TUGAS: Exploiting Unlabelled Data for Twitter Sentiment Analysis

Silvio Amir[+], Miguel Almeida[*†], Bruno Martins[+], João Filgueiras[+], and Mário J. Silva[+]
[+]INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
[*]Priberam Labs, Alameda D. Afonso Henriques, 41, 2º, 1000-123 Lisboa, Portugal
[†]Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Portugal
samir@inesc-id.pt, miguel.almeida@priberam.pt, bruno.g.martins@tecnico.ulisboa.pt
jfilgueiras@inesc-id.pt, mjs@inesc-id.pt

Abstract

This paper describes our participation in the message polarity classification task of SemEval 2014. We focused on exploiting unlabeled data to improve accuracy, combining features leveraging word representations with other, more common features, based on word tokens or lexicons. We analyse the contribution of the different features, concluding that unlabeled data yields significant improvements.

1 Introduction

Research in exploiting social media for measuring public opinion, evaluating popularity of products and brands, anticipating stock-market trends, or predicting elections showed promising results (O'Connor et al., 2010; Mitchell et al., 2013). However, this type of content poses a particularly challenging problem for text analysis systems. Typical messages show heavy use of Internet slang, emoticons and other abbreviations and discourse conventions. The lexical variation introduced by this creative use of language, together with the unconventional spelling and occasional typos, leads to very large vocabularies. On the other hand, messages are very short, and therefore word feature representations tend to become very sparse, degrading the performance of machine learned classifiers.

The growing interest in this problem motivated the creation of a shared task for Twitter Sentiment Analysis in the 2013 edition of SemEval. The **Message Polarity Classification** task was formalized as follows: *Given a message, decide whether the message is of positive, negative, or neutral sentiment. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen* (Nakov et al., 2013).

	Positive	Neutral	Negative
Train 2014	3230	4109	1265
Tweets 2013	1572	1640	601
Tweets 2014	982	669	202
SMS 2013	492	1207	394
Tweets Sarcasm 2014	33	13	40
LiveJournal 2014	427	411	304

Table 1: Number of examples per class in each SemEval dataset. The first row represents all training data; the other rows are sets used for testing.

We describe our participation on the 2014 edition of this task, for which a set of manually labelled messages was created. Complying with the Twitter policies for data access, the corpus was distributed as a list of message IDs and each participant was responsible for downloading the actual tweets. Using the provided script, we collected a training set with 8604 tweets. After submission, the 2014 test sets were also made available. Along with the Tweets 2014 test set, evaluation was also performed on a set of tweets with sarcasm, on a set of LiveJournal blog entries, and on sets of tweets and SMS messages from the 2013 edition of the task. Table 1 shows the class distribution for each of these datasets.

In the 2013 edition (task 2B), the NRC-Canada system (Mohammad et al., 2013) earned first place by scoring 69.02% on the Official SemEval metric (see Section 4) with a significant margin with respect to the other systems: the second (Günther and Furrer, 2013) and third (Reckman et al., 2013) best systems scored 65.27% and 64.86%, respectively. The main novelty in the NRC-Canada system was the use of sentiment lexicons, specific for the Twitter domain, generated from unlabeled tweets using emoticons and hashtags as indicators of sentiment. They found that these lexicons had a strong impact on the results – more than word and

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 673–677,
Dublin, Ireland, August 23-24, 2014.

character n-grams.

The automatically induced lexicons are a way to use information from unlabeled data to aid in the classification task. In our approach, we take this reasoning further, and focus on the impact of various ways to incorporate knowledge from unlabeled data. This allows us to mimic many real-world scenarios where labelled data is scarce but unlabeled data is plentiful.

2 Word Representations

In text classification it is common to represent documents as bags-of-words, i.e., as unordered collections of words. However, in the case of very short social media texts, these representations become less effective, as they lead to increased data sparseness. We focused our experiments in comparing and complementing these approaches with denser representations, which we now describe.

