MSejrKu at SemEval-2016 Task 14: Taxonomy Enrichment by Evidence Ranking

Michael Sejr Schlichtkrull
University of Copenhagen (Denmark)
michael.sejr@gmail.com

Héctor Martínez Alonso
Univ. Paris 7 - INRIA (France)
hector.martinez-alonso@inria.fr

Abstract

Automatic enrichment of semantic taxonomies with novel data is a relatively unexplored task with potential benefits in a broad array of natural language processing problems. Task 14 of SemEval 2016 poses the challenge of designing systems for this task. In this paper, we describe and evaluate several machine learning systems constructed for our participation in the competition. We demonstrate an f1-score of 0.680 for our submitted systems — a small improvement over the 0.679 produced by the hard baseline.

1 Introduction

This article describes our systems submitted for SemEval2016, task 14 on taxonomy enrichment. [1] The two submitted runs are based on Gaussian-kernel SVMs, and fall respectively under the constrained and the unconstrained condition of the shared task, namely using only WordNet and the training data, or incorporating outside sources. We chose our submitted models by evaluation on the trial data in lieu of a development set. Our runs perform better than the very hard baseline with a small 2% margin, while the best-performing heuristic outperforms the baseline by a 5% margin.

2 Related Work

Existing methods for taxonomy enrichment can roughly be divided into two categories: relying on alignment between multiple taxonomies, or relying on machine learning-based rating of subgraphs.

In (Jurgens and Pilehvar, 2015), Wordnet is extended with technical terms and rare lemmas from Wiktionary. In (Suchanek et al., 2007), relation-extraction is used to unify WordNet and Wikipedia. (Navigli and Ponzetto, 2012) further automates the alignment process itself. In (Toral et al., 2008), named entities are brought from Wikipedia to WordNet through pattern matching.

In (Snow et al., 2006), the probabilities of taxonomies are evaluated based on evidence vectors associated to edges. A similar approach formulated in terms of factor graphs can be seen in (Bansal et al., 2013). Finally, (Yamada et al., 2011) employ a hybrid strategy, scoring edges by likelihood of appearance in Wikipedia. Our approach also falls in the second category, relying on a machine learning process to rank hypernym-hyponym edges.

3 Data

There are three data splits: *train*, *trial*, and *test*, described in Table 1. We use the trial data for model selection. Notice that attach-actions represent the vast majority of instances. We therefore focus exclusively on finding correct integrations and disregard the merge-attach distinction.

Several observations can be made from the distribution of the labels in the training set. First, we define the *closest* synset $s_c(t, g)$ of a term t with gold standard g as the synset such that a lemma of that synset is represented in some description sentence $d \in D(t)$ and the shortest path $\delta(g, s_c(t, g))$ in the taxonomy is minimized. Here, $D(t)$ is the set of description sentences corresponding to t. In Figure 1, we plot $\delta(g, s_c(t, g))$ versus number of instances.

[1] http://alt.qcri.org/semeval2016/task14/

Proceedings of SemEval-2016, pages 1337–1341,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

Dataset	# Nouns	# Verbs	# Total	Sentences / instance	Tokens / sentence	# Attach	# Merge
Train	349	51	400	1.69	17.73	367	33
Trial	93	34	127	1.02	12.18	71	56
Test	512	82	600	1.21	18.45	569	31

Table 1: Overview of the datasets, showing the composition of nouns and verbs, merge-action and attach-actions, and the mean number of description sentences per term and number of tokens per description sentence.

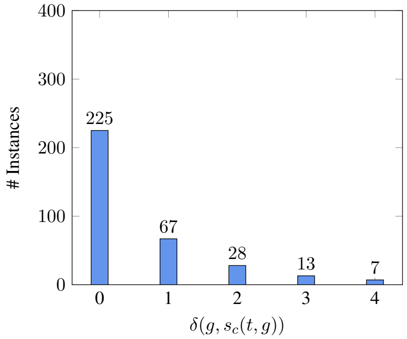

Figure 1: Number of instances in the training set such that $\delta(g, s_c(t, g))$ corresponds respectively to $0, 1, 2, 3,$ and 4.

