CogALex-V Shared Task: GHHH - Detecting Semantic Relations via Word Embeddings

Mohammed Attia
Google Inc.
New York City
NY, 10011
attia@google.com

Suraj Maharjan
Dept. of Computer Science
University of Houston
Houston, TX, 77004
smaharjan2@uh.edu

Younes Samih and **Laura Kallmeyer**
Dept. of Computational Linguistics
Heinrich Heine University,
Düsseldorf, Germany
samih,kallmeyer@phil.hhu.de

Thamar Solorio
Dept. of Computer Science
University of Houston
Houston, TX, 77004
solorio@cs.uh.edu

Abstract

This paper describes our system submission to the CogALex-2016 Shared Task on Corpus-Based Identification of Semantic Relations. Our system won first place for Task-1 and second place for Task-2. The evaluation results of our system on the test set is 88.1% (79.0% for TRUE only) f-measure for Task-1 on detecting semantic similarity, and 76.0% (42.3% when excluding RANDOM) for Task-2 on identifying finer-grained semantic relations. In our experiments, we try word analogy, linear regression, and multi-task Convolutional Neural Networks (CNNs) with word embeddings from publicly available word vectors. We found that linear regression performs better in the binary classification (Task-1), while CNNs have better performance in the multi-class semantic classification (Task-2). We assume that word analogy is more suited for deterministic answers rather than handling the ambiguity of one-to-many and many-to-many relationships. We also show that classifier performance could benefit from balancing the distribution of labels in the training data.

1 Introduction

Finding semantic relatedness between words is of crucial importance for natural language processing as it is essential for tasks like query expansion in information retrieval. So far, systems have relied mainly on manually constructed semantic hierarchies, such as ontologies and knowledge graphs. With the recent interest in neural networks and word embeddings, there are attempts to find semantic relations automatically from texts in an arithmetic fashion by measuring the distance between words in the vector space, assuming that words that are similar to each other will tend to have similar contextual embeddings.

This paper describes our system for the CogALex-V Shared Task on Corpus-Based Identification of Semantic Relations. We evaluated three methods for semantic classification based on word embeddings: word analogy, linear regression, and multi-task CNNs. In all these methods, we use publicly available pre-trained English word vectors.

2 Related Work

Semantic relatedness between single words (excluding phrases, sentences and multilingual parallel data) has been addressed in a number of shared tasks before, including relational similarity in SemEval-2012 (Jurgens et al., 2012), word to sense matching in SemEval-2014 (Jurgens et al., 2014), hyponym-hypernym relations in SemEval-2015 (Bordea et al., 2015), semantic taxonomy (hypernymy) in SemEval-2016 (Bordea et al., 2016), and semantic association in CogALex-2014 (Rapp and Zock, 2014).

Proceedings of the Workshop on Cognitive Aspects of the Lexicon,
pages 86–91, Osaka, Japan, December 11-17 2016.

The idea of representing words as vectors has been studied for about three decades (Hinton et al., 1986; Rumelhart et al., 1986; Elman, 1990; Bengio et al., 2003; Kann and Schütze, 2008; Mikolov et al., 2013b). The interest in word embeddings has intensified recently with the introduction of the new log linear architecture of Mikolov et al. (2013a). This architecture provided an efficient and simplified training methodology that minimizes computational complexity by doing away with the non-linear hidden layer, enabling training on much larger data than were previously possible. The public availability of word embedding training programs such as word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) allowed researchers to create models with different parameters and dimensionality sizes for different purposes.

The evaluation data[1] used in the development of the Google Continuous Bag of Words (CBOW) and skip-gram vectors (Mikolov et al., 2013a) focused on semantic similarities and coarse-grained semantic relations in the form of deterministic answers by analogy. These relationships were one-to-one including, for example, capitals (Athens: Greece - Baghdad: Iraq), currencies (India: rupee - Iran: rial), gender (king: queen - man: woman), derivation (amazing: amazingly - safe: safely), and inflection (enhance: enhancing - generate: generating). The evaluation data provided in the CogALex-V Shared Task includes five different semantic relations within the same training and test data, where the relationship between words is one-to-many. For example, while it is relatively easy to predict 'queen' as the answer to this query $x = king - man + woman$, you cannot expect 'contract' as the answer to the query $x = shoe - boot + lease$ with the same level of confidence if the relationship is expected to be either synonymy, antonymy, hyponymy, or hypernymy.

