Recognizing Textual Entailment in Twitter Using Word Embeddings

Octavia-Maria Şulea
Human Language Technologies Research Center,
Faculty of Mathematics and Computer Science, University of Bucharest,
Academiei 14, 010014, Bucharest
mary.octavia@gmail.com

Abstract

In this paper, we investigate the application of machine learning techniques and word embeddings to the task of Recognizing Textual Entailment (RTE) in Social Media. We look at a manually labeled dataset (Lendvai et al., 2016) consisting of user generated short texts posted on Twitter (tweets) and related to four recent media events (the Charlie Hebdo shooting, the Ottawa shooting, the Sydney Siege, and the German Wings crash) and test to what extent neural techniques and embeddings are able to distinguish between tweets that entail or contradict each other or that claim unrelated things. We obtain comparable results to the state of the art in a train-test setting, but we show that, due to the noisy aspect of the data, results plummet in an evaluation strategy crafted to better simulate a real-life train-test scenario.

1 Introduction

The ability to automatically deduce how the meaning of text flows from one sentence to the next is a central part of Natural Language Understanding (NLU) and highly important in many Natural Language Processing tasks (NLP). Recognizing Textual Entailment (RTE), started as a challenge in 2004 from this very need and reaching its 8th iteration in 2013 at SemEval[1], falls at the intersection between NLU, NLP, Information Extraction, and Information Retrieval (Dagan et al., 2006). Its goal is, given a pair of sentences dubbed *text* and *hypothesis*, to determine whether the meaning of either (traditionally, of the hypothesis) entails the meaning of the other, contradicts it, or whether nothing can be said of the relationship between the two sentences. Here, the notion of entailment and contradiction are not necessarily related to the linguistic notions, where entailment is always explained in contrast with presupposition and sometimes implicature(Sauerland, 2007), but are outlined in (de Marneffe et al., 2008).

Interest in this task was amplified with the creation of the SNLI corpus (Bowman et al., 2015) which lead to a few studies using Deep Neural Networks (DNN) (Wang and Jiang, 2016; Rocktäschel et al., 2015). Previous to this, the Excitement Open Platform [2] was considered the state-of-the-art model.

On the hand, interest in ways to represent text in order to improve performance in text classification (Lilleberg et al., 2015; Joulin et al., 2016), machine translation (Zou et al., 2013; Sulea et al., 2016) or question answering tasks (Sharp et al., 2016) has been rekindled with the introduction of the highly cited word2vec model (Mikolov et al., 2013) and the avenue of deep neural word embeddings (Palangi et al., 2016).

Our present research revolved around three questions:

- Can we apply state of the art neural methods created for large datasets or longer texts to small datasets containing very short texts?
- Will these methods work for fine grained contradictions?
- Can word embeddings, which were successfully used for word-level semantic tasks, improve performance in tasks pertaining to discourse level semantics?

[1] https://www.cs.york.ac.uk/semeval-2013/task7/

[2] http://hltfbk.github.io/Excitement-Open-Platform/

2 Approach

In this paper we investigate the application of several state-of-the art approaches to the RTE task in the social media domain and investigate the use of word embeddings for what is essentially a discourse level semantic task. We use neural networks and compare our results with the results obtained previously using classical "feature engineering" methods.

2.1 Data

The data we used was presented in (Lendvai et al., 2016). It contains around 5000 Tweet pairs, distributed over four recent media events, reported in the press. These pairs were hand labeled as being either in a relationship of *entailment*, meaning that the underlying sense of each of the two text snippets was effectively the same, a relationship of *contradiction*, meaning that information in one of the tweets as minor as the number of victims or location of the event at hand contradicted the information expressed in the other tweet, or the two tweets were labeled to be in an *unknown* relationship, meaning that their underlying *stories* did not entail nor contradict each other, although they were referring to the same event. The dataset was slightly unbalanced, with the majority class pertaining to that of the *unknown* relationship and the minority class to the *contradiction* relationship.

Since the contradiction manifested between tweet pairs labeled was very fine grained, we expected bag of word models to perform poorly and confusion between entailment and contradiction to be high. Also, since, for each event, all three classes were represented, there was an expectation that BOW and similarity measures based on BOW would also fail, since pairs of tweets talking about the same event, but being in completely different relationships were abundant. Indeed, as can be seen from the results reported in (Lendvai et al., 2016), the f1 measure is slightly above the random baseline when using such features.

2.2 Classifier Implementation and Settings

For the implementation of the Multi Layer Perceptron classifier, we used the python library Keras [3] which wraps over the Deep Learning library for Python, Theano (Theano Development Team, 2016). The pre-trained word2vec model offered by Google was loaded into our system using the Gensim library (Řehůřek and Sojka, 2010). For the word-mover distance, we used pyemd [4], a Python wrapper for Pele and Werman's implementation of the Earth Mover's Distance metric (Pele and Werman, 2008).