As is apparent from the Zipfian-like distribution, s_c constitutes an excellent guess for g. Defining $w_c(t, d, g)$ for each $d \in D(t)$ in similar fashion as the synset minimizing $\delta(g, w_c(t, g))$, an analogous observation can be made for each description sentence. The result can be seen in Figure 2 below:

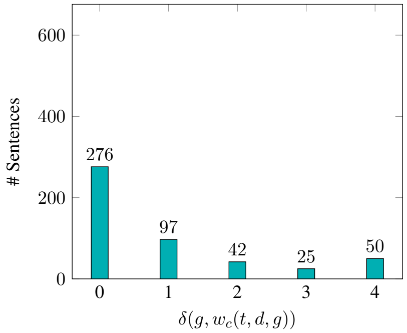

Figure 2: Number of sentences in the training set such that $\delta(g, w_c(t, d, g))$ corresponds respectively to $0, 1, 2, 3,$ and 4.

We again observe a Zipfian-like distribution, suggesting that each $w_c(t, d, g)$ represents a good candidate for $s_c(t, g)$. As terms often have multiple description sentences, we therefore have for each term several good guesses for $s_c(t, g)$ and therefore g. These observations form the basis of the strategies discussed in Section 5.2.

4 Features

Our systems use both lexical and syntactic features, with distributional features included in the second run. As a first step, we POS-tag and dependency parse the description sentences using the Mate parser in (Bohnet et al., 2013).[2] Additionally, we apply the unsupervised word sense disambiguation algorithm described in (Agirre and Soroa, 2009).[3]

4.1 Constrained Features

Description words are represented through features based on position, word shape, morphology, POS-tag, and dependency structure. Word senses are incorporated through the depths defined in (Devitt and Vogel, 2004). Morphological and shape features derived from the candidate term are also included, along with a binary feature representing the description word appearing in the target term.

For the direct classification strategies, we also use features derived from the candidate synset. These include POS-tag, overlap between synset- and term-description, and the length of the shortest path to the description. For the ranking systems, we also utilize pairwise features: the relative distance and position in the description sentence, the relative distance and position in the dependency tree, and the difference in number of overlapping lemmas between the description sentence and the most likely senses.

4.2 Embedding Features

We extend the features defined above with Skip-Gram embeddings as discussed in (Mikolov et al., 2013). We train the Gensim model (Řehůřek and Sojka, 2010) on a corpus consisting of the description sentences and the Wikipedia entries for each term, padded with the English part of the PolyGlot corpus presented in (Al-Rfou et al., 2013).

[2] Available at: https://code.google.com/archive/p/mate-tools/wikis/ParserAndModels.wiki .

[3] Available at: http://ixa2.si.ehu.es/ukb/ .

Word-level features are extended with embeddings for the word itself and the dependency head. Term-level features are extended with the sum of the embedding vectors of each word in the n-gram. Synset-level features are extended with the mean of the embedding vectors for each lemma. Finally, all pairwise features are extended with cosine distances.

5 Experiments

In this section, we explain the strategies we attempted. In all cases, we discount the merge action, focusing only on attaching.

5.1 Direct Classification

For comparison, we first attempt direct classification: Given a term and a candidate synset, predict whether that synset is a correct integration for the term. Enrichment is done by ranking all synsets according to prediction probability. We tried a linear logistic regression classifier, and a non-linear neural network classifier with a single hidden layer containing 100 units.

5.2 Right Word Classifier

As shown in Section 3, one can narrow down the search space by a large margin through finding $w_c(t, d, g)$. We propose to determine w_c through a machine learning process. Then, we estimate $s_c(t, g)$ by taking a majority vote over sentences. We experiment with four versions of the strategy.

