SUNNYNLP at SemEval-2018 Task 10: A Support-Vector-Machine-Based Method for Detecting Semantic Difference using Taxonomy and Word Embedding Features

Sunny Lai[1,3], Kwong Sak Leung[1,3] and Yee Leung[2,3]
[1]Department of Computer Science and Engineering
[2]Department of Geography and Resource Management
[3]Institute of Future Cities
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{slai, ksleung}@cse.cuhk.edu.hk, yeeleung@cuhk.edu.hk

Abstract

We present SUNNYNLP, our system for solving SemEval 2018 Task 10: "Capturing Discriminative Attributes". Our Support-Vector-Machine(SVM)-based system combines features extracted from pre-trained embeddings and statistical information from *Is-A* taxonomy to detect semantic difference of concepts pairs. Our system is demonstrated to be effective in detecting semantic difference and is ranked 1^{st} in the competition in terms of F1 measure. The open source of our code is coined SUNNYNLP[1].

1 Introduction

Measuring semantic similarity between words has been a fundamental issue in Natural Language Processing (NLP). Semantic similarity measurements are used to improve downstream applications including paraphrase detection (Xu et al., 2014), question answering (Lin, 2007), taxonomy enrichment (Jurgens and Pilehvar, 2016) and dialogue state tracking (Mrksic et al., 2016).

Despite the current success in using semantic model to measure semantic similarity, lesser attention is paid to teaching machines to make reference (Searle, 1969; Abbott, 2010) to the real world in detecting semantic difference. The semantic difference detection problem can be formalized as a binary classification task: given a triplet (*concept1, concept2, attribute*) which comprises two concepts *(e.g. apple, banana)* and one attribute *(e.g. red)*, determine whether the attribute characterizes the former concept but not the latter. Compared to pairwise semantic similarity detection, this problem is more complex than measuring similarity in general because of its underlying asymmetric property and the extra attribute involved. The SemEval 2018 Task 10 (Krebs et al., 2018) is therefore posed to attract attention to solving this problem.

Although the task of semantic difference detection is novel, similar tasks like referring expression generation (REG) have been studied in the literature. Resources such as ontologies, knowledge bases (Krahmer and Van Deemter, 2012) and images (Kazemzadeh et al., 2014; Lazaridou et al., 2016) are used to learn referring expressions. The major difference between the present task and referring expression is that REG systems can choose salient attributes for making successful reference to objects, while our system is required to decide whether a given attribute can be used to differentiate two similar objects.

The rest of the paper is organized as follows: Section 2 explains our motivation and approach. Section 3 describes the official and external data used. Section 4 details our system implementation. We analyze and discuss the result in Section 5 and conclude our work in Section 6.

2 General Approach

Our approach to this problem is to divide the ternary concept-concept-attribute relationship (*concept1, concept2, attribute*) into two concept-attribute relationships (*concept, attribute*)[2]. The ternary relationship will hold only when the first pair of concept-attribute relation is true and the second false. This approach allows us to use well developed pairwise similarity measurements to extract semantic information from the two concept-attribute pairs, and aggregate the features to train a support vector machine (Cortes and Vapnik, 1995) to detect semantic difference of the triplet, *i.e.* identifying whether a concept contains a specific attribute is a key task of our system.

[1] https://github.com/Yermouth/sunnynlp

[2] For instance, dividing the concept-concept-attribute relationship *(apple, banana, red)* into two concept-attribute relationships: *(apple, red)* and *(banana, red)*.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 741–746
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

Concept-instance example (*Is-A*) from Probase
(***yellow*** food, ***lemon***)
Possible concept-attribute (*Has-A*) pairs
(yellow, food), (food, yellow), (yellow, lemon), ***(lemon, yellow)***, (food, lemon), (lemon, food)
Useful concept-attribute (*Has-A*) pairs
(lemon, yellow)
Semantic difference triplet in official test cases
(***lemon***, cranberry, ***yellow***)

Table 1: Concept-attribute pairs (*Has-A*) can be inferred from concept-instance (*Is-A*) entries in taxonomy, and used to determine whether a semantic difference relationship (*concept1, concept2, attribute*) holds in official test cases.

By observation, we draw similarities between concept-attribute relationship and meronomy *(Has-A)*. They are similar in a sense that both describe subtype relationships. Although linguistics resources constructed by human subjects including norms and priming effect data can help us detect and verify these relationships effectively, they are not allowed to be used in this SemEval Task.

