CogALex-V Shared Task: ROOT18

Emmanuele Chersoni
Aix-Marseille University
emmanuelechersoni@gmail.com

Giulia Rambelli
University of Pisa
rambelligiulia@gmail.com

Enrico Santus
The Hong Kong Polytechnic University
esantus@gmail.com

Abstract

In this paper, we describe ROOT 18, a classifier using the scores of several *unsupervised distributional measures* as features to discriminate between semantically related and unrelated words, and then to classify the related pairs according to their semantic relation (i.e. *synonymy*, *antonymy*, *hypernymy*, *part-whole meronymy*). Our classifier participated in the CogALex-V Shared Task, showing a solid performance on the first subtask, but a poor performance on the second subtask. The low scores reported on the second subtask suggest that distributional measures are not sufficient to discriminate between multiple semantic relations at once.

1 Introduction

The system described in this paper has been designed for the CogALex-V Shared Task, focusing on the corpus-based identification of semantic relations. Since Distributional Semantic Models (henceforth DSMs) were proposed as a special topic of interest for the current edition of the CogALex workshop, we decided to base our classifier on a number of distributional measures that have been used by past Natural Language Processing (NLP) research to discriminate between a specific semantic relation and other relation types.

The task is splitted into the following subtasks:

- for each word pair, the participating systems have to decide whether the terms are semantically related or not (TRUE and FALSE are the only possible outcomes);
- for each word pair, the participating systems have to decide which semantic relation holds between the terms of the pair. The five possible semantic relations are synonymy (SYN), antonymy (ANT), hypernymy (HYPER), meronymy (PART_OF) and no semantic relation at all (RANDOM).

Our system managed to achieve good results in discriminating between related and random pairs in the first subtask, but unfortunately it struggled in the second one, also due to the high difficulty of the task itself. In particular, the recall for some of the semantic relations of interest seems to be extremely low, suggesting that our unsupervised distributional measures do not provide sufficient information to characterize them, and that it could be probably useful to integrate such scores with other sources of evidence (e.g. information on lexical patterns of word co-occurrence).

The paper is organized as follows: in section 2, we summarize related works on the task of semantic relation identification; in section 3, we introduce our system, by describing the classifier and the features. Finally, in section 4 we present and discuss our results.

2 The Task: Related Work

Distinguishing between related and unrelated words and, then, discriminating among semantic relations are very important tasks in NLP, and they have a wide range of applications, such as textual entailment, text summarization, sentiment analysis, ontology learning, and so on. For this reason, several systems

Proceedings of the Workshop on Cognitive Aspects of the Lexicon,
pages 98–103, Osaka, Japan, December 11-17 2016.

over the last few years have been proposed to tackle this problem, using both unsupervised and supervised approaches (see the works of Lenci and Benotto (2012) and Shwartz et al. (2016) on hypernymy; Weeds et al. (2014) and Santus et al. (2016a) on hypernymy and co-hyponymy; Mohammad et al. (2013) and Santus et al. (2014) on antonymy). However, many of these works focus on a single semantic relation, e.g. antonymy, and describe methods or measures to set it apart from other relations. There have not been many attempts, at the best of our knowledge, to deal with corpus-based semantic relation identification in a multiclass classification task. Few exceptions include the works by Turney (2008) on similarity, antonymy and analogy, and by Pantel and Pernacchiotti (2006) on Espresso, a weakly supervised, pattern-based algorithm. Both these systems are based on patterns, which are known to be more precise than DSMs, even though they suffer from lower recall (i.e. they in fact require words to co-occur in the same sentence). DSMs, on the other hand, offer higher recall at the cost of lower precision: while they are strong in identifying distributionally similar words (i.e. nearest neighbors), they do not offer any principled way to discriminate between semantic relations (i.e. the nearest neighbors of a word are not only its synonyms, but they also include antonyms, hypernyms, and so on).

The attempts to provide DSMs with the ability of automatically identifying semantic relations include a large number of unsupervised methods (Weeds and Weir, 2003; Lenci and Benotto, 2012; Santus et al., 2014), which are unfortunately far from achieving the perfect accuracy. In order to achieve higher performance, supervised methods have been recently adopted, also thanks to their ease (Weeds et al., 2014; Roller et al., 2014; Kruszewski et al., 2015; Roller and Erk, 2016; Santus et al., 2016a; Nguyen et al., 2016; Shwartz et al., 2016). Many of them rely on distributional word vectors, either concatenated or combined through algebraic functions. Others use as features either patterns or scores from the above-mentioned unsupervised methods. While these systems generally obtain high performance in classification tasks involving a single semantic relation, they have rarely been used on multiclass relation classification. On top of it, some scholars have questioned their ability to really learn semantic relations (Levy et al., 2015), claiming that they rather learn some lexical properties from the word vectors they are trained with. This was also confirmed by an experiment carried out by Santus et al. (2016a), showing that up to 100% synthetic switched pairs (i.e. *banana-animal*; *elephant-fruit*) are misclassified as hypernyms if the system is not provided with some of these negative examples during training.

