *arXiv preprint arXiv:1606.01781*.

Deriu, Jan et al. (2016). "SwissCheese at SemEval-2016 Task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision". In: *Proceedings of SemEval*, pp. 1124–1128.

Dong, Li et al. (2014). "Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Vol. 2. Association for Computational Linguistics, pp. 49–54.

Feldman, Ronen (2013). "Techniques and applications for sentiment analysis". In: *Communications of the ACM* 56.4, pp. 82–89.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*.

Graves, A., A. r. Mohamed, and G. Hinton (2013a). "Speech recognition with deep recurrent neural networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton (2013b). "Speech recognition with deep recurrent neural networks". In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, pp. 6645–6649.

He, K. et al. (2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

Hinton, Geoffrey E et al. (2012). "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hu, Minqing and Bing Liu (2004). "Mining and summarizing customer reviews". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 168–177.

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF models for sequence tagging". In: *arXiv preprint arXiv:1508.01991*.

Jabreel, Mohammed and Antonio Moreno (2016). "SentiRich: Sentiment Analysis of Tweets Based on a Rich Set of Features". In: *Artificial Intelligence Research and Development - Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence, Barcelona, Catalonia, Spain, October 19-21, 2016*, pp. 137–146. DOI: 10.3233/978-1-61499-696-5-137. URL: http://dx.doi.org/10.3233/978-1-61499-696-5-137.

Jabreel, Mohammed, Antonio Moreno, and Assumpció Huertas (2017). "Do Local Residents and Visitors Express the Same Sentiments on Destinations Through Social Media?" In: *Information and Communication Technologies in Tourism 2017*. Springer, pp. 655–668.

Jakob, Niklas and Iryna Gurevych (2010). "Extracting opinion targets in a single-and cross-domain setting with conditional random fields". In: *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 1035–1045.

Jebbara, Soufian and Philipp Cimiano (2016). "Aspect-Based Sentiment Analysis Using a Two-Step Neural Network Architecture". In: *Semantic Web Evaluation Challenge*. Springer, pp. 153–167.

Jiang, Long et al. (2011). "Target-dependent twitter sentiment classification". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pp. 151–160.

Jin, Wei, Hung Hay Ho, and Rohini K Srihari (2009). "A novel lexicalized HMM-based learning framework for web opinion mining". In: *Proceedings of the 26th annual international conference on machine learning*, pp. 465–472.

Kessler, Jason S and Nicolas Nicolov (2009). "Targeting Sentiment Expressions through Supervised Ranking of Linguistic Configurations." In: *ICWSM*.

Kim, Yoon (2014). "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Lafferty, John, Andrew McCallum, Fernando Pereira, et al. (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: *Proceedings of the eighteenth international conference on machine learning, ICML*. Vol. 1, pp. 282–289.

Lample, Guillaume et al. (2016). "Neural architectures for named entity recognition". In: *arXiv preprint arXiv:1603.01360*.

LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Li, Fangtao et al. (2010). "Structure-aware review mining and summarization". In: *Proceedings of the 23rd international conference on computational linguistics*. Association for Computational Linguistics, pp. 653–661.

Ling, Wang et al. (2015). "Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1520–1530.

Liu, Bing (2011). “Opinion mining and sentiment analysis”. In: *Web Data Mining*. Springer, pp. 459–526.

— (2012). “Sentiment analysis and opinion mining”. In: *Synthesis lectures on human language technologies* 5.1, pp. 1–167.

Liu, Pengfei, Shafiq Joty, and Helen Meng (2015). “Fine-grained opinion mining with recurrent neural networks and word embeddings”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Liu, Qian et al. (2015). “Automated Rule Selection for Aspect Extraction in Opinion Mining.” In: *IJCAI*, pp. 1291–1297.

— (2016). “Automated rule selection for opinion target extraction”. In: *Knowledge-Based Systems* 104, pp. 74–88.

Mikolov, Tomas et al. (2013). “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781*.

Mohammad, Saif, Svetlana Kiritchenko, and Xiaodan Zhu (2013). “NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets”. In: *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*. Atlanta, Georgia, USA.

Munoz, Andres (1959). “Machine Learning and Optimization”. In:

Pang, Bo and Lillian Lee (2008). “Opinion Mining and Sentiment Analysis”. In: *Found. Trends Inf. Retr.* 2.1-2, pp. 1–135. ISSN: 1554-0669. DOI: 10.1561/1500000011. URL: http://dx.doi.org/10.1561/1500000011.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). “Thumbs Up?: Sentiment Classification Using Machine Learning Techniques”. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. EMNLP ’02. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 79–86. DOI: 10.3115/1118693.1118704. URL: http://dx.doi.org/10.3115/1118693.1118704.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Perikos, Isidoros and Ioannis Hatzilygeroudis (2016). “Recognizing emotions in text using ensemble of classifiers”. In: *Engineering Applications of Artificial Intelligence* 51, pp. 191–201.

Popescu, Ana-Maria and Orena Etzioni (2007). “Extracting product features and opinions from reviews”. In: *Natural language processing and text mining*. Springer, pp. 9–28.

Ratinov, Lev and Dan Roth (2009). “Design challenges and misconceptions in named entity recognition”. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 147–155.

Sang, Erik F and Jorn Veenstra (1999). “Representing text chunks”. In: *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 173–179.

Scaffidi, Christopher et al. (2007). “Red Opal: product-feature scoring from reviews”. In: *Proceedings of the 8th ACM conference on Electronic commerce*. ACM, pp. 182–191.

Severyn, Aliaksei and Alessandro Moschitti (2015). “UNITN: Training deep convolutional neural network for Twitter sentiment classification”. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pp. 464–469.

Socher, Richard et al. (2013). "Recursive deep models for semantic compositionality over a sentiment treebank". In: *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631. Citeseer, p. 1642.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Sun, Yaming et al. (2015). "Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, pp. 1333–1339. ISBN: 978-1-57735-738-4. URL: http://dl.acm.org/citation.cfm?id=2832415.2832435.

Tang, Duyu et al. (2014a). "Coooolll: A Deep Learning System for Twitter Sentiment Classification". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, pp. 208–212. URL: http://www.aclweb.org/anthology/S14-2033.

Tang, Duyu et al. (2014b). "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 1555–1565. URL: http://www.aclweb.org/anthology/P14-1146.

Tang, Duyu et al. (2015). "Target-Dependent Sentiment Classification with Long Short Term Memory". In: *arXiv preprint arXiv:1512.01100*.

Toh, Zhiqiang and Jian Su (2016). "NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using Neural Network Features". In: *Proceedings of SemEval*, pp. 282–288.

Vo, Duy-Tin and Yue Zhang (2015). "Target-dependent twitter sentiment classification with rich automatic features". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 1347–1353.

Wang, Wenya et al. (2016). "Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis". In: *arXiv preprint arXiv:1603.06679*.

— (2017). "Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms." In:

Yin, Yichun et al. (2016). "Unsupervised word and dependency path embeddings for aspect term extraction". In: *arXiv preprint arXiv:1605.07843*.

Zhang, Meishan, Yue Zhang, and Duy-Tin Vo (2016). "Gated Neural Networks for Targeted Sentiment Analysis". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA. Association for the Advancement of Artificial Intelligence*, pp. 3087–3093.