This SemEval Task also limits the scope of concepts and attributes to concrete concepts and visual attributes only. As instances of the same concept are likely to share common attributes from our intuitive perspective[3], we would like to experiment on extracting meronomy (*Has-A*) information from hypernymy (*Is-A*) pairs. Taxonomies and ontologies which contain rich *Is-A* information in terms of concept-instance pairs are therefore the key external linguistic resources which we rely on to extract concept-attribute relationships.

Another intuition that guides our research direction is that modifiers such as adjectives, adverbs and noun modifiers are useful for capturing salient attribute of a specific class of objects[4]. As modifiers are used to describe the scope of concepts or specify context of instances, we can leverage on the co-occurrence probability of modifiers to analyze their dependence/independence relationships with different concepts, and hence, to determine whether a concept-attribute relationship holds.

As the SemEval Task limits the word length of the concept and attribute to be 1, we can enumerate all possible pairs of modifiers and concepts from large scale taxonomy and ontology and use them as features to train our system. Table 1 shows an *Is-A* entry in taxonomies which we find instructive for learning semantic difference relationship. For instance, verifying whether semantic difference relationship holds for the triplet *(lemon, cranberry, yellow)* would require the information of "*lemon* has the attribute *yellow*?" and "*cranberry* does not have the attribute *yellow*?". With the *Is-A* pair *(yellow food, lemon)* from Probase, we can extract possible concept-attribute pairs and their frequency to train our system, such that our system knows with high probability that *lemon* has the attribute *yellow* while *cranberry* does not.

3 Data

We use the official dataset together with two external linguistic resources, GloVe (Pennington et al., 2014) and Probase (Wu et al., 2012; Cheng et al., 2015), to train our system.

3.1 Official Dataset

Official datasets[5] are split into three parts – training, validation and testing, where the testing holds a disjoint attribute sets apart from training and validation. This further increases the difficulty of the task as it prevents *lexical memorization* (Roller et al., 2014; Levy et al., 2015; Weeds et al., 2014) and tests for generalization.

3.2 Probase

Probase is a web scale open domain taxonomy which uses Hearst patterns (Hearst, 1992) to extract *Is-A* relationship from web documents. Each *Is-A* entry in Probase is represented as a triplet form: super-concept, sub-concept and number of co-occurrence. We choose Probase for two main reasons:

1. Large number of concepts covered: The number of concepts covered in Probase (Wu et al., 2012) exceeds other publicly available

[3] As both apple and banana are hypernyms of fruit, i.e. apple *Is-A* fruit and banana *Is-A* fruit. If we know apple is "edible", then banana may have a higher chance of being "edible" by intuition because "edible" can be a common attribute for most fruits.

[4] When we want to differentiate one object from another, we usually use a salient and outstanding attribute to describe the object instead of using a common or similar attribute. Similar viewpoint is previously raised in (Pechmann, 1989; Dale and Haddock, 1991), which states that human beings prefer using efficient and sufficiently distinguishing description when they are constructing referring expressions.

[5] in the form of concept-concept-attribute triplet with human annotated label indicating whether semantic difference exists.

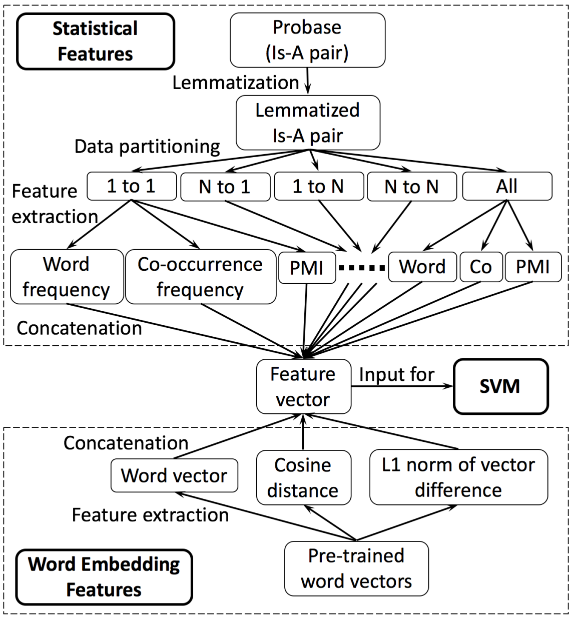

Figure 1: System architecture pipeline diagram.

taxonomies and ontologies including WordNet (Miller, 1995) and YAGO (Suchanek et al., 2007)[6].

