# Target-dependent Sentiment Analysis of Tweets using a Bi-directional Gated Recurrent Unit

Mohammed Jabreel[1,2] and Antonio Moreno[1]

[1] *Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA), Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Av. Països Catalans, 26, 43007 Tarragona, Spain*

[2] *Computer Science Department, Hodeidah University, Alduraihimi, 1820 Hodeidah, Yemen*

Keywords: Twitter, Target-dependent Sentiment Analysis, Recurrent Neural Networks.

Abstract: Targeted sentiment analysis classifies the sentiment polarity towards a certain target in a given text. In this paper, we propose a target-dependent bidirectional gated recurrent unit (TD-biGRU) for target-dependent sentiment analysis of tweets. The proposed model has the ability to represent the interaction between the targets and their contexts. We have evaluated the effectiveness of the proposed model on a benchmark dataset from Twitter. The experiments show that our proposed model outperforms the state-of-the-are methods for target-dependent sentiment analysis.

## 1 INTRODUCTION

*Automated sentiment analysis* is the problem of identifying opinions expressed in text. It normally involves the classification of text into categories such as positive, negative and neutral. Opinions are central to almost all human activities and they are key influencers of our behaviors.

Due to the ubiquity of the Internet and the recent emergence of social networks, sentiment analysis has been applied to analyze opinions on Twitter, Facebook or 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 (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.

*Target-dependent sentiment analysis* is the problem of identifying the opinion polarities towards a certain target in a given text (Jiang et al., 2011; Dong et al., 2014; Vo and Zhang, 2015). A*target* is an entity (person, organisation, product, object, etc.) referred to in a text, about which an opinion is expressed. The *context* of the target is the text surrounding it, that provides information about the polarity of the sentiment towards it. For example, the sentence *"I have got a new mobile. Its camera is wonderful but the battery life is too short"* has three targets (*"mobile"*, *"camera"* and *"battery life"*) and the sentiment polarities towards them can be seen as "neutral", "positive" and "negative", respectively.

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.

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



Jabreel, M. and Moreno, A.
Target-dependent Sentiment Analysis of Tweets using a Bi-directional Gated Recurrent Unit.
DOI: 10.5220/0006299900800087
In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 80-87
ISBN: 978-989-758-246-2

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.

In this paper we propose a neural network model based on *gated recurrent units* (GRU) 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.

The proposed model has been evaluated on a benchmark dataset from Twitter. Experiments show that our proposed model outperforms the state-of-the-art methods for target-dependent sentiment analysis. Empirical results prove that considering the full text to represent the contextual information and integrating it with the target information improves significantly the classification accuracy. We used Keras (Chollet, 2015) with theano back-end (Theano Development Team, 2016) to implement the model. We plan to make our model and the source code publicly available to be used by other researchers that work on sentiment analysis.

The rest of the paper is structured as follows. Section 2 presents some related work found in the literature of target-dependent sentiment analysis. In Section 3 the proposed model is described. The experiments and results are presented and discussed in Section 4. Finally, in the last section the conclusions and lines of future work are outlined.

## 2 RELATED WORK

This section 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*.

### 2.1 Sentiment Analysis

Most of the current studies on sentiment analysis are inspired by the work presented in (Pang et al., 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 et al., 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 *convolutional neural network* (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 handcrafted 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 et al., 2014; Tang et al., 2014b; Liu et al., 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 et al., 2015).

### 2.2 Target-dependent Sentiment Analysis

*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 et al., 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 et al., 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 in (Zhang et al., 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 section describes the proposed model in detail.

## 3 MODEL DESCRIPTION

Figure 1 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. This step is called vector representation of words or *word embedding* (subsection 3.1). Afterwards, the input sentence is represented by a real-valued vector using the TD-biGRU encoder (subsection 3.2). This 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 (subsection 3.3).

### 3.1 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 (Mikolov et al., 2013). For example, the words ”coffee” and ”tea” will be very close in the created space.

The main aim of this 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 et al., 2014). In this work, the embedding layer contains a pre-trained model which was learned using the *Glove* algorithm (Pennington et al., 2014) on a large corpus of two billions of tweets.

