

Deep Learning for Social Sensing from Tweets

Giuseppe Attardi, Laura Gorrieri, Alessio Miaschi, Ruggero Petrolito

Dipartimento di Informatica

Università di Pisa

Largo B. Pontecorvo, 3

I-56127 Pisa, Italy

attardi@di.unipi.it

Abstract

English. Distributional Semantic Models (DSM) that represent words as vectors of weights over a high dimensional feature space have proved very effective in representing semantic or syntactic word similarity. For certain tasks however it is important to represent contrasting aspects such as polarity, opposite senses or idiomatic use of words. We present a method for computing discriminative word embeddings can be used in sentiment classification or any other task where one needs to discriminate between contrasting semantic aspects. We present an experiment in the identification of reports on natural disasters in tweets by means of these embeddings.

Italiano. *I Distributional Semantic Model (DSM) rappresentano le parole come vettori di pesi in uno spazio di feature ad alte dimensioni, e si sono dimostrati molto efficaci nel rappresentare la similarità semantica o sintattica tra parole. Per certi compiti per importante rappresentare aspetti contrastanti come la polarità, significati opposti o parole usate con significato idiomatico. Presentiamo un metodo per calcolare dei word embedding discriminativi che possono essere usati nella sentiment classification o per qualunque altro compito dove vi sia necessità di discriminare tra aspetti semantici contrastanti. Presentiamo un esperimento sull'identificazione di tweet relativi a calamità naturali utilizzando questi embedding.*

1 Introduction

Distributional Semantic Models (DSM) that represent words as vectors of weights over a high dimensional feature space (Hinton et al., 1986), have proved very effective in representing semantic or syntactic aspects of lexicon. Incorporating such representations has allowed improving many natural language tasks. They also reduce the burden of feature selection since these models can be learned through unsupervised techniques from plain text.

Deep learning algorithms for NLP tasks exploit distributional representation of words. In tagging applications such as POS tagging, NER tagging and Semantic Role Labeling (SRL), this has proved quite effective in reaching state of art accuracy and reducing reliance on manually engineered feature selection (Collobert & Weston, 2008).

Word embeddings have been exploited also in constituency parsing (Collobert, 2011) and dependency parsing (Chen & Manning, 2014). Blanco et al. (2015) exploit word embeddings for identifying entities in web search queries.

Traditional embeddings are created from large collections of unannotated documents through unsupervised learning, for example building a neural language model (Collobert et al. 2011; Mikolov et al. 2013) or through Hellinger PCA (Lehrer and Collobert, 2013). These embeddings are suitable to represent syntactic similarity, which can be measured through the Euclidean distance in the embeddings space. They are not appropriate though to represent semantic dissimilarity, since for example antonyms end up at close distance in the embeddings space

In this paper we explore a technique for building *discriminative word embeddings*, which incorporate semantic aspects that are not directly

obtainable from textual collocations. In particular, such embedding can be useful in sentiment classification in order to learn vector representations where words of opposite polarity are distant from each other.

2 Building Word Embeddings

Word embeddings provide a low dimensional dense vector space representation for words, where values in each dimension may represent syntactic or semantic properties.

For creating the embeddings, we used DeepNL¹, a library for building NLP applications based on a deep learning architecture. DeepNL provides two methods for building embeddings, one is based on the use of a neural language model, as proposed by Collobert et al. (2011) and one based on a spectral method as proposed by Lebre et al. (2013).

The neural language method can be hard to train and the process is often quite time consuming, since several iterations are required over the whole training set. Some researcher provide pre-computed embeddings for English².

Mikolov et al. (2013) developed an alternative solution for computing word embeddings, which significantly reduces the computational costs and can also exploit concurrency through the Asynchronous Stochastic Gradient Descent algorithm. An optimistic approach to matrix updates is also exploited to avoid synchronization costs.

The authors published single-machine multi-threaded C++ code for computing the word vectors³. A reimplementation of the algorithm in Python, but with core computations in C, is included in the Genism library (Řehůřek and Sojka, 2010)

Lebre et al. (2013) have shown that embeddings can be efficiently computed from word co-occurrence counts, applying Principal Component Analysis (PCA) to reduce dimensionality while optimizing the Hellinger similarity distance.

Levy and Goldberg (2014) have shown similarly that the skip-gram model by Mikolov et al. (2013) can be interpreted as implicitly factorizing a word-context matrix, whose values are the pointwise mutual information (PMI) of the re-

spective word and context pairs, shifted by a global constant.

2.1 Discriminative Word Embeddings

For certain tasks, as for example sentiment analysis, semantic similarity is not appropriate, since antonyms end up at close distance in the embeddings space. One needs to learn a vector representation where words of opposite polarity are distant.