2.1 Bag-Of-Words and ΔBM25

In a representation based on bags-of-words, each message is represented as a vector $\mathbf{m} = \{w_1, w_2, ..., w_n\} \in \mathbb{R}^V$, where V is the size of the vocabulary. In order to have weights that reflect how relevant a word is to each of the classes, we weighted the individual terms according to the ΔBM25 heuristic (Paltoglou and Thelwall, 2010):

$$\Delta\mathrm{BM25}(\mathrm{w_i}) = tf_i \times \log\left(\frac{(N_p - df_{i,p} + s)\cdot df_{i,n} + s}{(N_n - df_{i,n} + s)\cdot df_{i,p} + s}\right), \quad (1)$$

where tf_i represents the frequency of term i in the message, N_a is the size of corpus a, $df_{i,a}$ is the document frequency of term i in the corpus a (i.e., in one of two subsets for the training data, corresponding to either positive or negative messages), and s is a smoothing constant, which we set to 0.5. This term weighting function was previously shown to be effective for sentiment analysis.

2.2 Brown Clusters

Brown et al. (1992) proposed a greedy agglomerative hierarchical clustering procedure that groups words to maximize the mutual information of bigrams. Clusters are initialized as consisting of a single word each, and are then greedily merged according to a mutual information criterion, to form a lower-dimensional representation of a vocabulary. The hierarchical nature of the clustering allows words to be represented at different levels in the hierarchy. This approach provides a denser representation of the messages, mitigating the feature sparseness problem. We used a publicly available[1] set of 1000 Brown clusters induced from a corpus of 56 million Twitter messages.

We leveraged the word clusters by mapping each word to the corresponding cluster, and we then represented each message as a bag-of-clusters vector in $\mathbb{R}^K$, where $K = 1000$ is the number of clusters. These word cluster features were also weighted with the ΔBM25 scheme.

2.3 Concise Semantic Analysis

Concise Semantic Analysis is a form of term and document representation that assigns, to each term, its weight on each of the classes (Li et al., 2011). These weights, computed from the frequencies of the term on the training data, reflect how associated the term is to each class. The weight of term j in class c is given by (Lopez-Monroy et al., 2013):

$$w_{cj} = \sum_{k \in P_c} \log_2\left(1 + \frac{tf_{kj}}{\mathrm{len}(k)}\right), \quad (2)$$

where P_c is the set of documents with label c and tf_{kj} is the term frequency of term j in document k. To prevent labels with a higher number of examples, or terms with higher frequencies, to have stronger weights, an additional normalization step is performed to obtain nw_{cj}, the normalized weight of term j in class c:

$$nw_{cj} = \frac{w_{cj}}{\sum_{l \in L} w_{lj} \times \sum_{t \in T} w_{ct}}. \quad (3)$$

In the formula, L is the set of class labels and T is the set of terms, making w_{lj} the weight of term j for a class l, and w_{ct} the weight of a term t in class c. After defining every term as a vector $\mathbf{t}_j = \{nw_{1j}, \ldots, nw_{Cj}\} \in \mathbb{R}^C$, where C is the number of classes, each message m is represented by summing each of its terms' weight vectors:

$$\mathbf{m}_{csa} = \sum_{j \in m} \frac{tf_j}{\mathrm{len}(m)} \times \mathbf{t}_j. \quad (4)$$

In the formula, tf_j is the frequency of term j in m.

2.4 Dense Word Vectors

Efficient approaches have recently been introduced to train neural networks capable of producing continuous representations of words (Mikolov

[1] http://www.ark.cs.cmu.edu/TweetNLP/

Lexicon	#1-grams	#2-grams	#pairs
Bing Liu	6789	-	-
MPQA	8222	-	-
SentiStrength	2546	-	-
NRC EmoLex	14177	-	-
Sentiment140	62468	677698	480010
NRC HashSent	54129	316531	308808

Table 2: Number of unigrams, bigrams, and collocation pairs, in the lexicons used in our system.

et al., 2013). These approaches allow fast training of projections from a representation based on bags-of-words, where vectors have very high dimension (of the order of 10^4), but are also very sparse and integer-valued, to vectors of much lower dimensions (of the order of 10^2), with full density and continuous values.