2. Rich in semantic features: Probase provides *Is-A* relationship pairs with concepts of different senses and abstraction levels, which allows our system to extract rich statistical information for training. For instance, *Is-A* pairs in Table 2 are extracted from Probase.

3.3 GloVe

Pre-trained embeddings such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2016) encode syntactic and semantic relationships of words in low-dimension space, which is crucial to the capturing of semantic difference. We use the GloVe embedding pre-trained on both Gigaword corpus and 2014 Wikipedia dump in our final submission system.

4 System Description

Our system architecture pipeline (Figure 1) includes the process of data preprocessing, feature extraction and classifier selection.

4.1 Data Preprocessing

Preprocessing procedure is applied to Probase including:

Partition	Concept-Instance(*Is-A*) example	Size
1 to 1	(fruit, banana)	2.12M
N to 1	(high sugar fruit, banana)	7.91M
1 to N	(plant, banana tree)	9.32M
N to N	(dried fruit, banana chip)	14.01M
Total		33.37M

Table 2: Partitioning of dataset into 4 subsets with an example entrand partition size provided.

Frequency Type	Feature Extracted
Individual word	(dried), (fruit), (banana), (chip)
Concept-Concept	(dried, fruit)
Instance-Instance	(banana, chip)
Concept-Instance	(dried, banana), (dried, chip), (fruit, banana), (fruit, chip)

Table 3: Example of how individual word frequency (the first row) and three types of co-occurrences (the last three rows) are counted for the *Is-A* pair (dried fruit, banana chip) in Probase.

- Lemmatization: As Probase is crawled using a rule-based system, we lemmatize the data using stanford CoreNLP (Manning et al., 2014) to reduce words of different forms and allow better matching between *Is-A* entries in taxonomy and official dataset.

- Data partitioning: To give our system additional information regarding the adjectives, adverbs and modifiers of both concepts and instances, we partition Probase into 4 sub datasets, according to the word length of the concept and instance pair. For instance, partition "1 to 1" indicates that both concept and instance are of word length 1. Partition "N to 1" indicates concepts of arbitrary word length (more than 1) and instances of word length 1. Example of each partition are given in Table 2.

4.2 Feature Extraction

4.2.1 Statistical Features

As for statistical features, we consider the statistical features of the individual words, *i.e. concept1, concept2, attribute*, and the two concept-attribute relationship pairs, *i.e.* $(concept1, attribute)$, $(concept2, attribute)$) using individual or co-occurrence frequency in Probase.

Word frequency is extracted from individual words, and the following features are extracted from the *Is-A* pairs:

- Co-occurrence frequency

[6] Probase includes 2,653,872 concepts while WordNet and YAGO contain 25,229 and 352,297, respectively.

Model (Features)	Valid(cv=5)	Train/Test	Train+Valid/Test	Valid/Test
SVM(GloVe, Probase)	**0.790**	**0.644**	0.714	**0.754**
SVM(FastText, Probase)	0.764	**0.649**	0.709	**0.757**
SVM(Word2Vec, Probase)	0.757	0.636	**0.721**	0.732
Logistic Regression(Probase)	0.698	0.602	0.644	0.717
SVM(Probase)	0.730	0.553	0.637	0.691
Logistic Regression(GloVe, Probase)	0.753	0.607	0.674	0.674
SVM(GloVe)	0.712	0.597	0.652	0.668
SVM(FastText)	0.689	0.563	0.593	0.650
SVM(Word2Vec)	0.667	0.556	0.581	0.634

Table 4: Result (F1-score) obtained by our system. The underlined value represents the score of our official submission. Best scores for each partition are denoted in boldface.

- Pointwise Mutual Information(PMI) (Fano, 1961; Church and Hanks, 1990)

- Asymmetric Pointwise Mutual Information(APMI)

There are three types of pairwise word co-occurrence frequencies, including Concept-Concept, Instance-Instance and Concept-Instance. All types of frequencies are calculated for all partitions as distinct features. Table 3 gives an example of how occurrence and co-occurrence are counted. We apply logarithm to the statistical features to reduce the scale of frequently occurring words.

