INESC-ID at SemEval-2016 Task 4-A: Reducing the Problem of Out-of-Embedding Words

Silvio Amir, Ramon F. Astudillo, Wang Ling, Mário J. Silva, Isabel Trancoso
Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento
Rua Alves Redol 9
Lisbon, Portugal
{samir,ramon.astudillo,wlin,mjs,isabel.trancoso}@inesc-id.pt

Abstract

We present the INESC-ID system for the 2016 edition of SemEval Twitter Sentiment Analysis shared task (subtask 4-A). The system was based on the Non-Linear Sub-space Embedding (NLSE) model developed for last year's competition. This model trains a projection of pre-trained embeddings into a small subspace using the supervised data available. Despite its simplicity, the system attained performances comparable to the best systems of last edition with no need for feature engineering. One limitation of this model was the assumption that a pre-trained embedding was available for every word. In this paper, we investigated different strategies to overcome this limitation by exploiting character-level embeddings and learning representations for out-of-embedding vocabulary words. The resulting approach outperforms our previous model by a relatively small margin, while still attaining strong results and a consistent good performance across all the evaluation datasets.

1 Introduction

Pre-trained word embeddings provide a simple means to attain semi-supervised learning in Natural Language Processing (NLP) tasks (Collobert et al., 2011). They can be trained with large amounts of unsupervised data and be fine tuned as the initial building block of a semi-supervised system. However, in domains with a significant number of typos, use of slang and abbreviations, such as social media, the high number of singletons leads to a poor fine tuning of the embeddings. In previous work, we addressed this by learning a projection of the embeddings into a small sub-space (Astudillo et al., 2015b). This allowed us to attain representations also for Out-Of-Vocabulary (OOV) words, provided that embeddings for those words are available. However, even if the embeddings are estimated from large amounts of unlabeled text, in noisy domains, such as Twitter, a significant number of words will not be seen and therefore will not have an embedding. We refer to those words as the Out-of-Embedding Vocabulary (OOEV).

In this paper, we focus on the problem of obtaining good representations for OOEV words. We experimented with character to word models (C2W) and investigated different strategies for initializing and updating OOEVs from the available training data. The best results were attained by using the labeled data to perform small updates to these representations in the first few epochs of training. The resulting system outperforms that of the previous evaluation, although by a small margin. It ranks fourth in the 2016 evaluation with a consistently high performance in all years.

2 NLSE Model Overview

In this section, we briefly review the approach introduced in the 2015 evaluation (Astudillo et al., 2015a). For a particular regression or classification task, only a subset of all the latent aspects captured by the word embeddings will be useful. Therefore, instead of updating the embeddings directly with the available labeled data, we estimate a projection of these embeddings into a low dimensional sub-space. This simple method brings two fundamental advan-

Proceedings of SemEval-2016, pages 238–242,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

tages. Firstly, the lower dimensional embeddings require fewer parameters fitting the complexity of the target task and the available training data. Secondly, the learned projection can be used to adapt the representations for all words with an embedding, even if they do not occur in the labeled dataset.

Assuming we are given a matrix of pre-trained embeddings, where each column represents a word from a vocabulary $\mathcal{V}$, let such matrix be denoted by $\mathbf{E} \in \mathbb{R}^{e \times |\mathcal{V}|}$, where e is the number of latent dimensions. We define the adapted embedding matrix as the factorization $\mathbf{S} \cdot \mathbf{E}$, where $\mathbf{S} \in \mathbb{R}^{s \times e}$, with $s \ll e$. The parameters of matrix $\mathbf{S}$ are estimated using the labeled dataset, while $\mathbf{E}$ is kept fixed. In other words, we determine the optimal projection of the embedding matrix $\mathbf{E}$ into a sub-space of dimension s. In what follows, we will refer to this approach as Non-Linear Sub-space Embedding (NLSE) model.