The neural network was trained using several settings for the hyper-parameters (batch size and number of epochs) and we report the results for a batch size of 50 over 100 epochs. We also investigated several ways of representing the *t* and *h* text pairs.

2.3 Feature Representation

The first choice in representing the sentence pairs was to sum the 300-sized vector representation for each word in each of the two sentences separately and then concatenate the resulting 300-sized vectors into one. This lead to one 600-sized vectorial representation of the sentence pair. The second strategy lead to a 900-sized vector: the first 300 elements represented the sum of the vectors of words in the *t* text, the following 300 elements were 0s representing the separation vector, and the final 300 positions in the vector represented the sum of the vector for each word in the *h* text.

We also applied different similarity measures, including cosine and word mover's distance (Kusner et al., 2015), over the vectorial representations of the tweets. Ultimately, in terms of feature representation, we wanted to test two things:

- whether distance metrics between vectorial representations (weather one-hot or word2vec) of tweets are sufficient in predicting the RTE class
- whether inserting a *separation vector* between the two vectors for each of the texts in the pair leads to better results.

2.4 Event-based Cross Validation

In order to have a testing setup as close to a real-life scenario as possible, we employed the event-based cross validation, as proposed in (Lendvai et al., 2016). This effectively meant that, for each of the four events, we kept the tweets related to the other three events for training and used the tweets from the fourth event for testing. This meant that we had a 4-fold cross validation, where, for each *fold*, the train-test split was based on the event

[3] https://keras.io

[4] https://github.com/wmayner/pyemd

each pair belonged to. This in turn meant that, although the event label was never directly used as a feature or as the predicted label, it was indirectly used in cross-validation. This strategy was employed to simulate a real-life scenario where the end user of such an RTE system (e.g. a journalist trying to make sense of a large set of tweets on one event), would already have at their disposal a classification model pre-trained on other, possibly unrelated, events. We compared the results of event based cross-validation with typical train-test split results.

3 Results

Table 1 show the event-based cross validation results for the 3-way classification task when the features used are cosine distance between the sum of word2vec representation of the words in each tweet and word mover distance. More precisely, the cosine value and the word-mover distance value were concatenated to form a Nx2 feature matrix, where N was the number of input examples.

Model	Method	**P**	**R**	**F**
SVM	avg.	0.45	0.52	0.45
LR	avg.	0.46	0.52	0.45
Base	avg.	0.33	0.33	0.33
SVM	cont	0.38	0.49	0.40
LR	cont	0.43	0.53	0.46
Base	cont	0.26	0.31	0.28

Table 1: Event-based CV results using cosine similarity and word mover distance on the minority class and averaged

As can be seen from Figure 1, the distance metrics on occurrence vectors and word2vec summation vector are not good features to separate the three classes.

Logistic regression performed similar to Linear SVC when applied to the word2vec representation coupled with the word mover distance and averaging the event-based cross-validation results over all three classes. However, for the minority class contradiction, LR seemed to perform slightly better, although the standard deviation computed over each fold was higher.

For the 600 and 900 dimensional vector representation, the event-based CV results were slightly lower, as can be seen from Table 2.

Model	Method	**P**	**R**	**F**
MLP	avg.	0.41	0.34	0.30
LR	avg.	0.42	0.47	0.41
Dummy	avg.	0.35	0.35	0.35

Table 2: Event Based CV results for 900 dimensional vectors

Table 3 shows the train-test split results for the MLP and LR models over 900 dimensional vector

Model	Method	**P**	**R**	**F**
MLP	avg.	0.91	0.90	0.90
LR	avg.	0.78	0.78	0.78
Dummy	avg.	0.33	0.33	0.33
MLP	cont	0.87	0.77	0.82
LR	cont	0.62	0.60	0.61
Dummy	cont	0.26	0.26	0.26

Table 3: Train-Test Split results for LSTM and Logistic Regression

4 Conclusions and Future Work

In this paper we've investigated the use of current day classification tools for the task of recognizing textual entailment in Twitter data. We've shown that the same neural network models successfully used in the same task but on larger datasets perform similarly well (with only a small drop in performance) in a train-test split evaluation setting, but they perform as poorly as any other classifier in the event-based cross validation setting, a novel evaluation strategy, which was previously proposed to better simulate real life scenarios of RTE systems on Twitter.