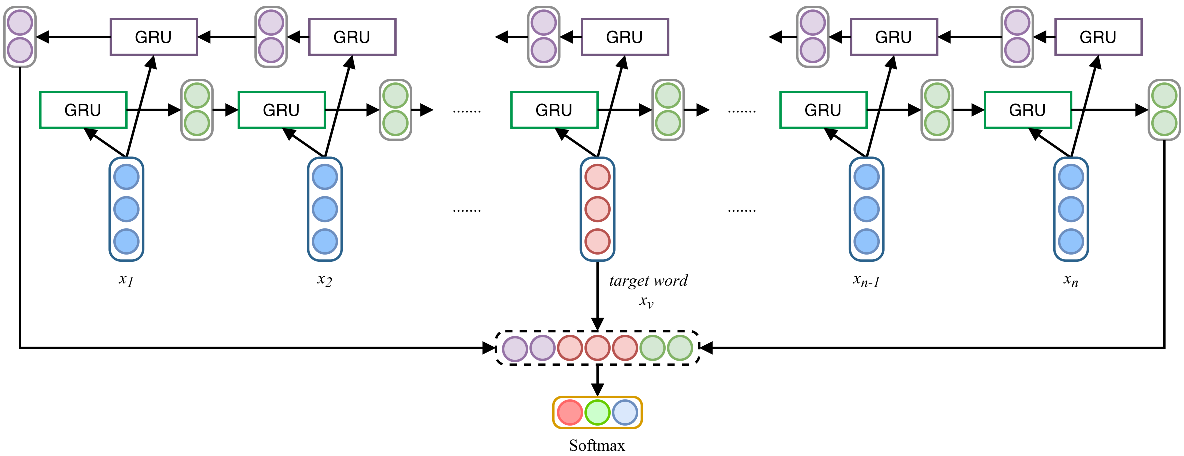


Figure 1: TD-biGRU model for target-dependent sentiment classification.

### 3.2 Sentence-target Representation using TD-biGRU

A recurrent neural network has the ability to represent sequences, e.g. sentences. However, in practice learning long-term dependencies with a vanilla RNN is difficult due to vanishing/exploding gradients (Bengio et al., 1994). *Gated recurrent units* (Cho et al., 2014) were designed to have more persistent memory, making them very useful to capture long-term dependencies between the elements of a sequence.

Gated recurrent units are the basic components of our model. Figure 2 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 if it finds that $h_{t-1}$ is irrelevant to the computation of the new memory, whereas the later is responsible for determining how much of $h_{t-1}$ should be carried forward to the next state.

We describe in this section how we extended this baseline model to represent the syntactic and semantic information of the sentence and the interaction between the sentence and the target.

Let $x_1, x_2, \ldots x_v, \ldots x_n$ be the sequence of word vectors of the sentence obtained in the previous step, where $n$ is the length of the sentence and $x_v$ 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 ”screen resolution”, its representation is the average of the embedding vectors of the words (Sun et al., 2015).

We use two 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 $h_1, h_2, \ldots h_n$ as follows:

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

$$z_t = \sigma(W_z \cdot [h_{t-1}; x_t] + b_z) \tag{2}$$

$$\widetilde{h}_t = tanh(W_h \cdot [(r_t \odot h_{t-1}); x_t] + b_h) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \tag{4}$$

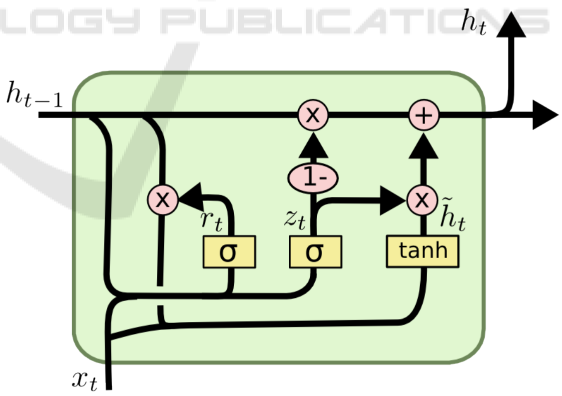


Figure 2: Gated Recurrent Unit (GRU)[1].

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

[1]Figure source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

$d_h$ 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 sentence-target input is represented by the concatenation of the vectors $h_n^f, h_n^b$ and $x_v$, formally:

$$X = [h_n^f; x_v; h_n^b] \tag{5}$$

### 3.3 Softmax Classifier

The sentence-target vector representation $X \in \mathbb{R}^{(2d_h+d)}$ is passed through a *softmax* layer, which computes the probability of classifying the sentence as positive, neutral or negative (thus, this output layer has size three). The softmax function is calculated as follows:

$$P(y=i|X) = \frac{exp(w_i^T X + b_i)}{\sum_{j=1}^{C} exp(w_j^T X + b_j)}, i = 1, ...C \tag{6}$$

In this formula $C$ is the number of classes, and $W \in \mathbb{R}^{(2d_h+d)\times C}$ and $B \in \mathbb{R}^C$ are the parameters of the softmax layer (i.e. the weight matrix and the bias), with $w_i \in W$ and $b_i \in B$ .

### 3.4 Model Training

The objective function is the cross-entropy error.

$$J = -\frac{1}{|S|} \sum_{s \in S} \sum_{c=1}^{C} G_c(s) log(P(y=c|s)) \tag{7}$$

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 $\theta = [W_r, b_r, W_z, b_z, W_h, b_h, W, b]$ , and the parameters are updated with 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.

## 4 EXPERIMENTS AND RESULTS

### 4.1 Dataset

We evaluated the effectiveness of our model by using it in the supervised task of target-dependent sentiment classification on the benchmark dataset provided in (Dong et al., 2014). The dataset contains 6248 training examples and 692 examples in the testing set. Each example in the dataset contains the sentence, the target and the label of sentiment polarity. The numerical description of the positive, negative and neutral examples is shown in table 1.