4.2.2 Word Embedding Features

We use the Python package Gensim (Rehurek and Sojka, 2010) to match each word in the triplet (*concept1, concept2, attribute*) in the official dataset with their corresponding pre-trained vectors $v_{con1}, v_{con2}, v_{attr}$, each of 300 dimensions. We then divide the triplet into three pairwise relationships *i.e.* (v_{con1}, v_{con2}), (v_{con1}, v_{attr}), and (v_{con2}, v_{attr}), and calculate the cosine similarity and L1-norm of the vector difference of these pairs as features. Dot-product is considered initially but removed as it adversely affects the performance of our system.

4.3 Classifiers

Using the same set of word embedding and statistical features, we compared the performance of four off-the-shelf classifiers including SVM (Cortes and Vapnik, 1995), Logistic Regression Classifier, Decision Tree Classifier and Random Forest Classifier. SVM classifier with RBF kernel (Vert et al., 2004) is used in our system as it outperforms other classifiers in terms of precision and F1-score.

5 Results and Discussion

5.1 Results

We provide the results of our system with different combinations of features and datasets in Table 4. Column *Train+Valid/Test* represents the F1-score obtained by training our system with both the training partition and validation partition, while column *Train/Test* and *Valid/Test* are F1-score obtained by training our system on the training partition and validation partition individually. Training our SVM system with Probase and GloVe (or FastText) gives the best result in terms of F1-score for official evaluation (column *Valid/Test*). Our system achieves a F1-score of 0.754 and outperforms those of the other teams.

5.2 Discussion

During the competition phase, we noticed that our system performs better when we did not use training partition together with validation partition. As the entries in the training partition are automatically generated, there may be false entries or noise which can adversely affect our system. Since the validation partition comprises manually curated examples, we evaluate our models using 5-fold cross validation on the clean validation partition only (indicated by column *Valid(cv=5)*).

6 Conclusion

In this paper, we have discussed how our simple yet effective SVM system leverages on hypernymy (*Is-A*) relationships and word embeddings to detect single word semantic difference relationship. SVM has been shown useful especially in performing semantic relationship detection tasks (Filice et al., 2016; Panchenko et al., 2016). We would like to extend our system for detecting multiple-words semantic difference relationship, and to broaden the scope of concepts and attributes from visual only to sound and taxonomic.

As our system separates a concept-concept-instance relationship into two concept-instance relationships, our system is relatively weak in capturing attributes that are comparative or fuzzy, for instance, *young* and *tall*. It would be interesting to explore how semantic difference relationship can be embedded into taxonomies, ontologies and vector representations, so that comparative attributes can be comprehensively and directly captured.

Acknowledgments

This research was supported by the Chinese University of Hong Kong - Utrecht University - University of Toronto tripartite collaboration fund of The Chinese University of Hong Kong.

References

B. Abbott. 2010. *Reference*. Oxford Surveys in Semantics & Pragmatics No.2. OUP Oxford.

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

Jianpeng Cheng, Zhongyuan Wang, Ji-Rong Wen, Jun Yan, and Zheng Chen. 2015. Contextual text understanding in distributional semantic space. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 133–142. ACM.

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

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Robert Dale and Nicholas Haddock. 1991. Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4):252–265.

Robert M Fano. 1961. Transmission of information: A statistical theory of communications.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1116–1123, San Diego, California. Association for Computational Linguistics.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

David Jurgens and Mohammad Taher Pilehvar. 2016. Semeval-2016 task 14: Semantic taxonomy enrichment. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1092–1102.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Alicia Krebs, Alessandro Lenci, and Denis Paperno. 2018. Semeval-2018 task 10: Capturing discriminative attributes. In *Proceedings of the 12th international workshop on semantic evaluation (SemEval 2018)*.

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.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976.

Jimmy Lin. 2007. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems (TOIS)*, 25(2):6.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

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.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *CoRR*, abs/1606.03777.

Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cedrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016. Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327, San Diego, California. Association for Computational Linguistics.

Thomas Pechmann. 1989. Incremental speech production and referential overspecification. *Linguistics*, 27(1):89–110.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036.

John R Searle. 1969. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press.

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.

Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. 2004. A primer on kernel methods. *Kernel methods in computational biology*, 47:35–70.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259. Dublin City University and Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.