Recently, count based vectors have been substituted by prediction-based ones, which seem to slightly improve the performance in some tasks, such as similarity estimation (Baroni et al., 2014), even though Levy et al. (2015) demonstrated that these improvements were most likely due to the optimization of hyperparameters that were instead left unoptimized in count-based models (for an overview on word embeddings, see Gladkova et al. (2016)). On top of it, when combined with supervised methods, the low interpretability of their dimensions makes it even harder to understand what the classifiers actually learn (Levy et al., 2015).

Finally, the recent attempt of Shwartz et al. (2016) of combining patterns and distributional information achieved extremely promising results in hypernymy identification.

3 System description

Our system, ROOT18, is a Random Forest classifier (Breiman, 2001) and it is based on the 18 features described in the following subsections. The system in its best setting makes use of the Gini impurity index as the splitting criterion and has 10 as the maximum tree depth. The half of the total number of features were considered for each split.

3.1 Data

Our data come from *ukWaC* (Baroni et al., 2009), a 2 billion tokens corpus of English built by crawling the .uk Internet domain. For the extraction of our features, we generated several distributional spaces, which differ according to the window size and to the statistical association measure that was used to weight raw co-occurrences. Since we obtained the best performances with window size 2 and Positive Pointwise Mutual Information (Church and Hanks, 1990), we report the results only for this setting.

3.2 Features

Frequency It is a basic property of words and it is a very discriminative information. In this type of task, it proved to be competitive in identifying the directionality of pairs of hypernyms (Weeds and Weir, 2003), since we expect hypernyms to have higher frequency than hyponyms. For each pair, we computed three features: the frequency of each word (*Freq1,2*) and their difference (*DiffFreq*).

Co-occurrence We compute the co-occurrence frequency (*Cooc*) between the two terms in each pair. This measure has been claimed to be particularly useful to spot antonyms (Murphy, 2003), since they are expected to occur in the same sentence more often than chance (e.g. *Are you friend or foe?*).

Entropy In information theory, this score is related to the informativeness of a message: the lower its entropy, the higher its informativeness (Shannon, 1948). Moreover, subordinate terms tend to have higher amounts of informativeness than superordinate ones. We computed the entropy of each word in the pair (*Entr1,2*), plus the difference between entropies (*DiffEntr*).

Cosine similarity It is a standard measure in DSMs to compute similarity between words (Turney and Pantel, 2010). This measure is very useful to discriminate between related and unrelated terms.

$$sim(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|}$$

LinSimilarity LinSimilarity (Lin, 1998) is a different similarity measure, computed as the ratio of shared context between *u* and *v* to the contexts of each word:

$$Lin(\vec{u}, \vec{v}) = \frac{\sum_{c \in \vec{u} \bigcap \vec{v}} [\vec{u}[c] + \vec{v}[c]}{\sum_{c \in \vec{u}} \vec{u}[c] + \sum_{c \in \vec{v}} \vec{v}[c]}$$

Directional similarity measures We extracted several directional similarity measures that were proposed to detect hypernyms, such as *WeedsPrec*, *cosWeeds*, *ClarkeDe* and *invCL* (for a review, see Lenci and Benotto (2012)). They are all based on the *Distributional Inclusion Hypotesis*, according to which if a word *u* is semantically narrower to *v*, then a significant number of the salient features of *u* will be included also in *v*.

APSyn This measure and the following *APAnt* do not rely on the full distribution of words, but on the top *N* most related contexts of the words according to some statistical association measure. APSyn (Santus et al., 2016b) computes a weighted intersection of the top *N* context of the target words:

$$APSyn(w_1, w_2) = \sum_{f \in N(F_1) \bigcap N(F_2)} \frac{1}{(rank_1(f) + rank_2(f))/2}$$

That is, for every feature f included in the intersection between the top N features of w_1 and w_2 ($N(F_1)$, $N(F_2)$ respectively), the measure adds 1 divided by the average rank of the feature in the rankings of the top N features of w_1 and w_2.

APAnt *APAnt* (Santus et al., 2014) is defined as the inverse of APSyn. This unsupervised measure tries to discriminate between synonyms and antonyms by relying on the hypothesis that words with similar distribution (i.e. high vector cosine) that do not share their most relevant contexts (i.e. what APSyn computes) are likely to be antonyms. For each pair, we computed APSyn and APAnt for the top 1000 and for the top 100 contexts.