Tang et al. (2013) propose an approach for learning sentiment specific word embeddings, by incorporating supervised knowledge of polarity in the loss function of the learning algorithm. The original hinge loss function in the algorithm by Collobert et al. (2011) is:

$$\mathcal{L}_{CW}(x, x^c) = \max(0, 1 - f_s(x) + f_s(x^c))$$

where x is an ngram and x^c is the same ngram corrupted by changing the target word with a randomly chosen one, $f_s(\cdot)$ is the feature function computed by the neural network with parameters θ . The sentiment specific network outputs a vector of two dimensions, one for modeling the generic syntactic/semantic aspects of words and the second for modeling polarity.

A second loss function is introduced as objective for minimization:

$$\mathcal{L}_{SS}(x, x^c) = \max(0, 1 - \delta_s(x) f_p(x)_1 + \delta_s(x) f_p(x^c)_1)$$

where the subscript in $f_p(x)_1$ refers to the second element of the vector and $\delta_s(x)$ is an indicator function reflecting the sentiment polarity of a sentence, whose value is 1 if the sentiment polarity of x is positive and -1 if it is negative.

The overall hinge loss is a linear combination of the two:

$$\mathcal{L}(x, x^c) = \alpha \mathcal{L}_{CW}(x, x^c) + (1 - \alpha) \mathcal{L}_{SS}(x, x^c)$$

Generalizing the approach to discriminative word embeddings entails replacing the loss function $\mathcal{L}_{ss}$ with a one-vs-all hinge loss function:

$$\mathcal{L}_h(x, t) = \max(0, 1 + \max_{y \neq t} (f(x)_t - f(x)_y))$$

where t is the index of the correct class.

The DeepNL library provides a training algorithm for discriminative word embedding that performs gradient descent using an adaptive learning rate according to the AdaGrad method. The algorithm requires a training set consisting of documents annotated with their discriminative value, for example a corpus of tweets with their sentiment polarity, or in general documents with

¹ <https://github.com/attardi/deepnl>

² <http://ronan.collobert.com/senna/>,
<http://metaoptimize.com/projects/wordreprs/>,
<http://www.fit.vutbr.cz/~imikolov/rnnlm/>,
<http://ai.stanford.edu/~ehuang/>

³ <https://code.google.com/p/word2vec>

multiple class tags. The algorithm builds embeddings for both unigrams and ngrams at the same time, by performing variations on a training sentence replacing not just a single word, but a sequence of words with either another word or another ngram.

3 Deep Learning Architecture

The Deep Learning architecture used for training discriminative word embeddings consists of the following layers:

1. Lookup layer: extracts the embedding vector associated to each token
2. Linear layer
3. Activation layer: using the hardtanh function
4. Linear layer
5. Hinge loss layer

4 Experiments

We tested the use of discriminative word embeddings in the task of social sensing, i.e. of detecting specific signals from social media. In particular we explored the ability to monitor and alert about emergencies caused by natural disasters. We explored the corpus of Social Sensing⁴, which consist of 5,642 tweets about natural catastrophic events like earthquakes or floods. To obtain a balanced training set, we combined this corpus with a set of generic tweets, consisting of 23,507 tweets. The combined corpus, consisting of 29,149 tweets, was randomly split into a training, development and test set consisting respectively of 23,850, 2,649 and 2,650 tweets.

4.1 Lexicon

Most sentiment analysis systems exploit a specialized lexicon (Rosenthal et al, 2014; Rosenthal et al, 2015). We built a lexicon of words related or indicative of disasters, by using the Italian Word Embeddings interface⁵. Starting from a seed set of few specialized words we produced a lexicon of 292 words (including words with a hashtag).

4.2 Classifier

For detecting tweets reporting about natural disasters, we exploit an SVM classifier, which uses as continuous features the word embeddings created from the text of the Italian Wikipedia. Addi-

tionally a set of discrete features is used, similar to those used in the top scoring system in the task 10 of SemEval 2014 on Sentiment Analysis in Twitter (Mohammad et al., 2014). These features are summarized in the following table:

<i>Type</i>	<i>Description</i>
<i>allcaps</i>	feature telling whether a word is all in uppercase
<i>EmoPos</i>	Presence of a positive emoticon
<i>EmoNeg</i>	Presence of a negative emoticon
<i>Elongated</i>	Presence of an elongated word
<i>Lexicon count</i>	Number of word present in a lexicon
<i>Lexicon min</i>	Lowest score of word in lexicon
<i>Lexicon last</i>	Score of the last word present in lexicon
<i>Lexicon sum</i>	Sum of the scores of words present in lexicon
<i>Negation</i>	Count of negative words
<i>Elongated punct</i>	Count of multiple punctuations (e.g. !!!)
<i>Ngrams</i>	Ngrams of length 2-4

4.3 Results

We created generic word embeddings on the corpus consisting of the plain text extracted from the Italian Wikipedia, for a total of 1,096,243,235 tokens, 4,456,972 distinct.

We selected the 100,000 most frequent words and we created word embeddings for them, with a space dimension of 64.

The table below shows the results obtained with the discriminative word embeddings compared to a baseline obtained with the same classifier using the generic embeddings.