In this paper we try three different methods for handling semantic classification in the shared task: word analogy, linear regression and multi-task CNN. Using word analogy for identifying semantic relations has been discussed in a number of papers including (Levy et al., 2015; Gladkova et al., 2016; Vylomova et al., 2015). The basic idea is to use vector-oriented reasoning based on the offsets between words (Mikolov et al., 2013b) assuming that pairs of words that share a certain semantic relation will have similar cosine distance.

Linear regression classifiers, including Naive Bayes, Logistic Regression and Support Vector Machines, have been used for the identification of semantic relations. For example, GuoDong et al. (2005) used SVM to extract semantic relationships between entities relying on features extracted from lexical, syntactic, and semantic knowledge. Hatzivassiloglou and McKeown (1997) used a log-linear regression model to predict the similarity of conjoined adjectives. Snow et al. (2004) use a logistic regression classifier for hypernym pair identification. Costello (2007) used Naive Bayes to learn associations between features extracted from WordNet and predict relation membership categories. In our work, we do not use any lexical, syntactic or semantic features, other than the word embeddings and we score similarity using the well known cosine similarity metric.

CNNs have also been applied to the task. Zeng et al. (2014) use a convolutional deep neural network (DNN) to extract lexical features learned from word embeddings and then fed into a softmax classifier to predict the relationship between words. Similar approaches have been applied in (Santos et al., 2015) and (Xu et al., 2015).

3 Data Description

3.1 Shared Task Data

The shared task organizers provide a training set of 3,054 word pairs for 318 target words. In Task-1, we are given a pair of words and we need to determine if the words are semantically related or not. Some examples of Task-1 are shown in 1. In Task-2 participants are required to detect the type of the relationship: HYPER, PART_OF, SYN, ANT, or RANDOM.

3.2 Pre-Trained Word Vectors

In our experiments we experimented with three large-scale, publicly available pre-trained word vectors:

[1] http://www.fit.vutbr.cz/ imikolov/rnnlm/word-test.v1.txt

Word 1	Word 2	Task-1: Related?	Task-2: Which Relation?
lease	contract	TRUE	HYPER
brain	head	TRUE	PART_OF
cheat	deceive	TRUE	SYN
move	rest	TRUE	ANT
bright	mature	FALSE	RANDOM
...	...	...	...

Table 1: Training data for Task-1 and Task-2.

Task	Prec.	Rec.	F1
Task-1	75.3	61.1	63.0
Task-2	68.1	34.0	42.6

Table 2: Word Analogy results

Google News[2]. This is built with the word2vec architecture from a news corpus of 100B words (3M vocabulary entries) with 300 dimensions, negative sampling, using continuous bag of words and window size of 5.
Common Crawler[3]. This is built with the GloVe architecture from a corpus of 840B words (2.2M vocabulary entries) with 300 dimensions, and applying the adaptive gradient algorithm (AdaGrad) (Duchi et al., 2011).
Wikipedia + Gigaword 5[4]. This is built with the GloVe architecture from a corpus of 6B words (400K vocabulary entries) with 300 dimensions, and applying AdaGrad with context size of 20.

4 Experiments and Results

In this section we outline the experiments and report the results for the three approaches we tested: word analogy, linear regression and multi-task CNN. The results reported in this section are on the training set for all labels including "FALSE" for Task-1 and "RANDOM" for Task-2. Results on the test set of our selected systems are reported in Section 5.

4.1 Word Analogy

In word analogy, similar to Levy et al. (2014), we query the word vector directly to obtain the closest match to the given example using the formula: $predicted\_word = example\_word1 - example\_word2 + target\_word$. We iterate the query over all the examples in the training set and limit the search scope to the vocabulary items within the set (a set is the target word and all potentially related words). Then we take the average of the responses. The results in Table 2 show that this approach does not work as well for this current task. As we will show, the scores are much lower than those of the other approaches we explored here.

4.2 Linear Regression

We extract similarity distance between words from word vectors, then we use a number of ML classifiers to detect labels based only on the numerical value of the similarity distance. In the initial stage, Table 3, we compare ML algorithms (using 10-fold cross validation) trained on the similarity cosine distance extracted from Google News vectors as the only feature.

We notice from Table 3 that Simple Logistic and Multi-task CNNs have the best score for Task-1 and Task-2 respectively. Now we compare the performance on the three word vector resources: Google News, Common Crawler and Wikipedia + Gigaword 5. Table 4 shows that the best results are obtained by Common-Crawler for Task-2, and by combining the similarity scores from two models of Google News and Wiki+Gigaword for Task-1. We combined them by feeding into the classifier the cosine distance from each word embedding as a feature.

