INESC-ID: Sentiment Analysis without hand-coded Features or Liguistic Resources using Embedding Subspaces

Ramon F. Astudillo, Silvio Amir, Wang Ling, Bruno Martins[†], Mário Silva, Isabel Trancoso
Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento
Rua Alves Redol 9
Lisbon, Portugal
{ramon.astudillo, samir, wlin, mjs, isabel.trancoso}@inesc-id.pt
[†]bruno.g.martins@tecnico.ulisboa.pt

Abstract

We present the INESC-ID system for the message polarity classification task of SemEval 2015. The proposed system does not make use of any hand-coded features or linguistic resources. It relies on projecting pre-trained structured skip-gram word embeddings into a small subspace. The word embeddings can be obtained from large amounts of Twitter data in unsupervised form. The sentiment analysis supervised training is thus reduced to finding the optimal projection which can be carried out efficiently despite the little data available. We analyze in detail the proposed approach and show that a competitive system can be attained with only a few configuration parameters.

1 Introduction

Web-based social networks are a rich data source for both businesses and academia. However, the sheer volume, diversity and rate of creation of social media, imposes the need for automated analysis tools. The growing interest in this problem motivated the creation of a shared task for Twitter Sentiment Analysis (Nakov et al., 2013). The Message Polarity Classification task consists in classifying a message as positive, negative, or neutral in sentiment.

A great deal of research has been done on methods for sentiment analysis on user generated content. However, state-of-the-art systems still largely depend on linguistic resources, extensive feature engineering and tuning. Indeed, if we look at the best performing systems from SemEval 2014 (Zhu et al., 2014), (Malandrakis et al., 2014), both make extensive use of these resources, including hundreds of thousands of features, special treatment for negation, multi-word expressions or special strings like emoticons.

In this paper we present the INESC-ID system for the 2015 SemEval message polarity classification task (Rosenthal et al., 2015). The system is able to learn good message representations for message polarity classification directly from raw text with a simple tokenization scheme. Our approach is based on using large amounts of unlabeled data to induce *word embeddings*, that is, continuous word representations containing contextual information. Instead of using these word embeddings directly with, for instance, a logistic regression classifier, we estimate a *sentiment subspace* of the embeddings. The idea is to find a projection of the embedding space that is meaningful for the supervised task. In the proposed model, we jointly learn the sentiment subspace projection and the classifier using the SemEval training data. The resulting system attains state-of-the-art performance without hand-coded features or linguistic resources and only a few configuration parameters.

2 Unsupervised Learning of Word Embeddings

Unsupervised word embeddings trained from large amounts of unlabeled data have been shown to improve many NLP tasks (Turian et al., 2010; Collobert et al., 2011). Embeddings capture generic regularities about the data and can be trained with virtually an infinite amount of data in unsupervised

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 652–656,
Denver, Colorado, June 4-5, 2015.

fashion. Once trained, they can be used as features for supervised tasks or to initialize more complex models (Collobert et al., 2011; Chen and Manning, 2014; Bansal et al., 2014). Other unsupervised approaches that can also be used for feature extraction include brown clustering (Brown et al., 1992) and LDA (Blei et al., 2003),

One popular objective function for embeddings is to maximize the prediction of contextual words. In the work described in (Mikolov et al., 2013), commonly referred as word2vec, the models defined estimate the optimal word embeddings by maximizing the probability that the words within a given window size are predicted correctly. In the work presented here, a structured skip-gram (Ling et al., 2015) was used to generate the embeddings. Central to the skip-gram (Mikolov et al., 2013) is a log linear model of word prediction. Let $w = i$ denote that a word at a given position of a sentence is the i-th word on a vocabulary of size v. Let $w^p = j$ denote that the word p positions further in the sentence is the j-th word on the vocabulary. The skip-gram models the following probability:

$$p(w^p = j | w = i; \mathbf{C}, \mathbf{E}) \propto \exp\left(\mathbf{C}_j \cdot \mathbf{E} \cdot \mathbf{w}^i\right). \quad (1)$$

Here, $\mathbf{w}^i \in \{1, 0\}^{v \times 1}$ is a one-hot representation of $w = i$. That is, a vector of zeros of the size of the vocabulary v with a 1 on the i-th entry of the vector. The symbol $\cdot$ denotes internal product and $\exp()$ acts element-wise. The log-linear model is parametrized by two matrices. $\mathbf{E} \in \mathbb{R}^{e \times v}$ is the embedding matrix, transforming the one-hot sparse representation into a compact real valued embedding vector of size $e \times 1$. The matrix $\mathbf{C} \in \mathbb{R}^{v \times e}$ maps the embedding to a vector with the size of the vocabulary v. In the particular case of the structured skip-gram, here used, a different prediction matrix is trained for each relative position between words $\mathbf{C}_p$. After exponentiating and normalizing over the v possible options, the j-th element of the resulting vector corresponds thus to the probability of $w^p = j$.

