Wolves at SemEval-2018 Task 10: Semantic Discrimination based on Knowledge and Association

Shiva Taslimipoor[1], **Omid Rohanian**[1], **Le An Ha**[1], **Gloria Corpas Pastor**[2] **and Ruslan Mitkov**[1]

[1]Research Group in Computational Linguistics, University of Wolverhampton, UK
[2]University of Malaga, Spain
{shiva.taslimi, m.rohanian, l.a.ha, r.mitkov}@wlv.ac.uk
gcorpas@uma.es

Abstract

This paper describes the system submitted to SemEval 2018 shared task 10 'Capturing Discriminative Attributes'. We use a combination of knowledge-based and co-occurrence features to capture the semantic difference between two words in relation to an attribute. We define scores based on association measures, ngram counts, word similarity, and ConceptNet relations. The system is ranked 4th (joint) on the official leaderboard of the task.

1 Introduction

When it comes to investigating semantic similarities, it is worth noting that similarity between two words can be too general to quantify. Accordingly, the discriminating power of a model is also important in limiting the scope of similarity between words.

The main idea behind distributional semantics, known as Distributional Hypothesis (DH), states that linguistic items with similar distributions have similar meanings (Blevins, 2016). Therefore these methods are biased towards finding similarities between concepts. The SemEval shared task 10 'Capturing Discriminative Attributes' poses the new problem of semantic difference detection, thus putting difference, rather than similarity at the forefront. It is about modeling semantic difference in the case of already related words. The idea is that while similarity can group words together in a generic way, understanding semantic differences sheds additional light on the meaning of each individual word.

A semantic model can potentially become more robust if it can benefit from sensitivity to differences alongside similarities in meaning. Considering difference can also help researchers assess semantic representations more rigorously. The effectiveness of a semantic similarity model can be evaluated further by quantifying its strength in finding differences between words.

In the shared task, semantic difference is operationalised as the relation between two semantically related words and a discriminative feature. This relation is realised if the feature characterises only the first word. An example is the triple *airplane*, *helicopter*, *wings*. In this formulation, semantic difference is an asymmetric relation.

In this work, we compute several scores for word pairs and triples with the aim of capturing different semantic relations. Specifically, we define scores based on a knowledge-based ontology and co-occurrence counts. For knowledge-based features we rely on ConceptNet semantic network (Speer and Havasi, 2013), and our co-occurrence based features are derived from association measures, ngrams and pre-trained embeddings. We use the scores in both supervised and unsupervised scenarios to identify triples that constitute semantic difference; i.e. the attribute (third) word is discriminative between the first two words. The code and data used for this system are freely available.[1]

The rest of this paper is organised as follows: Section 2 describes related work. Section 3 provides a description of the approach including the details of the features we use. Sections 4 and 5 discuss experiments and results. Section 6 involves error analysis and some closing remarks and finally the paper concludes with Section 7.

2 Related Work

Distributional similarity methods rely on classical DH, meaning in order to determine how similar two words are, they consider similarity of their contexts. This similarity is usually approximated by taking the cosine of the word vectors. In this

[1]https://github.com/shivaat/discriminative_attribute

way, semantic difference can be modeled as the subtraction of vectors from semantically related words. As a classic example, subtraction of word vectors for *king* and *man* is similar to that of *queen* from *woman* (Mikolov et al., 2013).

However not all semantic differences can be adequately captured using this method. There are many cases where the difference between two words originates from the lack or presence of a feature that cannot be directly mapped to the vector difference between two related words. One such example is *dolphin* and *narwhal* that only differ in having a *horn* (Krebs and Paperno, 2016).

Therefore, combining linguistic and conceptual information would potentially strengthen a semantic model in capturing meaning of a word. To tackle this issue, some studies rely on human annotated list of different attributes related to a concept which are called feature norms (McRae et al., 2005). Despite their strength in encoding semantic knowledge, feature norms have not been widely used in practice because they are small in size and require a lot of work to assemble (Fagarasan et al., 2015). Lazaridou et al. (2016) is an earlier attempt at identification of discriminative features which focuses on visual attributes.

3 Approach

Our goal is to define a simple interpretable metric, using which we can gauge semantic difference and identify discriminative attributes. We hypothesise that for a triple in this task, a stronger relation between the first word and the attribute (in comparison with the second word and the attribute)[2] is indicative of the attribute word being discriminative between the two words.

For each triple we define a discriminative score $Disc\_Score(w1, w2, attr)$ as follows:

$$Disc\_Score(w1, w2, attr) = Score(w1, attr) - Score(w2, attr) \quad (1)$$

where $w1$, $w2$ and $attr$ are the first, second, and third word respectively. $Score$ is a variable function of relation between two words that can be any of the scores explained in Sections 3.1, 3.2, 3.3, and 3.4.

[2]This stronger relation corresponds to more common semantic context and/or higher co-occurrence probability.

3.1 Association-based Score

Statistical association measures have a long history in language processing. With the availability of huge corpora, these measures can be even more effective than before in finding collocations and associations between words.

