

Conditional Random Fields for Metaphor Detection

Anna Mosolova¹, Ivan Bondarenko² and Vadim Fomin³

^{1,2,3}Novosibirsk State University

¹a.mosolova@g.nsu.ru ²i.yu.bondarenko@gmail.com ³wadimiusz@gmail.com

Abstract

We present an algorithm for detecting metaphor in sentences which was used in Shared Task on Metaphor Detection by First Workshop on Figurative Language Processing. The algorithm is based on different features and Conditional Random Fields.

1 Introduction

In this paper, we present a system which predicts metaphoricity of the word depending on its neighbors. We used VU Amsterdam corpus (Steen et al., 2010) given by competition’s organizers, 10 features which were also given by competition’s organizers and algorithm of Conditional Random Fields for predictions that are depending on previous ones.

2 Related Work

A lot of papers describe methods for metaphor detection, but the closest in performance is the article by Rai et al. (2016). It proposes to use Conditional Random Fields for metaphor detection. The authors also use features based on syntax, concepts, affects, and word embeddings from MRC Psycholinguistic Database and coherence and analogy between words which are taken from word embeddings given by Huang et al. (2012). Moreover, they use synonymy from WordNet.

This work is very similar to our due to some similar features and the main algorithm which is CRF.

3 Data

3.1 Dataset

As a dataset was used VU Amsterdam corpus (Steen et al., 2010). It consists of 117 texts divided into 4 parts (academic, news, fiction, conversation).

It was divided into two parts: train and test. The model was trained on the train set and evaluated on the test set.

3.2 Features

Features were given by competition’s organizers. Set of features consists of:

- Unigrams: All words from the training data without any changes;
- Unigram lemmas: All words from the training data in their normal form;
- Part-of-Speech tags: They were generated by Stanford POS tagger 3.3.0 (Toutanova et al. 2003);
- Topical LDA: Latent Dirichlet Allocation (Blei et al., 2003) for deriving a 100-topic model from the NYT corpus years 2003-2007 (Sandhaus, 2008) for representing common topics of public discussions. The NYT data was lemmatized using NLTK (Bird, 2006) and the model was built using the gensim toolkit (R. Řehůřek and P. Sojka, 2010);
- Concreteness: For this feature was used Brysbaert et al. (2013) database of concreteness ratings for about 40,000 English words. The mean ratings, ranging 1-5, are binned in 0.25 increments; each bin is used as a binary feature;

- WordNet: 15 lexical classes of verbs based on their general meanings;
- VerbNet: Classification based on syntactic frames of verbs ;
- Corpus: 150 clusters of verbs using their subcategorization frames and the verb’s nominal arguments as features for clustering.

All of these features were described in Beigman Klebanov et al. (2014), Beigman Klebanov et al. (2015) and Beigman Klebanov et al. (2016).

3.3 Algorithm

As an algorithm for classification was used Conditional Random Fields which was described in Lafferty et al. (2001). This algorithm depends on previous predictions making the future ones and it was crucial because metaphoricity of a word in a sentence relies on its neighbors. Also, this classifier can work with a big amount of features, so we used a lot of them in this work and it was helpful for the further results.

4 Experiments

We tried different parameters that were provided in the crfsuite (Okazaki, 2007). There were five training algorithms such as lbfgs (gradient descending using the L-BFGS method), l2sgd (stochastic gradient descend with L2 regularization term), Averaged Perceptron, Passive Aggressive, Adaptive Regularization Of Weight Vector. The best training algorithm was lbfgs.

Moreover, we used a different amount of iterations, and its amount affects the loss because there is no limit to the number of iterations in the lbfgs-algorithm.

Furthermore, some experiments with regularization were conducted. Regularization was used for reducing the generalization error and it is important in CRF. For the selection of the most appropriate parameters for regularization, we used RandomizedSearchCV from scikit-learn (<http://scikit-learn.org>).

We used sklearn-crfsuite that is the special wrapper of crfsuite written in C for Python (<https://github.com/TeamHG-Memex/sklearn-crfsuite>) for computing the algorithm.

As a metric for evaluating the score was taken F-score.

The best F-score had the algorithm with 200 iterations, lbfgs-algorithm, c1 regularization and c2 regularization that equal to 0.1.

The result obtained with these parameters was evaluated using a held-out set from the train set. F-score of this model and other experiments are presented in table 1 for All-POS track and for Verb track.

Parameters	F-score for all-POS	F-score for Verbs track
lbfgs, 200 iterations, c1=c2=0.1	0.8621	0.7417
lbfgs, 100 iterations, c1=c2=0.1	0.8593	0.739
lbfgs, 50 iterations, c1=c2=0.1	0.8601	0.7333
lbfgs, 100 iterations, c1=0.2353, c2=0.0329,	0.8586	0.7528
l2sgd, 100 iterations, c2=0.1	0.8455	0.6343
Averaged Perceptron, 100 iterations	0.8303	0.7165
Passive Aggressive, 100 iterations	0.8483	0.7327

Table 1 The results of the experiment for All-POS and Verb tracks.

Adaptive Regularization Of Weight Vector, 100 iterations	0.8459	0.6973
--	--------	--------

5 Results

As a result, our best-trained model was based on 10 features described below and CRF classifier with lbfgs and 200 iterations and it has F-score equal to 0.8621 for All-POS track. As for the Verb track, the best model was also based on lbfgs, had 100 iterations and c1 equal to 0.2353, c2 equal to 0.0329 with F-score 0.7528.

These results are obtained using validation with a part of the train set, and as for the test set, for All-POS track, the result measured by F-score is 0.138 and for Verb track is 0.246.

The results differ as it is possible that validation on a small part of the train set (33%) is not as accurate as validation on the test set which usually consists of the larger number of sentences.

6 Conclusion

We used Conditional Random Fields for the task of metaphor detection. Due to the large number of features, this classifier worked very well, and it is assumed that increasing the number of features will improve the performance of the algorithm.

References

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. *Latent Dirichlet Allocation*. *Journal of Machine Learning Research*, 3:993–1022.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2013. *Concreteness ratings for 40 thousand generally known english word lemmas*. *Behavior Research Methods*, pages 1–8.
- Beata Beigman Klebanov, Chee Wee Leong, Michael Heilman, and Michael Flor. 2014. *Different texts, same metaphors: Unigrams and beyond*. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 11–17, Baltimore, MD, June. Association for Computational Linguistics.
- Beata Beigman Klebanov, Chee Wee Leong, and Michael Flor. 2015. *Supervised word-level metaphor detection: Experiments with concreteness and re-weighting of examples*. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 11–20, Denver, Colorado, June. Association for Computational Linguistics.
- Huang, Eric H., Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. *Improving word representations via global context and multiple word prototypes*. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics.
- John Lafferty, Andrew McCallum, Fernando Pereira. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann. pp. 282–289.
- Naoaki Okazaki. 2007. *CRFsuite: a fast implementation of Conditional Random Fields (CRFs)*. <http://www.chokkan.org/software/crfsuite/>
- Sunny Rai, Shampa Chakraverty, and Devendra K. Tayal. 2016. *Supervised metaphor detection using conditional random fields*. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 18–27, San Diego, California. Association for Computational Linguistics.
- 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*, pages 45–50, Valletta, Malta, May. ELRA.
- Evan Sandhaus. 2008. *The New York Times Annotated Corpus*. LDC Catalog No: LDC2008T19.
- Gerard Steen, Aletta Dorst, Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A Method for Linguistic Metaphor Identification*. Amsterdam: John Benjamins.
- Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. In *Proceedings of NAACL*, pages 252–259.