In practice, due to the large value of v, various techniques are used to avoid having to normalize over the whole vocabulary.

After the embeddings are trained, the low dimensional embedding of each word $\mathbf{E} \cdot \mathbf{w}^i \in \mathbb{R}^{e \times 1}$ encapsulates the information about each word and its surrounding contexts. This embedding can thus be used as input to other learning algorithms to further enhance performance.

3 Using Embeddings for Sentiment Prediction

3.1 Sentiment Embedding Subspace

There are multiple ways in which embeddings could be incorporated as a pre-training step into a supervised task. The initial attempts for the proposed system included log-linear classifiers using the embeddings as initialization values or features, but these led to poor results. Ideally, embeddings should be adapted to the supervised task. However, this faces an additional difficulty: only a small subset of the words will actually be present in the training set of the supervised task. Words not present in the supervised training set will never get their embeddings updated.

To avoid this, here we employ a simple projection scheme. We consider the adapted embeddings $\mathbf{S} \cdot \mathbf{E}$, where $\mathbf{E} \in \mathbb{R}^{e \times v}$ is the original unadapted embedding matrix and $\mathbf{S} \in \mathbb{R}^{s \times e}$, with $s \ll e$, is a projection matrix trained on the supervised data. The idea is that, by only training $\mathbf{S}$ on the supervised data, we determine a sub-space of the embeddings which is optimal for the supervised task. An additional advantage is that, unlike with a direct re-estimation of $\mathbf{E}$, all embeddings are updated based on the supervised task data. This simple approach proved very useful and it accounts for most of the performance attained in our system.

3.2 Non-linear Sub-space Model

Based on the sub-space concept, various log-linear and non-linear models were explored. Most of the models attempted were prone to get trapped in poor local minima or showed stability problems during training. The only exception identified is the non-linear model here presented, which showed both fast convergence and high performance.

In what follows, we will denote a message, e.g. a tweet, of n words as a matrix $\mathbf{m} \in \{0, 1\}^{v \times n}$, where each column is a one-hot representation of each word. The vocabulary v is equal to that of the unsupervised pre-training. Words of the SemEval task not appearing in that vocabulary are represented as a vector of zeros, equivalent to an embedding of e zeros. In the SemEval task, each message has to be classified as neutral, negative or positive. Let y denote a categorical random variable over those three classes. The sub-space non-linear model estimates thus the probability of each possible category $y = k$ given a message $\mathbf{m}$ as

$$p(y = k|\mathbf{m}; \mathbf{C},\mathbf{S}) \propto \exp\left(\mathbf{C}_k \cdot \sigma\left(\mathbf{S} \cdot \mathbf{E} \cdot \mathbf{m}\right) \cdot \mathbf{B}\right), \quad (2)$$

where $\sigma()$ is a sigmoid function acting on each element of the matrix. The matrix $\mathbf{C} \in \mathbb{R}^{3\times s}$ maps the embedding sub-space to the classification space and $\mathbf{B} \in 1^{n\times 1}$ is a matrix of ones that sums the scores for all words up prior to normalization. This simplification, equivalent to a bag of words assumption, outperformed other approaches like convolution.

The model is thus equivalent to a multi-layer perceptron (MLP) (Rumelhart et al., 1985) with one hidden sigmoid layer and a soft-max output layer. The input to the MLP would be the fixed word embeddings attained by applying $\mathbf{E}$. The input layer $\mathbf{S}$ learns a projection of $\mathbf{E}$ into a small sub-space of size $s \ll e$.

4 Proposed System

4.1 Unsupervised Word Embeddings Learning

The embedding matrix $\mathbf{E}$ was trained in unsupervised fashion using the structured skip-gram model, described in Section 2.

We used the corpus of 52 million tweets used in (Owoputi et al., 2013) with the tokenizer described in the same work. The words that occurred less than 40 times in the data were discarded from the vocabulary. To train the model, we used a negative sampling rate of 25 words, sampled from a multinomial of unigram word probabilities over all the vocabulary (Goldberg and Levy, 2014). Embeddings of 50, 200, 400 and 600 dimensions were trained.

It should be noted that the training configuration is generic and was not adapted to the SemEval task. One consequence of this is a relatively strong pruning of the vocabulary. Around 23% of words in the SemEval tasks did not have an embedding and thus were set to have an embedding of e zeros.

4.2 Supervised Embedding Sub-space Learning

Text normalization for the supervised task employed the CMU tokenizer plus the following additional steps: messages were lower-cased, Twitter user mentions and URLs were replaced with special tokens and any character repetition above 3 was mapped to 3 characters.