To induce word embeddings, a corpus of 17 million Twitter messages was collected with the Twitter crawler of Boanjak et al. (2012). Then, using *word2vec*[2], we induced representations for the word tokens occurring in the messages. All the tokens were represented as vectors $\mathbf{w}_j \in \mathbb{R}^n$, with $n = 100$. A message was modeled as the sum of the vector representations of the individual words:

$$\mathbf{m}_{vec} = \sum_{j \in m} \mathbf{w}_j. \quad (5)$$

We also created a *polarity class vector* $\mathbf{p_c}$ for each class c, defined as:

$$\mathbf{p}_c = \frac{1}{N_c} \sum_{m \in c} \mathbf{m}_{vec}, \quad (6)$$

where m is a message of class c and N_c is the total number of instances in class c. These vectors can be interpreted as prototypes of their classes, and are used in the **classVec** features described below.

3 The TUGAS System

We now describe the TUGAS approach, detailing the considered features and our modeling choices.

3.1 Word Features

To reduce the feature space of the model, messages were lower-cased, Twitter user mentions (*@username*) were replaced with the token <USER> and URLs were replaced with the <URL> token. We also normalized words to include at most 3 repeated characters (*e.g.*, *"hellooooooo!"* to *"hellooo!"*). Following Pang et al. (2002), negation was directly integrated into the word representations. All the tokens occurring between a negation word and the next punctuation mark, were suffixed with the _**NEG** annotation.

We used the following groups of features:

- **bow-uni**: vector of word unigrams
- **bow-bc**: vector of Brown word clusters
- **csa**: Concise Semantic Analysis vector $\mathbf{m}_{csa}$
- **wordVec**: *word2vec* message vector $\mathbf{m}_{vec}$
- **classVec**: Euclidean distance between message vector $\mathbf{m}_{vec}$ and each class vector $\mathbf{p}_c$

3.2 Lexicon Features

The document model was enriched with features that take into account the presence of words with a known prior polarity, such as *happy* or *sad*. We included words from manually annotated sentiment lexicons: Bing Liu Opinion Lexicon (Hu and Liu, 2004), MPQA (Wilson et al., 2005) and the NRC Emotion Lexicon (Mohammad and Turney, 2013). We also used the two automatically generated lexicons from Mohammad et al. (2013): the NRC Hashtag Sentiment Lexicon and the Sentiment140 Lexicon. Table 2 summarizes the number of terms of each lexicon.

As Mohammad et al. (2013), we added the following set of **lexicon** features, for each lexicon, and for each combination of negated/non-negated words and positive/negative polarity.

- The sum of the sentiment scores of all (negated/non-negated) terms with (positive/negative) sentiment
- The largest of those scores
- The sentiment score of the last word in the message that is also present in the lexicon
- The number of terms within the lexicon

Notice that terms can be unigrams, bigrams, and collocations pairs. A group of these features was computed for each of the sentiment lexicons.

3.3 Syntactic Features

We extracted **syntactic** features aimed at the Twitter domain, such as the use of heavy punctuation, emoticons and character repetition. Concretely, the following features were computed from the original Twitter messages:

- Number of words originally with more than 3 repeated characters
- Number of sequences of exclamation marks and/or question marks