Same POS We realized that many of the random pairs in the data included words with different parts of speech. Therefore, we decided to add a boolean value to our set of features: 1 if the most frequent POS of the words in the pair were the same, 0 otherwise.

3.3 Evaluation dataset

The task organizers provided a training and a test set extracted from EVALution 1.0, a resource that was specifically designed for evaluating systems on the identification of semantic relations (Santus et al., 2015). EVALution 1.0 was derived from WordNet (Fellbaum, 1998) and ConceptNet (Liu and Singh, 2004) and it consists of almost 7500 word pairs, instantiating several semantic relations.

The training and the test set included, respectively, 3054 and 4260 word pairs and they are lexical-split, that is, the two sets do not share any pair. Since words were not tagged, we performed POS-tagging with the TreeTagger (Schmid, 1995).

4 Results

Model	P (task1)	R (task1)	F (task1)	P (task2)	R (task2)	F (task2)
Random Baseline	0.283	0.503	0.362	0.073	0.201	0.106
Cosine Baseline	0.589	0.573	0.581	0.170	0.165	0.167
ROOT18(100)	0.818	0.657	0.729	0.304	0.213	0.249
ROOT18(500)	0.818	0.650	0.724	0.313	**0.227**	**0.262**
ROOT18(1000)	**0.823**	**0.657**	**0.731**	**0.343**	0.218	0.261

Table 1: Precision, Recall and F-measure scores for subtask 1 and 2. The numbers between parentheses in the ROOT18 rows refer to the number of estimators used by the classifier.

As it can be seen from table 1, ROOT18 has a solid performance on the subtask 1, and it is quite accurate in separating related terms from unrelated ones. Generally speaking, we noticed that the classifier performs better when Gini impurity index is used as a splitting criterion instead of entropy. The model with 1000 estimators is our best performing one, with Precision = 0.823, Recall = 0.657 and F-score = 0.731. Concerning the contribution of the features, APSyn1000 and vector cosine have the highest relative importance, with respective contributions of 0.29 and 0.12 to the prediction function. This is not at all surprising, since APSyn and cosine already proved to be strong predictors of semantic similarity.

Relation	Precision	Recall	F-measure
SYN	0.309	0.179	0.226
ANT	0.298	0.206	0.243
HYPER	0.397	0.343	0.368
PART-OF	0.200	0.116	0.147

Table 2: Precision, recall and F-measure for each relation in subtask 2 (ROOT-18 with 500 estimators).

Relation	SYN	ANT	HYPER	PART-OF	RANDOM
SYN	42	29	58	24	82
ANT	29	74	38	23	196
HYPER	32	46	131	30	143
PART-OF	15	43	59	26	81
RANDOM	18	56	44	27	2914

Table 3: Confusion matrix for subtask 2 (ROOT-18 with 500 estimators).

Results are much less convincing for subtask 2. In particular, the recall values are extremely low, especially for some of the semantic relations: part_of, for example, is often below 0.15. For such relation we have no dedicated features in our system, so the difficulty in identifying meronyms are not a surprise. On the other hand, ROOT18 showed the benefits of the inclusion of several measures targeting hypernymy, since the latter is the most accurately recognized relation (precision often > 0.4), recording also the higher recall (always > 0.3, even in the worst performing models).

The performance did not show any particular improvement by increasing the number of the decision trees, so that our best overall results are obtained by the model with 500 estimators (precision = 0.343, recall = 0.218 and F-score = 0.261). As for the contributions of the single features, APSyn1000 (0.19) and cosine (0.09) are still the top ones, followed by cosWeeds (0.07) and APAnt1000 (0.06).

Table 4 describes the confusion matrix, which shows that randoms are properly working as distractors for the model, leading to a large number of misclassification. Synonyms are often confused with hypernyms and this might be due to the fact that the difference between the two is subtle. These results suggest that measures based on the Distributional Inclusion Hypothesis are not always efficient in discriminating between synonyms and hypernyms.

Antonyms are confused with hypernyms and *vice versa*, which might be due to the fact that neither share their most relevant features, obtaining therefore similar APAnt scores (Santus et al., 2015b). Meronyms, finally, are mostly confused with hypernyms, which is almost surely due to the generality spread that characterize both relations and that is captured by both frequency and entropy in our system.

4.1 Conclusions

Our results clearly highlight the difficulty of DSMs in discriminating between several semantic relations at once. Such models, in fact, rely on a vague definition of semantic similarity (i.e. distributional similarity) which does not offer any principled way to distinguish among different types of semantic relations.