We consider letting each sentence vote for the most likely sense of the best candidate for $w_c(t, d, g)$ with a weight of 1. Secondly, we consider including votes for the second-best candidate and the second-most likely senses, weighted by hyperparameters α and β. We determine the values for α and β through crossvalidation on the training set. We perform word sense disambiguation either through most frequent sense, or through Personalized Pagerank.

Ranking problems can be approached as pointwise regression, or as pairwise classification. Ranking by classification probability as discussed in the previous section corresponds to a pointwise regression with logistic loss. Following (Hang, 2011), transforming a pointwise problem into a pairwise problem can improve performance, especially when training data is scarce.

To transform the training set into a pairwise training set $\mathcal{L}_{pairwise}$, we combine for each description sentence d with content words of appropriate POS-tag $W_{t,d}$ every pair $(w_1, w_2) \in W_{t,d} \times W_{t,d}$ such that one equals $w_c(t, d, g)$ and the other does not. If w_1 is the match and w_2 is not, we add (w_1, w_2) to $\mathcal{L}_{pairwise}$ with the positive label. Else, we add (w_1, w_2) to $\mathcal{L}_{pairwise}$ with the negative label. We train a classifier on this dataset, observing that a relation $\prec_{t,d}$ is induced as positive and negative predictions on w_1 and w_2 correspond respectively to $w_1 \succ_{t,d} w_2$ and $w_1 \prec_{t,d} w_2$.

We experiment with various classifiers and parameter settings through cross-validation. As $(w_1, w_2) \in \mathcal{L}_{pairwise} \Leftrightarrow (w_2, w_1) \in \mathcal{L}_{pairwise}$, the dataset is balanced and chance performs at an accuracy of 0.5. The best-performing classifier was the Gaussian kernel SVM, with an accuracy of 0.81.

Through this classification, we learn for each term t a relation $\prec_{t,d}$ on the words $w \in W_{t,d}$ of each description sentence $d \in D(t)$. This relation, however, is asymmetric and therefore does not constitute an ordering. We therefore define a new ordering $\prec^*_{t,d}$ averaging over both directions of $\prec_{t,d}$:

$$p(w_1 \prec^*_{t,d} w_2) = \frac{p(w_1 \prec_{t,d} w_2) + p(w_2 \succ_{t,d} w_1))}{2}$$

We then let $w_1 \prec^*_{t,d} w_2$ if and only if $p(w_1 \prec^*_{t,d} w_2) > 0.5$. As $\prec^*_{t,d}$ is an ordering on the words of each description sentence, we let the largest element with respect to $\prec^*_{t,d}$ be our guess for w_c.

6 Results

The results for each system can be seen in Table 2. We provide results on the training data, the test data, and the trial data which was used as a development set in the model selection process. Apart from the submitted runs, all systems have a perfect recall of 1.0. For the submitted runs, a software error caused recall to drop to 0.97 as description words containing underscores were incorrectly processed.

The best-performing system on the trial data under both conditions is the vote-based ranking without sense disambiguation, outperforming the *first word, first sense*-baseline by a small margin. Following this observation, we chose to submit the constrained and unconstrained variants of this system - marked in bold in Table 2.

System	Train		Trial		Test	
	F1	Lemma M.	F1	Lemma M.	F1	Lemma M.
Random	0.35	0.00	0.39	0.00	0.37	0.00
First-word fist-sense	0.64	0.33	0.66	0.25	0.68	0.42
Logistic Regression	0.53	0.17	0.46	0.15	0.47	0.15
NN-classification	0.53	0.18	0.46	0.14	0.50	0.17
Rank-FS	0.69	0.43	0.66	0.28	0.68	0.42
Rank-WSD	0.72	0.42	0.65	0.29	0.70	0.42
Rank-vote-FS	0.70	0.42	0.67	0.27	**0.67**	**0.43**
Rank-vote-WSD	0.72	0.43	0.66	0.27	0.70	0.41
Rank-embed-FS	0.69	0.39	0.67	0.28	0.68	0.41
Rank-embed-WSD	0.73	0.42	0.67	0.28	0.68	0.42
Rank-embed-vote-FS	0.70	0.42	0.67	0.28	**0.68**	**0.43**
Rank-embed-vote-WSD	0.73	0.42	0.67	0.28	0.69	0.41