[2] https://code.google.com/p/word2vec/

Features	Tweets Test 2013			Tweets Test 2014			SMS 2013			Live Journal 2014			Tweets Sarcasm 2014		
	Acc	F1	Official	Acc	F1	Official	Acc	F1	Official	Acc	F1	Official	Acc	F1	Official
bow-uni	65.62	59.30	54.60	69.94	66.30	**65.60**	68.80	62.40	54.90	60.42	58.30	56.60	47.67	43.90	41.50
submitted	**69.55**	**67.50**	**65.60**	**71.45**	**69.00**	**69.00**	**70.57**	**67.60**	**62.70**	**68.21**	**68.20**	**69.80**	**53.49**	**50.10**	**52.90**
- lexicons	66.90	64.30	61.70	70.37	67.00	**66.40**	66.46	63.50	58.30	64.27	64.20	65.50	48.84	45.10	47.00
- classVec	69.37	67.30	65.40	71.83	69.30	**69.60**	69.14	66.60	62.10	67.51	67.50	69.30	53.49	50.10	52.90
- wordVec	69.63	67.70	66.00	70.32	67.70	**68.00**	66.79	64.90	60.90	68.04	68.00	69.70	53.49	50.50	53.50
- bow-bc	68.06	66.40	65.10	67.40	64.30	**65.30**	67.89	65.20	60.40	68.30	68.30	70.00	52.33	49.90	49.90
+ syntactic	69.58	67.60	65.70	71.24	68.30	**68.50**	70.38	67.40	62.40	67.95	68.00	69.70	52.33	48.80	50.00
+ csa	67.45	63.70	60.50	70.10	67.30	**67.50**	71.48	67.60	62.10	66.11	66.00	68.30	53.49	51.30	50.30
+ bow-uni	67.69	62.50	58.50	70.64	67.30	**66.70**	72.77	67.10	60.40	67.60	67.20	67.10	51.16	48.00	43.90

Table 3: Impact of removing or adding groups of features. The row marked as submitted, in bold, is the one that we submitted to the shared task. The bold column is the official score used to rank participants.

- Number of positive/negative emoticons, detected with a pre-existing regular expression[3]
- Number of capitalized words

3.4 Model Training

We used the L2-regularized logistic regression implementation from *scikit-learn*[4]. Given a set of m instance-label pairs $(\mathbf{x_i}, y_i)$, with $i = 1, \ldots, m$, $\mathbf{x_i} \in \mathbb{R}^n$, and $y_i \in \{-1, +1\}$, learning the classifier involves solving the following optimization problem, where $C > 0$ is a penalty parameter.

$$\min_w \frac{1}{2}\mathbf{w}'\mathbf{w} + C\sum_{i=1}^{m}\log(1 + e^{-y_i w'\mathbf{x_i}}). \quad (7)$$

In *scikit-learn*, the problem is solved through a trust region Newton method, using a wrapper over the implementation available in the *liblinear*[5] package. For multi-class problems, *scikit-learn* uses the one-vs-the-rest strategy. This particular implementation also suports the introduction of class weights, which we set to be inversely proportional to the class frequency in the training data, thus making each class equally important.

The selection of groups of features to be included in the submitted run, as well as the tuning of the regularization constant, were obtained by cross-validation on the training dataset.

4 Results

We report results using the following metrics:

- **Accuracy**, defined as the percentage of tweets correctly classified.
- Overall **F1**, computed by averaging the F1 score of all three classes.
- The **Official** SemEval score, computed by averaging the F1 scores of the positive and negative classes (Nakov et al., 2013).

[3] http://sentiment.christopherpotts.net/

[4] http://scikit-learn.org/

[5] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

Feature group	Acc	F1	Official
bow-bc	66.33	63.30	60.30
wordVec	62.34	60.00	57.90
bow-uni	65.62	59.30	54.60
csa	61.58	56.70	52.90

Table 4: Performance comparison using different word representations in isolation.

We tried including or excluding various groups of features, and obtained the best results on the training set using Brown clusters (**bow-bc**), lexicon features (**lexicon**), word2vec word representations (**wordVec**), and the Euclidean distance between the word2vec representation and each class vector (**classVec**). These features were the ones used in our submission. Inclusion of syntactic features (**syntactic**), Concise Semantic Analysis (**csa**), and word unigrams (**bow-uni**) was found to decrease performance during cross-validation, and thus these features were not included.

Table 4 shows the results on the Twitter 2014 test set using only a single group of word representation features to train the model, from each of the techniques introduced in Section 2. This table suggests that exploiting unlabeled data is beneficial, as representing words through their Brown clusters (**bow-bc**) or through word2vec (**wordVec**) yields better results than unigrams or CSA.