Nonetheless, it is still feasible for traditional DSMs to achieve good performances on the recognition of taxonomical relations (Santus et al., 2016a), for which metrics can be defined on the basis of feature inclusion, of context informativeness etc. For other relations, such as antonymy and meronymy, it is not easy to define measures based on distributional similarity (for the latter relation, it is difficult even to find an univocal definition: see Morlane-Hondère (2015)): APAnt works relatively well in discriminating antonyms from synonyms, but – as noticed by Santus et al. (2015b) – this measure has also a bias towards hypernyms, which explains why these are often confused. A possible solution, in our view, would be the integration of DSMs with pattern-based information, in a way that is already being shown by some of the current state-of-the-art systems (see, for example, Shwartz et al. (2016)). Such integration has the advantage of combining the precision of the patterns with the high recall of DSMs.

Finally, we may assume that also the configuration of the original dataset could contribute to our results, since some pairs in the dataset have ambiguous words and the target relations hold for only one of the their meanings. Disambiguating the pairs, at least by Part-Of-Speech, would certainly help in improving the results. A simple method might consist in computing the vector cosine for the pairs with the target words declined in all possible POS (i.e. VV, NN, JJ) and then maintain in the dataset only the pair with the higher value.

5 Acknowledgements

This work has been carried out thanks to the support of the A*MIDEX grant (n ANR-11-IDEX-0001-02) funded by the French Government "Investissements d'Avenir" program.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209-226.

Baroni, Marco, Georgiana Dinu and German Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proceedings of ACL*, Vol (1).

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5-32.

Kenneth Ward Church and Patrick Hanks 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22-29.

Christiane Fellbaum. 1998. WordNet. Wiley Online Library.

Anna Gladkova, Aleksandr Drozd and Satoshi Matsuoka 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't *Proceedings of SRW@HLT-NAACL*

German Kruszewski, Denis Paperno and Marco Baroni. 2015. Deriving Boolean structures from distributional vectors. *TACL*, Vol.3: 375-388

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. *Proceedings of *SEM*.

Omer Levy, Steffen Remus, Chris Biemann and Ido Dagan Do Supervised Distributional Methods Really Learn Lexical Inference Relations? *Proceedings of NAACL HLT*

Omer Levy, Yoav Goldberg and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL, Vol. 3: 211-225*

Dekang Lin. 1998. An information-theoretic definition of similarity. *ICML*, 98:296-304.

Hugo Liu and Push Singh 2004. ConceptNet: a practical commonsense reasoning toolkit. *BT technology journal*, 22(4):211-226.

Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst and Peter D. Turney. 2013. Computing Lexical Contrast. *Computational Linguistics*, Vol. 39(3): 555–590. MIT Press.

François Morlane-Hondère. 2015. What can distributional semantic models tell us about part-of relations? *Proceedings of NetWordS*: 46-50.

Lynne G Murphy. 2003. Semantic relations and the lexicon: Antonymy, synonymy and other paradigms. Cambridge University Press.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction. *Proceedings of ACL*.

Patrick Pantel and Marco Pennacchiotti Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations *Proceedings of COLING ACL*: 113–120

Stephen Roller, Katrin Erk and Gemma Boleda. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. *Proceedings of COLING*: 1025-1036.

Stephen Roller and Katrin Erk. 2016. Relations such as Hypernymy: Identifying and Exploiting Hearst Patterns in Distributional Vectors for Lexical Entailment. *Proceedings of EMNLP*.

Enrico Santus, Qin Lu, Alessandro Lenci and Chu-Ren Huang. 2014. Taking antonymy mask off in vector space. *Proceedings of PACLIC*.

Enrico Santus, Frances Yung, Alessandro Lenci and Chu-Ren Huang. 2015. EVALution 1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. *Proceedings of the ACL Workshop on Linked Data in Linguistics*: 64-69.

Enrico Santus, Alessandro Lenci, Qin Lu and Chu-Ren Huang. 2015. *Italian Journal on Computational Linguistics* aAccademia University Press

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Nine Features in a Random Forest to Learn Taxonomical Semantic Relations. *Proceedings of LREC*.

Enrico Santus, Tin-Shing Chiu, Qin Lu, Alessandro Lenci and Chu-Ren Huang. 2016. What a Nerd! Beating Students and Vector Cosine in the ESL and TOEFL Datasets. *Proceedings of LREC*.

Helmut Schmid. 1995. Treetagger: a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*.

Claude Shannon. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27: 379-423 and 623-656.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *Proceedings of ACL*.

Peter Turney. 2008 A uniform approach to analogies, synonyms, antonyms, and associations *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*: 905-912.

Peter Turney and Patrick Pantel. 2010. From frequency to meaning: Vector Space Models for semantics. *Journal of Artificial Intelligence Research*, 37: 141-188.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. *Proceedings of EMNLP*: 81-88.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David J Weir and Bill Keller. 2014. Learning to Distinguish Hypernyms and Co-Hyponyms. *Proceedings of COLING*: 2249-2259.