The NLSE can be interpreted as a simple feed-forward neural network model (Rumelhart et al., 1985) with one single hidden layer utilizing the embedding sub-space approach. Let

$$\mathbf{m} = [\mathbf{w}_1 \cdots \mathbf{w}_n] \tag{1}$$

denote a message of n words. Each column $\mathbf{w} \in \{0,1\}^{v \times 1}$ of $\mathbf{m}$ represents a word in one-hot form, that is, a vector of zeros of the size of the vocabulary v with a 1 on the i-th entry of the vector. Let y denote a categorical random variable over K classes. The NLSE model, estimates the probability of each possible category $y = k \in K$ given a message $\mathbf{m}$ as

$$p(y = k|\mathbf{m}; \theta) \propto \exp\left(\mathbf{Y}_k \cdot \mathbf{h} \cdot \mathbf{1}\right) \tag{2}$$

with parameters $\theta = \{\mathbf{S}, \mathbf{Y}\}$. Here, $\mathbf{h} \in [0,1]^{e \times n}$ are the activations of the hidden layer for each word, given by

$$\mathbf{h} = \sigma\left(\mathbf{S} \cdot \mathbf{E} \cdot \mathbf{m}\right) \tag{3}$$

where $\sigma()$ is a sigmoid function acting on each element of the matrix. The matrix $\mathbf{Y} \in \mathbb{R}^{3 \times s}$ maps the embedding sub-space to the classification space and $\mathbf{1} \in 1^{n \times 1}$ is a matrix of ones that sums the scores for all words together, prior to normalization. This is equivalent to a bag-of-words assumption. Finally, the model computes a probability distribution over the K classes, using the *softmax* function.

3 Out-of-Embedding Vocabulary Words

Despite the fact that word embeddings are typically estimated from very large amounts of unlabeled data, it is often the case that a number of words appearing on the training or test sets are not present on the unlabeled corpus. These words will not be represented in $\mathbf{E}$. This problem is even more significant in social media environments like Twitter, where there is a significant lexical variation and where novel words, expressions and slang can be introduced over time. In Table 1, we show the percentage of OOV and OOEV words on each Twitter dataset.

The näive way of dealing with this issue, is to simply set the embeddings of unknown words to zero, essentially ignoring them. As will see later, a better approach is to treat these words as model parameters and use the training signal to learn a better representation for them.

3.1 Character-level Embeddings

One natural way of avoiding OOEV in neural network models, is to learn character-level embeddings and define a composition function to combine them into word representations, thus obtaining representations for any given word.

We experimented using *C2W*, a simple compositional model for learning word representations, from character embeddings. Given a word w, the *C2W* model generates a set of character n-grams $\{c_1, \ldots, c_m\}$, and projects each n-gram c_i into a vector $\mathbf{e}_{c_i} \in \mathbb{R}^d$, where d is the number of latent dimensions. The individual character representations are then combined to obtain a fixed-size representation for word w as $\mathbf{e}_w = \mathbf{e}_{c_1} \oplus \ldots \oplus \mathbf{e}_{c_m}$, where $\oplus$ denotes pointwise sum. These word representations can be used as the input to a standard neural language model where the parameters are estimated from unlabeled data by learning to predict words within a context.

3.2 Mapping *C2W* to *SSG* Embeddings

Unfortunately the *C2W* embeddings performed very poorly in our model. Therefore, to have embeddings for all the words we employed an approach similar to (Mikolov et al., 2013). We learn a mapping between the embedding spaces induced by *C2W* and

	2013	2014	2015	2016
OOV	70.9%	37.9%	39.3%	65.1%
OOEV	15.0%	11.2%	11.5%	22%
OOV & OOEV	14.8%	11.0%	11.3%	21.8%

Table 1: Out Of Vocabulary (OOV) and Ouf Of Embedding Vocabulary (OOEV) statistics for the different SemEval Task4-B datasets. Embeddings reported are the Structured Skipgram embeddings used in the experiments.

System	2013	2014	2015
2015 hyperparameters	0.618	0.646	0.591
+lower neutral cost	0.706	0.702	0.669
+shuffle per epoch	0.723	0.721	0.649
+update OOEVs 2 iter	0.725	0.729	0.657
Best SemEval 2015	0.722	0.727	0.652

Table 2: Effect of the improvements on the NLSE model.