We've also seen that using only the distance (cosine, word mover) between vector representations of the tweets, be those bow or sum of word2vec, was not enough to distinguish the minority class in the event-based cross validation setting and that using concatenation of word2vec leads to minor improvements in the same setting, but considerable in the train-test one.

Acknowledgements

Work presented in this paper has been supported by the PHEME FP7 project (grant No. 611233).

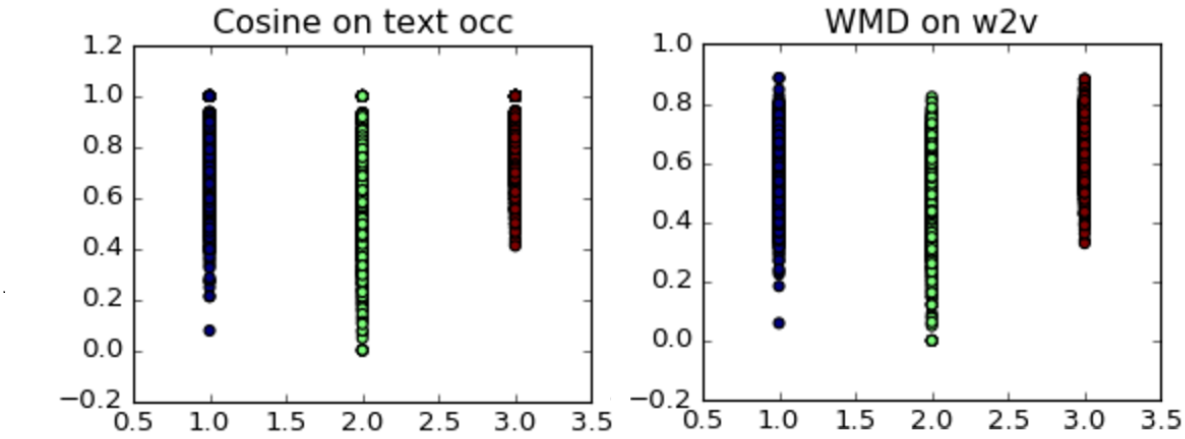

Figure 1: Plot of cosine distance on occurrence vector representation and word mover distance on word2vec summation representation for h and t; the horizontal axis represents the three classes and the vertical represents the distance values

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. The Association for Computational Linguistics, pages 632–642. http://aclweb.org/anthology/D/D15/D151075.pdf.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*. Springer-Verlag, Berlin, Heidelberg, MLCW'05, pages 177–190.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*. The Association for Computer Linguistics, pages 1039–1047. http://www.aclweb.org/anthology/P08-1118.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR* abs/1612.03651. http://arxiv.org/abs/1612.03651.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. JMLR.org, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 957–966. http://jmlr.org/proceedings/papers/v37/kusnerb15.html.

Piroska Lendvai, Isabelle Augenstein, Kalina Bontcheva, and Thierry Declerck. 2016. Monolingual social media datasets for detecting contradiction and entailment. In Nicoletta Calzolari (Conference Chair), Khalid Choukri (Conference Chair), Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hlne Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. ELRA, ELRA, 9, rue des Cordelires, 75013 Paris, 5/2016.

Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. 2015. Support vector machines and word2vec for text classification with semantic features. In Ning Ge, Jianhua Lu, Yingxu Wang, Newton Howard, Philip Chen, Xiaoming Tao, Bo Zhang, and Lotfi A. Zadeh, editors, *14th IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC 2015, Beijing, China, July 6-8, 2015*. IEEE Computer Society, pages 136–140. https://doi.org/10.1109/ICCI-CC.2015.7259377.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*. pages

3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio, Speech & Language Processing* 24(4):694–707. https://doi.org/10.1109/TASLP.2016.2520371.

Ofir Pele and Michael Werman. 2008. A linear time histogram metric for improved sift matching. In *Computer Vision–ECCV 2008*. Springer, pages 495–508.

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, pages 45–50. `http://is.muni.cz/publication/884893/en`.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR* abs/1509.06664. http://arxiv.org/abs/1509.06664.

U Sauerland. 2007. *Presupposition and Implicature in Compositional Semantics*.

Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. 2016. Creating causal embeddings for question answering with minimal supervision. *CoRR* abs/1609.08097. http://arxiv.org/abs/1609.08097.

Octavia-Maria Sulea, Sergiu Nisioi, and Liviu P. Dinu. 2016. Using word embeddings to translate named entities. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2016/summaries/1167.html.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688. http://arxiv.org/abs/1605.02688.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. The Association for Computational Linguistics, pages 1442–1451. http://aclweb.org/anthology/N/N16/N16-1170.pdf.

Will Y. Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pages 1393–1398. http://aclweb.org/anthology/D/D13/D13-1141.pdf.