Table 2: Performance reported as the f1 of Wu & Palmer-score and recall, along with Lemma Matches. Results are shown for the baselines, the direct classifiers, and the constrained and unconstrained versions of the ranking-based systems. The submitted runs are marked in bold. Recall is 1.00 for all systems except for the submitted runs, where a software error caused a recall of 0.97.

7 Discussion

Investigating the errors made by the voting-based first-sense systems, we see a pattern in which hypernyms of the correct integration mistakenly are chosen. Examples include *steroid* instead of *anabolic steroid*, *drug* instead of *opiate*, and *person* instead of *woman*. Although these errors occur under both conditions, they are slightly more frequent in the constrained system. In (Levy et al., 2015), evidence is presented that distributional embeddings mainly encode hierarchical *level* rather than hierarchical *branch*. Our findings seem to support this conclusion. Refining the usage of the embeddings may well resolve this particular source of error.

In Table 1, we see how the training dataset not only on the average has more description sentences per term, but also longer description sentences. As the Personalized Pagerank algorithm relies on a context created from these sets of tokens, this may be the source of the poor performance of sense disambiguation compared to most frequent sense on the trial data seen in Table 2.

Similarly, the voting component was observed to have almost no effect on the test and trial data, while improving the performance on the training data by a wide margin. When the number of description sentences is smaller, it follows naturally that the number of candidates for s_c drops, thus reducing the effectiveness of voting.

Examining the difference between the predictions made by our systems and the baseline, we notice that no terms exist where we our prediction is equal to the gold standard, and the baseline is not. Rather, the improvement demonstrated by our systems comes in the form of wrongful guesses being closer to the gold than is the case for the baseline. This is further supported by the Zipfian observed in Section 3: our guesses mostly fall within a distance of 0–2 edges of the gold standard.

8 Conclusion

We have described several strategies for taxonomy enrichment, showing F1-scores of respectively 0.68 and 0.70 for the submitted and the best-performing systems. Our results show much better performance on the training data, even for the unsupervised sense disambiguation component. We have argued that the difference in number of description sentences and number of tokens per description sentence accounts partially for this difference, and therefore propose the gathering of additional evidence from outside sources to be added to the descriptions as the natural next step for future work. As our system demonstrates high performance for candidate generation and low performance for candidate selection, another possibility for future work would be to combine our approach with a filtering strategy in an ensemble or pipeline.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August.

Mohit Bansal, David Burkett, Gerard De Melo, and Dan Klein. 2013. Structured learning for taxonomy induction with belief propagation.

Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas, Filip Ginter, and Jan Hajič. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1:415–428.

Ann Devitt and Carl Vogel. 2004. The topology of wordnet: Some metrics.

LI Hang. 2011. A short introduction to learning to rank. *IEICE TRANSACTIONS on Information and Systems*, 94(10):1854–1862.

David Jurgens and Mohammad Taher Pilehvar. 2015. Reserating the awesometastic: An automatic extension of the wordnet taxonomy for novel terms.

Omer Levy, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. 2015. Do supervised distributional methods really learn lexical inference relations?

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.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

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.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. Association for Computational Linguistics.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Antonio Toral, Rafael Munoz, and Monica Monachini. 2008. Named entity wordnet.

Ichiro Yamada, Jong-Hoon Oh, Chikara Hashimoto, Kentaro Torisawa, Jun'ichi Kazama, Stijn De Saeger, and Takuya Kawada. 2011. Extending wordnet with hypernyms and siblings acquired from wikipedia. In *IJCNLP*, pages 874–882.