Structured Skip-Gram embeddings (*SSG*) (Ling et al., 2015), allowing us compute an approximate *SSG* embedding for all the words. To this end, we first obtained $\mathcal{C}$, the set of words present in the two embeddings spaces. Then, we learned a linear map $\mathbf{T}$ by solving for the following objective:

$$\mathbf{T} \leftarrow \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{w \in \mathcal{C}} ||\mathbf{T} \cdot \mathbf{c}_w - \mathbf{s}_w||^2 \qquad (4)$$

where, $\mathbf{c}_w$ denotes the *C2W* embedding for word w and $\mathbf{s}_w$ denotes the *SSG* embedding for word w. This mapping, was then used to compute a *SSG* embeddings for each OOEV as $\mathbf{s}_{w'} = \mathbf{T} \cdot \mathbf{c}_{w'}$.

3.3 Partial Update of Embeddings during Training

Given the small amount of supervised data, directly updating the embeddings with the SemEval train set leads to very poor results. It is however possible to update only the OOEV words present in the training set simultaneously to the computation of the subspace (Astudillo et al., 2015a). To obtain positive results with this approach, it was also necessary to reduce the effect of training by lowering the learning rate to 0.001 and updating the embeddings only in the first two iterations.

4 Main Improvements over the 2015 NLSE

One complication with Twitter-based evaluations is the need of the participant to retrieve the tweets themselves, since some of the tweets may no longer be available. The INESC-ID system presented in 2015 employed a train set of 8604 tweets, considerably smaller than the original dataset (with 11328 tweets). For this edition, it was possible to get ahold of the full dataset, as utilized by Severyn and Moschitti (2015). For reproducibility and comparison purposes our systems this year were developed with this dataset.

The system presented in 2015 was very simple both in its structure and the number hyperparameters. Furthermore, tunning and selection of candidate systems was also performed without automatic grid-search. It was therefore expected that our previous setup would outright produce better results by training on a larger dataset. Disappointingly, this was not the case. In fact, the NLSE optimized for the 2015 competition seemed to be sitting on a local optimum that was difficult to come out from. To overcome this problem, we introduced two modifications in the training procedure[1]. The NLSE is trained by minimizing the negative log-likelihood. This cost function is sub-optimal taking into account the evaluation metric, as it weights equally positive, negative and neutral predictions. A simple improvement over this cost is an asymmetric weighting that penalizes the predictions of neutral tweets. This was incorporated as a multiplicative factor on the log-likelihood of values $4/3$, $4/3$ and $1/3$ for the positive, negative and neutral classes, respectively. To reduce the risk of getting trapped into a local minimum, the train data was shuffled before each training epoch. The asymmetric cost and randomization led to a slower, less consistent convergence. For this reason the number of iterations was increased from 8 to 12. The learning rate was changed from 0.01 to 0.005. Table 2 shows the effect of the improvements on the submitted system.

After introducing these two improvements, we investigated different methods to address the problem of OOEV as described in the previous sec-

[1] After paper revision the model in `https://github.com/ramon-astudillo/NLSE` will be updated to reflect the new system.

tion. Namely those exploiting *C2W* embeddings, mapping *C2W* embeddings to *SSG* embeddings and training the embeddings for OOEVs. The results of these strategies are displayed in Table 3.

System	2013	2014	2015	2016
baseline	0.721	0.721	0.649	0.609
C2W embeddings	0.659	0.689	0.613	0.543
C2W→ *SSG*	**0.724**	0.715	0.652	**0.613**
update OOEVs 2 iter	0.723	**0.728**	**0.656**	0.610

Table 3: Comparision of strategies to address the problem of OOEV

5 The Submitted System

As mentioned in the previous section, the system submitted is an improvement over our 2015 system (Astudillo et al., 2015b). It therefore shares the same training characteristics as the previous model. The 52 million tweets used by Owoputi et al. (2013) and the tokenizer described in the same work were used to train the word embeddings *Structured Skip-Gram* (SSG). For this submission, the *C2W* embeddings were also trained using a publicly available toolkit[2]. For the annotated SemEval training data, the messages were previously pre-processed as follows: lower-casing, replacing Twitter user mentions and URLs with special tokens and reducing any character repetition to at most 3 characters. Following Astudillo et al. (2015a), we used embeddings with 600 dimensions and set the sub-space size to 10 dimensions.