The small amount of supervised data available was the main driving factor behind the design and optimization of the supervised training component. In order to maintain the number of free parameters low, small sizes of the subspace were selected with values ranging from 5 to 30. Training was also kept as simple as possible. The training set of SemEval was split into 80% for parameter learning and 20% for hyper-parameter selection, maintaining the original sentiment relative frequencies in each set. The 2013 and 2014 SemEval sentiment analysis test sets were used to validate the different candidate models. The most probable class was selected as the model prediction.

The parameters of subspace model in Equation 2, $\mathbf{S}$ and $\mathbf{C}$ were estimated to minimize the negative log-likelihood of the correct class. Training employed conventional Stochastic Gradient Descent (Rumelhart et al., 1985) with mini-batch size 1 and random uniform initialization similar to (Glorot and Bengio, 2010). After some initial experiments, it was determined that a learning rate of 0.01 and selecting the model with the best accuracy on the 20% set after 8 iterations led to the best results.

5 Experiments and Results

5.1 Sensibility Analysis

This section analyzes the performance of the proposed system on the message polarity classification task of SemEval 2015. In general, the sentiment subspace model showed consistent and fast convergence towards the optimum in very few iterations. Despite using class log-likelihood as training criterion and accuracy as stopping criterion, the model showed good performance in terms of average F-measure for positive and negative sentiments. This was not always the case for other tested models.

Regarding the two main parameters, embedding size e and sub-space size s, sensibility analysis were carried out and are shown in Tables 1 and 2. For these experiments learning rate and stopping condition were left fixed to the previously indicated values. Variations of learning rate to smaller values e.g. 0.005 were explored but did not lead to a clear pattern.

Table 1 shows the effect of embedding size on the system's performance. Very small embeddings lead clearly to worse results. Larger embeddings not always provide the best performance. However, they provide more consistent results across test sets. It was also inferred from other tasks that using larger embeddings had in general a positive effect.

Emb Size (e)	Dev	2013	2014
50	65.96	68.35	70.54
200	**70.65**	70.28	72.80
400	70.19	71.54	72.24
600	70.08	**72.16**	**72.72**

Table 1: Avg. F-measure on SemEval development and test sets varying with embedding size e. Sub-space size $s = 10$. Best model per column in bold.

Table 2 shows the variation of system performance with sub-space size. The optimal value was consistently found to be at $s = 10$ regardless of embedding size.

Subsp. Size (s)	Dev	2013	2014
5	69.78	71.82	72.17
10	**70.08**	**72.16**	**72.72**
20	69.18	71.97	72.52
30	67.81	70.97	72.45

Table 2: Avg. F-measure on SemEval test sets varying with embedding sub-space size s. Embedding size $e = 600$. Best model per column in bold.

5.2 Submitted System and Revised Candidates

Due to time constraints, not all planned configurations could be tested prior to system submission. Consequently, some of the experiments shown in the previous section were carried out after submission. Based on these results, two new candidates were selected and then tested on the 2015 dataset. These were a system that showed a very stable performance using $e = 600$ and $s = 10$ and a good system with a smaller embedding size using $e = 200$, $s = 10$. The same configuration, learning rate and number of iterations, as in the submitted model were used for these experiments.

The results for the submitted system and the a posteriori selected ones are displayed in Table 3. The results on 2015, confirm the sensibility analysis of e and s. The high performance of the $e = 600$, $s = 10$ model on the 2015 dataset was however unexpected, since it tops the submitted system by more than a 1% absolute. The second model selected, using a smaller e size displayed a performance comparable to that of the submitted system thus showing the overall robustness of the approach.

e	s	Dev	2013	2014	2015
600	20	69.18	71.97	72.52	64.12
600	10	70.08	**72.16**	72.72	**65.19**
200	10	**70.65**	70.28	**72.80**	64.09

Table 3: Avg. F-measure of the submitted system (top) and posteriorly selected candidates (bottom). Best model per column in bold.

It should be noted as well that there is small difference between the result attained in the submitted predictions (64.17) and the ones reported here for the submitted system (64.12). Upon revision of the code we could determine that this was due to a minor bug affecting how the embeddings of the $\mathbf{E}$ matrix were constructed.

6 Conclusions

We have presented the INESC-ID system for the SemEval 2015 message classification task. The system does not make use of any hand-coded features or linguistic resources and employs a simple tokenization scheme. The system is however able to attain state-of-the-art performance with few configuration parameters and a small number of iterations. The results are also consistent across sets and configuration settings.

Acknowledgments

This work has been partially supported by the FCT through national funds with reference UID/CEC/50021/2013 and grant number SFRH/BPD/68428/2010.

References

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

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. *Comput. Linguist.*, 18(4):467–479, December.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Nikolaos Malandrakis, Michael Falcone, Colin Vaz, Jesse Bisogni, Alexandros Potamianos, and Shrikanth Narayanan. 2014. Sail: Sentiment analysis using semantic similarity and contrast. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

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

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '2015, Denver, Colorado, June.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, DTIC Document.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif Mohammad. 2014. NRC-Canada-2014:: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*.