Collocational behaviour between two words is a strong signal that suggests one of the words can identify the other. As an example, in the triple $(hair, body, curly)$, the association score in $(hair, curly)$ is much more than $(body, curly)$, suggesting that *curly* is a discriminative attribute between the other two words.

For each triple in this task, collocational behaviour of the attribute word with the first two words is measured to see whether the first word can be a better collocate than the other. To this end, we use several different association measures to compute the outputs of the $Score$ function in Eq. 1.

We measure the association of two words based on their co-occurrence in the span of 5 words. We use SketchEngine (Kilgarriff et al., 2004) to extract these statistics from the huge enTenTen corpus (Jakubíček et al., 2013). Specifically, for each pair of words, we extract PMI (Church and Hanks, 1990) (known as MI in SketcEngine), MI3 (Oakes, 1998), log-likelihood (Dunning, 1993), T-score (Krenn and Evert, 2001), log-Dice (Dice, 1945), and Salience (Kilgarriff et al., 2004) all as defined in SketchEngine.

3.2 Google Ngrams

Ngrams are frequently used in computational linguistics for a variety of purposes including language modeling and association measures based on lexical co-occurrence. Google Books Ngram Dataset[3] is a collection of phrases (between 1 and 5 words long) extracted from over 8 million books printed between 1500 and 2008.

We use PhraseFinder (Trenkmann, 2016), a free web API that makes it possible to look up words or phrases from this dataset using a wildcard-supporting query language. Using this resource, we derive two different features. In the first one, we only consider bigrams, and in the other, we consider up to 5-grams. In both cases, we count the number of times that words occur near one another in a span of interest regardless of order. We follow the same formula as defined

[3]https://books.google.com/ngrams

in Eq. 1. In order to eliminate the bias of high/low frequency words we devide $Disc\_Score$ by $Score(w1, attr) + Score(w2, attr)$ that we compute from ngram co-occurrence counts.

3.3 Word Embedding Based Score

In distributional semantics, word embeddings are used to induce meaning representations for words. These methods are inspired by neural network language modeling and have become a basic building block for most applications in computational linguistics. The most popular word embedding method is word2vec (with the skip-gram architecture) which learns dense vector representations for words using an unsupervised model. Word2vec's training objective is based on DH, defined so that the model can learn word vectors that are good at predicting nearby words (Mikolov et al., 2013). Another popular embedding technique is GloVe, which like word2vec, preserves semantic analogies in the vector space. One major difference between the two models is that GloVe utilises corpus statistics by training on global co-occurrence counts rather than local context windows (Pennington et al., 2014).

In our system we use a concatenation of two sets of pre-trained embeddings. The first is trained on English Wikipedia using a variation of word2vec (Bojanowski et al., 2016). The other called ConceptNet Numberbatch (Speer and Lowry-Duda, 2017), is an ensemble of pre-trained Glove and word2vec vectors whose values are readjusted using a technique called retrofitting (Faruqui et al., 2014). In retrofitting, the values of the embeddings are updated using a training function that considers relational knowledge.

Using each word embedding, we compute cosine similarity between each word in a triple and the attribute word to account for the statistics $Score(w1, attr)$ and $Score(w2, attr)$ in Eq. 1.

3.4 ConceptNet Score

Co-occurrence based measures are not sufficient to account for all the various semantic relations that can exist between two words. Knowledge-based ontologies (e.g. ConceptNet, BabelNet etc) encode information about words and their relations in a structured way. This additional source of semantic information can be used to determine whether or not an attribute is discriminative. Because of its free web interface and ease of use, we use ConceptNet to empower our system with relational knowledge (Speer and Havasi, 2013).

For any given $(w1, w2, attr)$ triple, using ConceptNet's REST API we query $w1$, limiting the number of search results to $1,000$. The output is a JSON file that contains all relations between the queried word and other concepts. We traverse all the relations and count the number of times $attr$ is linked to $w1$ to compute $score(w1, attr)$. We repeat the procedure for $w2$ and compute $score(w2, attr)$ and substitute them in Eq. 1.

4 Experimental Settings

We use the data as provided by the organisers of the shared task. We train our model on the train set and find the optimised parameters based on the validations set. Predictions were made on the held-out test data.

The final feature set is the collection of $Disc\_Score$ measures based on the set of proposed scores. As a result we have 6 association-based scores, 2 google ngram based scores, 2 embedding based scores, and 1 ConceptNet score. In total, we have 11 scores as our features.

In ConceptNet, reliability of each relation is given by a weight score. We decided to ignore this information and opted for raw counts because it didn't help performance. Furthermore, binarising the scores based on raw counts (with 0 as a threshold) slightly improved the results.

We use the features in both a supervised scenario (using SVM) and an unsupervised scenario (using KMeans). In both cases all of the 11 features are exploited.

The evaluation in this shared task is in terms of the average of positive and negative F1-scores. In this paper, we report the precision, recall and F1-score for both positive and negative labels separately, along with the average F1-score.