Table 1: Numerical description of the dataset.

| | Training | Testing | Percentage |
|---|---|---|---|
| #Positives | 1562 | 173 | 25% |
| #Neutrals | 3124 | 346 | 50% |
| #Negatives | 1562 | 173 | 25% |
| Total | 6248 | 692 | |

### 4.2 Parameter Settings

Table 2 lists the values of the hyper-parameters of the model. All the values have been experimentally tuned using 5-fold cross-validation on the training set.

Table 2: Values of the hyper-parameters used in our model.

| Parameter name | Symbol | Value |
|---|---|---|
| *Model hyper-parameters* | | |
| Lookup pre-trained model | | Glove |
| Embedding vector dimension | $d$ | 100 |
| Hidden state dimension | $d_h$ | 64 |
| Number of classes | $C$ | 3 |
| *Training hyper-parameters* | | |
| Learning rate | | 0.1 |
| Learning rate decay | | 0.0001 |
| Momentum | | 0.9 |
| Dropout probability | | 0.5 |

### 4.3 Comparison with other Methods

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

- **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).
- **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 evaluation metrics were 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).

### 4.4 Results and Discussions

We evaluated the effectiveness of our system by comparing it with the state-of-the-art models mentioned above. The values under the section "A" in Table 3 represent the results of the baseline model (basic bi-directional gated recurrent units - biGRU - without incorporating target information) and the new TD-biGRU model. Section "B" contains the results of the compared models (obtained from their associated papers). With the exception of AdaRNN, each approach presented in Table 3 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.

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.

Table 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** |
| B. State-of-the-art systems | | |
| SVM-indep | 62.70 | 60.20 |
| SVM-dep | 63.40 | 63.30 |
| 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 |

The novel TD-biGRU model outperforms the state-of-the-art models both in terms of accuracy and Macro-F1. To get more insight on this result, we analyzed the confusion matrix to figure out which are the most common incorrect cases. Figure 3 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 leave the study of this hypothesis for the future work.

## 5 CONCLUSION

We have developed a novel target-dependent Twitter sentiment analysis system called TD-biGRU. The proposed model has the ability of representing the relatedness between the targets and its contexts. The effectiveness of the proposed model has been evalu-

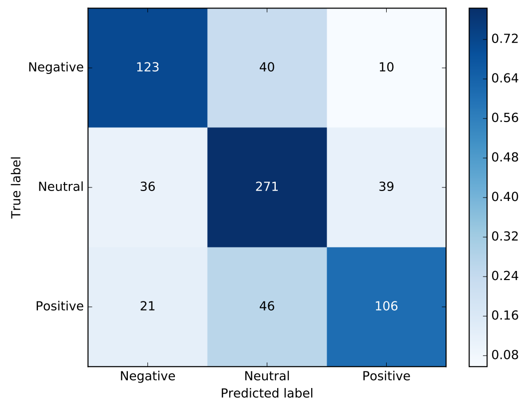


Figure 3: Confusion Matrix.

ated on a benchmark of tweets, obtaining results that outperform the state-of-the-art models. We conducted some experiments to compare LSTM with GRU, and we found that the results are similar. We finally decided to use GRU because it has a number of parameters lower than LSTM. 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 model to handle this weakness by integrating more information such as lexicon information and/or the dependency tree.

## ACKNOWLEDGEMENTS

The authors acknolwedge the support of Univ. Rovira i Virgili through a Martí i Franqués PhD grant and the Research Support Funds 2015PFR-URV-B2-60.

## REFERENCES

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F. (2015). Keras. https://github.com/fchollet/keras.

Deriu, J., Gonzenbach, M., Uzdilli, F., Lucchi, A., De Luca, V., and Jaggi, M. (2016). Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. *Proceedings of SemEval*, pages 1124–1128.

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., and Xu, K. (2014). Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL (2)*, pages 49–54.

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Jabreel, M. and Moreno, A. (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*, pages 137–146.

Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

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

Liu, B. (2011). Opinion mining and sentiment analysis. In *Web Data Mining*, pages 459–526. Springer.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Liu, P., Joty, S., and Meng, H. (2015). Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mohammad, S., Kiritchenko, S., and Zhu, X. (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.

Pang, B. and Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (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, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Severyn, A. and Moschitti, A. (2015). UNITN: Training deep convolutional neural network for Twitter sentiment classification. In *Proceedings of the 9th In-*

*ternational Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pages 464–469.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., and Wang, X. (2015). Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1333–1339. AAAI Press.

Tang, D., Qin, B., Feng, X., and Liu, T. (2015). Target-dependent sentiment classification with long short term memory. *arXiv preprint arXiv:1512.01100*.

Tang, D., Wei, F., Qin, B., Liu, T., and Zhou, M. (2014a). Coooolll: A Deep Learning System for Twitter Sentiment Classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (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)*, pages 1555–1565, Baltimore, Maryland. Association for Computational Linguistics.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Vo, D.-T. and Zhang, Y. (2015). Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1347–1353.

Zhang, M., Zhang, Y., and Vo, D.-T. (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*.