Data	System	Precision	Recall	F1
Develop	baseline	85.91	72.66	78.73
Develop	DE	87.08	76.37	81.37
Test	baseline	86.87	70.96	78.11
Test	DE	85.94	75.05	80.12

The results show a significant improvement in recall with respect to the baseline, which leads to over a 2-point improvement in F1.

4.4 Related Work

Social sensing research is a rapidly growing field; however, it is difficult to compare our work with others since the data sets used are different.

The only experiment performed on the same data set, is described in (Cresci et al., 2015), which focuses on distinguishing whether damage

⁴ <http://socialsensing.it/en/datasets>

⁵ <http://tanl.di.unipi.it/embeddings/>

was reported, rather than just reporting a disaster. Sixteen experiments were carried out, using four subsets of the corpus for training, corresponding to four disaster events, and testing on either different events (*cross-event*) or same/different disaster types (*in-domain*, *out-domain*). F1 scores in detecting non relevant tweets ranged between 19% and 28% for *cross-event* and *out-domain* and reached 73% for *in-domain* in one of the *in-domain* tests.

5 Conclusions

We have presented the notion of discriminative word embeddings that were designed to cope with semantic dissimilarity in tasks like sentiment analysis or multiclass classification.

As an example of the effectiveness of this type of embeddings in other applications, we have explored their use in detecting tweets reporting alerts or notices about natural disasters.

Our approach consisted in using a classifier trained on a corpus of annotated tweets, using discriminative embeddings as features, instead of the typical manually crafted features or dictionaries employed in tweet classification tasks as sentiment analysis.

In the future, we plan to explore the use a convolutional network classifier, also provided by DeepNL, without any additional features, as Severyn and Moschitti (2015) have done for the SemEval 2015 task on Sentiment Analysis in Twitter.

References

- R. Al-Rfou, B. Perozzi, and S. Skiena. 2013. Polyglot: Distributed Word Representations for Multilingual NLP. arXiv preprint arXiv:1307.1662.
- R. K. Ando, T. Zhang, and P. Bartlett. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.
- Roi Blanco, Giuseppe Ottaviano, Edgar Meij, 2015. Fast and Space-efficient Entity Linking in Queries, ACM WSDM 2015.
- D. Chen and C. D. Manning. 2014. Fast and Accurate Dependency Parser using Neural Networks. In: Proc. of EMNLP 2014.
- R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008.
- R. Collobert. 2011. Deep Learning for Efficient Discriminative Parsing. In *AISTATS*, 2011.
- R. Collobert et al. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12, 2461–2505.
- S. Cresci, M. Tesconi, A. Cimino and F. Dell Orletta. 2015. A Linguistically-driven Approach to Cross-Event Damage Assessment of Natural Disasters from Social Media Messages. Proceedings of the 24th international conference companion on World Wide Web (WWW'15).
- M. Grbovic, N. Djuric, V. Radosavljevic, F. Silvestri, N. Bhamidipati. 2015. Context- and Content-aware Embeddings for Query Rewriting in Sponsored Search. *Proceedings of SIGIR 2015*, Santiago, Chile.
- Huang et al. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes, *Proc. of the Association for Computational Linguistics 2012 Conference*.
- G.E. Hinton, J.L. McClelland, D.E. Rumelhart. Distributed representations. 1986. In *Parallel distributed processing: Explorations in the microstructure of cognition*. Volume 1: Foundations, MIT Press, 1986.
- Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014. JMLR:W&CP volume 32.
- R. mi Lebrete and Ronan Collobert. 2013. Word Embeddings through Hellinger PCA. *Proc. of EACL 2013*.
- Omer Levy and Yoav Goldberg. 2014. Neural Word Embeddings as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.
- Christopher D. Manning and Hinrich Sch tze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press. Cambridge, Massachusetts.
- Saif M. Mohammad, Xiaodan Zhu, Svetlana Kiritchenko. 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), June 2013, Atlanta, USA.
- Saif M. Mohammad, Xiaodan Zhu, Svetlana Kiritchenko. 2014. Nrc-canada-2014: Recent improvements in sentiment analysis of tweets, and the Voted Perceptron. In Eighth International Workshop on Semantic Evaluation Exercises (SemEval-2014).
- T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010*, 11th Annual Conference of the International

- Speech Communication Association, Makuhari, Chiba, Japanfmikol.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*, 2013.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*, 2013.
- Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513-553.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, ELRA, Valletta, Malta, pp. 45-50.
- Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval 14*, pages 73-80, Dublin, Ireland.
- Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. *Proc. of the ninth International Workshop on Semantic Evaluation (SemEval-2015)*, Denver, USA.
- Aliaksei Severyn, 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, USA.
- S. Srivastava, E. Hovy. 2014. Vector space semantics with frequency-driven motifs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 634-643, Baltimore, Maryland, USA.
- Tang et al. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1555-1565, Baltimore, Maryland, USA, June 23-25 2014.
- 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*, pp. 384-394. Association for Computational Linguistics. 2013.