5 Results and Discussion

Table 1 shows the results on the validation set both in the supervised (SVM) and the unsupervised scenario (KMeans).

In this table, we mainly focus on the results that we achieved with our best system after the official evaluation. We also briefly report our official result for $TEST$ as recorded on the shared task leaderboard. The only difference between our system in official evaluation and post evaluation is the setting we have used to extract measures

			Precision	Recall	F1-score	Average F1-score
Validation	SVM	pos	0.7679	0.5652	0.6512	0.6913
		neg	0.6548	0.8284	0.7315	
	KMeans	pos	0.7039	0.6833	0.6935	**0.6972**
		neg	0.6910	0.7113	0.7010	
TEST (Official Evaluation)	SVM					0.69
TEST (Post Evaluation)	SVM	pos	0.7299	0.6065	0.6625	**0.7142**
		neg	0.7197	0.8183	0.7658	
	KMeans	pos	0.6464	0.7001	0.6722	0.6930
		neg	0.7396	0.6899	0.7139	

Table 1: Results on Validation and TEST sets.

from SketchEngine. For the official evaluation, by querying SketchEngine we extracted all the collocations of an attribute word. However, the lists of resulted entries in SketchEngine are limited to a $1,000$ for each query. We later bypassed this limitation by searching for the attribute word with only a limited number of words (from the dataset) in its context. This improves the results on validation and test sets.

Surprisingly, it can be seen in the first part of Table 1 that the unsupervised model (KMeans) can cluster the validation data as well as or even better than the supervised classification approach (SVM).[4] This can be explained by the fact that the features we employ for this task are all computed using a formula that is specifically defined to represent semantic difference, and finding whether a feature is discriminative between two words closely correlates with the semantic difference between them.

It can be concluded from the results that the features are well generalised as they lead to even better performance on the held-out test data.

6 Error Analysis

A sizable portion of the train and test triples bear on genealogical and kinship relations, as in $(grandson, brother, male)$. Some require hierarchical reasoning, as in $(invertebrate, insect, shell)$. Our model captures these kinds of relations very well, as it has access to information from a knowledge base.

In order to see the effectiveness of the scores we obtain from ConceptNet, we re-train the model excluding the ConceptNet based measure and also the vectors derived from Numberbatch embedding. As a result, the validation performance dropped to 0.6857 and the test result decreased to 0.6969 in terms of average F1-score.

A large part of the test triples require the knowledge to understand whether something is a constituent of another entity, as in $(beer, wine, foam)$. It appears that these relations are well captured using co-occurrence based metrics alone since deleting knowledge-base features leaves the results for these triples for the most part unchanged.

7 Conclusions

For this shared task we develop a classification system to determine whether an attribute word can distinguish one word from another. To model semantic difference, we define a discriminative score, and make use of a variety of different association measures derived from huge corpora, and also pre-trained distributional semantic vectors. To augment our method with structured knowledge, we utilise a knowledge-based ontology. We use the feature set in supervised and unsupervised settings. The results suggest that the defined score is capable of generating features that can help our model in capturing instances where a feature is discriminative between two words. Our system shows particular strength in recognising kinship and genealogical relations that are not consistently captured using naive distributional semantic techniques.

In the future, we intend to exploit ConceptNet in a more sophisticated way rather than limiting ourselves to number of relations. It would also be interesting to extract co-occurrence measures from various corpora including domain-specific resources in order to improve the coverage of the model.

[4] In order to evaluate the results from KMeans, we label the clusters in a way that best matches the truth values. These values need to be known to perform this analysis. Therefore, we used SVM for official submission since the TEST data is blind.

References

James P Blevins. 2016. *Word and paradigm morphology*. Oxford University Press.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

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

Lee R. Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *COMPUTATIONAL LINGUISTICS*, 19(1):61–74.

Luana Fagarasan, Eva Maria Vecchi, and Stephen Clark. 2015. From distributional semantics to feature norms: grounding semantic models in human perceptual data. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 52–57.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlỳ, and Vít Suchomel. 2013. The tenten corpus family. In *7th International Corpus Linguistics Conference CL*, pages 125–127.

Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. Itri-04-08 the Sketch Engine. *Information Technology*, 105:116.

Alicia Krebs and Denis Paperno. 2016. Capturing discriminative attributes in a distributional space: Task proposal. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 51–54.

Brigitte Krenn and Stefan Evert. 2001. Can we do better than frequency? a case study on extracting pp-verb collocations. *Proceedings of the ACL Workshop on Collocations*, pages 39–46.

Angeliki Lazaridou, Marco Baroni, et al. 2016. The red one!: On learning to refer to things based on discriminative properties. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 213–218.

Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559.

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.

Michael P. Oakes. 1998. *Statistics for Corpus Linguistics*. Edinburgh University Press.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Robert Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer.

Robert Speer and Joanna Lowry-Duda. 2017. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 85–89.

Martin Trenkmann. 2016. PhraseFinder – Search millions of books for language use. http://phrasefinder.io/. Accessed: 2018-01-30.