To train the model, the development set was split into 80% for parameter learning and 20% for model evaluation and selection, maintaining the original relative class proportions in each set. The weights were all randomly initialized uniformly with ranges of $[-0.001, 0.001]$, $[-0.1, 0.1]$ and $[-0.7, 0.7]$ for the OOEVs, subspace and classification layers respectively. The training procedure entailed minimizing the negative log-likelihood over the training data with respect to the parameters, using standard Stochastic Gradient Descent (Rumelhart et al., 1985) with a fixed learning rate of 0.005 and mini-batch of size 1, i.e., updating the weights after each message was processed. We reshuffled the training examples after each training epoch and performed model selection by early stopping after 12 iterations. The candidate for submission was manually selected by observing the performance across 2013, 2014 and 2015 datasets. Priority was given to models that presented a consistent high performance in all the datasets. In retrospect, this was most probably a sub-optimal decision judging from the evaluation results.

System	2013	2014	2015	2016	Avg
SwissCheese	0.700_5	0.716_5	0.671_1	0.633_1	0.680_2
SENSEI-LIF	0.706_4	0.744_2	0.662_2	0.630_2	0.686_1
unimelb	0.687_7	0.706_7	0.651_4	0.617_3	0.665_4
INESC-ID	0.723_2	0.727_3	0.657_3	0.610_4	0.679_3
aueb	0.666_8	0.708_6	0.623_7	0.605_5	0.651_5

Table 4: Official test-set results for the top five systems in SemEval 2016 Task 4-B. Subscript number indicates position in general ranking.

Table 4 displays the performance for the top 5 systems in SemEval 2016 task 4-B (Nakov et al., 2016). The NLSE system (labeled INESC-ID) ranks forth with a stable performance across all years. The results are particularly strong for 2013 with a difference of 0.017 points over the next best performing system on the top five. This is consistent with the divide noticed during system selection between performance in 2013 and 2015. High-performing systems in 2014, and particularly in 2013, do not appear to be equally performing in recent years.

6 Conclusions

We presented the INESC-ID system for the SemEval 2016 task 4-A, built on top of the successful Non-Linear Subspace Embedding model. We found that training with a larger dataset required a more careful procedure to avoid overfitting. Reproducing the best results obtained in SemEval 2015 required shuffling the data before each training epoch and adapting the cost function to better reflect the evaluation metric.

To address the problem of out-of-embedding words, we tried to introduce character-level embeddings in our model but found these to be detrimental. We obtained better results by learning embeddings for these words during the training. Even though the performance gains were not very pronounced, our system still attained very strong results across all the evaluation datasets.

[2] https://github.com/wlin12/wang2vec

Acknowledgments

This work was partially supported by Fundação para a Ciência e Tecnologia (FCT), through contracts UID/CEC/50021/2013, EXCL/EEI-ESS/0257/2012 (DataStorm), grant number SFRH/BPD/68428/2010 and Ph.D. scholarship SFRH/BD/89020/2012.

References

Ramón Astudillo, Silvio Amir, Wang Ling, Mario Silva, and Isabel Trancoso. 2015a. Learning word representations from scarce and noisy data with embedding subspaces. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1074–1084, Beijing, China, July. Association for Computational Linguistics.

Ramon F. Astudillo, Silvio Amir, Wang Ling, Bruno Martins, Mário Silva, and Isabel Trancoso. 2015b. Inesc-id: Sentiment analysis without hand-coded features or liguistic resources using embedding subspaces. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '2015, Denver, Colorado, June. Association for Computational Linguistics.

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.

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*. Association for Computational Linguistics.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *In Proceedings of NAACL*.

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

Aliaksei Severyn 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, Denver, Colorado, June. Association for Computational Linguistics.