Table 3 shows results on five different test sets, including two from the 2013 challenge (Nakov et al., 2013), when features are added or removed from the official submission, one group at a time. Adding representations like **bow-uni** or **csa** actually *hurts* the performance, suggesting that, given the relatively small set of training instances, using coarse-level features in isolation, such as Brown clusters, can yield better results.

More importantly, we verify that lexicon-based and Brown cluster features have the largest impact

(2.6% and 3.7%, respectively, in the official metric). These results indicate that leveraging unlabeled data yields significant improvements.

5 Conclusions

This paper describes the participation of the TUGAS team in the message polarity classification task of SemEval 2014. We showed that there are significant gains in leveraging unlabeled data for the task of classifying the sentiment of Twitter texts. Our score of 69% ranks at fifth place in 42 submissions, roughly 2% points below the top score of 70.96%. We believe that the direction of leveraging unlabeled data is still vastly unexplored and, for future work, we intend to: (a) experiment with semi-supervised learning approaches, further exploiting unlabeled tweets; and (b) make use of domain adaptation strategies to leverage on labelled non-Twitter data.

Acknowledgements

This work was partially supported by the EU/FEDER programme, QREN/POR Lisboa (Portugal), under the Intelligo project (contract 2012/24803). The researchers from INESC-ID were supported by Fundação para a Ciência e Tecnologia (FCT), through contracts Pest-OE/EEI/LA0021/2013, EXCL/EEI-ESS/0257/2012 (DataStorm), project grants PTDC/CPJ-CPO/116888/2010 (POPSTAR), and EXPL/EEI-ESS/0427/2013 (KD-LBSN), and Ph.D. scholarship SFRH/BD/89020/2012.

References

Matko Boanjak, Eduardo Oliveira, José Martins, Eduarda Mendes Rodrigues, and Luís Sarmento. 2012. Twitterecho: a distributed focused crawler to support open research with twitter data. In *21st International Conference Companion on World Wide Web*, pages 1233–1240.

Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Tobias Günther and Lenz Furrer. 2013. GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *7th International Workshop on Semantic Evaluation*, pages 328–332.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Zhixing Li, Zhongyang Xiong, Yufang Zhang, Chunyong Liu, and Kuan Li. 2011. Fast text categorization using concise semantic analysis. *Pattern Recognition Letters*, 32(3):441–448.

Ádrian Pastor Lopez-Monroy, Manuel Montes-y Gomez, Hugo Jair Escalante, Luis Villasenor-Pineda, and Esaú Villatoro-Tello. 2013. INAOE's participation at PAN'13: Author profiling task. In *CLEF 2013 Evaluation Labs and Workshop*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *27th Annual Conference on Neural Information Processing Systems*, pages 3111–3119.

Lewis Mitchell, Kameron Decker Harris, Morgan R Frank, Peter Sheridan Dodds, and Christopher M Danforth. 2013. The geography of happiness: connecting twitter sentiment and expression, demographics, and objective characteristics of place. *PLoS ONE*, 8(5):e64417.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: building the state-of-the-art in sentiment analysis of tweets. In *7th International Workshop on Semantic Evaluation*, pages 321–327.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *7th International Workshop on Semantic Evaluation*, pages 312–320.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *4th International AAAI Conference on Weblogs and Social Media*, pages 122–129.

Georgios Paltoglou and Mike Thelwall. 2010. A study of information retrieval weighting schemes for sentiment analysis. In *48th Annual Meeting of the Association for Computational Linguistics*, pages 1386–1395.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *ACL-02 Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Hilke Reckman, Baird Cheyanne, Jean Crawford, Richard Crowell, Linnea Micciulla, Saratendu Sethi, and Fruzsina Veress. 2013. teragram: Rule-based detection of sentiment phrases using SAS sentiment analysis. In *7th International Workshop on Semantic Evaluation*, pages 513–519.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354.