We observe that the classes in the training data are highly imbalanced, where 27% of the pairs are related, while 73% are unrelated. We assume that this disproportion could bias the classifier to prefer

[2]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/
[3]http://nlp.stanford.edu/data/glove.6B.zip
[4]http://nlp.stanford.edu/data/glove.840B.300d.zip

Classifier	Task-1	Task-2
Logistic Regression	77.2	56.6
Simple Logistic	**89.0**	70.1
Decision Trees (J48)	87.7	61.5
NaiveBayes	88.5	**77.4**
LazyIBk	83.5	74.0
LazyKStar	87.8	70.1
Single task CNN	81.8	75.3
Multi-task CNN	83.2	**77.4**

Table 3: F1 Score (%) comparison of ML classifiers.

Classifier (Word Vectors)	Task-1	Task-2
Simple Logistic (G)	89.0	70.1
Simple Logistic (WG)	86.6	75.5
Simple (CC)	89.0	76.0
Simple Logistic (G+WG)	**89.4**	77.2
NaiveBayes (CC)	86.9	76.6
NaiveBayes (G+WG)	88.7	77.8
Multi-task CNN (G)	83.2	77.4
Multi-task CNN (WG)	85.1	78.0
Multi-task CNN (CC)	86.0	**78.4**

Table 4: Comparison of word vectors (G=Google News, WG=Wikipedia+Gigaword and CC=Common Crawler).

Limit	TRUE	FALSE	Average	Diff
1	91.8	79.5	88.3	12.3
2	89.1	86.5	88.1	2.6
3	86.6	89.1	88.0	**2.5**
4	83.6	90.1	87.5	6.5
5	82.2	91.4	88.2	9.2
No limit	79.3	93.1	89.4	13.8

Table 5: Results for different limits of unrelated pairs.

Method	Task-1	Task-2
SimpleLogistic	**79.0**	28.7
Multi-task CNN	71.0	**42.3**

Table 6: Final F1 Scores (%) on the test set.

the majority labels over the minority ones. We try to correct this imbalanced distribution by reducing the number of unrelated pairs and see if this can improve the performance of the classifiers. We conduct our experiments using our best model so far for Task-1 (SimpleLogistic) over different limits of the unrelated words (1, 2, 3, 4, 5 and all) as shown in Table 5. We choose limit 3 as our best model as it has the smallest difference between the f-score for TRUE and FALSE. For Task-2, reduction of unrelated words did not lead to any improvement in the system, so we apply it only to Task-1.

4.3 Multi-task Convolution Neural Network (CNN)

The CNN architecture is similar to the one used by Collobert and Weston (2008). We first feed the pair of input words to the embedding layer, which is initialized with the pre-trained embeddings discussed in Section 3.2. Next in the model is a stack of convolution modules with 500 filters each for filter sizes 1 and 2. We then apply 1-MaxPooling operation, after which we have a Dense layer with 32 neurons. Finally, we have two softmax classifiers since our system uses a multitask approach to jointly learn both tasks. More precisely, the loss function L combines the loss for Task-1 and Task-2, as defined in Equation 1. Here, y_i^{task1}, y_j^{task2}, $\hat{y}_i^{task1}$ and $\hat{y}_j^{task2}$ represent the labels and prediction probabilities for Task-1 and Task-2 respectively. Multitask architectures are preferred over single task ones as the constituent tasks can act as regularizers (Ian Goodfellow and Courville, 2016). There are dropouts after Embedding, Convolution and Dense layers to regularize the network.

$$L(X, Y) = -\sum_i \left(y_i^{task1} \ln \hat{y}_i^{task1} \right) - \sum_j \left(y_j^{task2} \ln \hat{y}_j^{task2} \right) \quad (1)$$

Parameter tuning: We used 20% of the training data as parameter tuning dataset and used it to tune various hyper-parameters like dropout ranges, filters and filter sizes of CNN modules and learning rate. We then use the best model's parameters to perform 10-fold cross-validation experiments with the training data. The results are shown in Table 3 and 4. Additionally, we also experimented with models specific to either Task-1 or Task-2. The results show that the multi-task setting yields better performance than the single task setting.

Label	Precision	Recall	F-Score
RND	87.4	91.1	89.2
SYN	20.9	20.0	20.4
ANT	47.8	42.2	44.8
HYP	50.6	47.6	49.1
PRT	57.6	43.8	49.7
All	75.6	76.7	76.0

Table 7: Detailed results for Task-2 labels.

Label	RND	SYN	ANT	HYP	PRT
RND	2787	85	78	78	31
SYN	86	47	50	41	11
ANT	132	39	152	26	11
HYP	112	42	27	182	19
PRT	70	12	11	33	98

Table 8: Confusion Matrix for Task-2.

5 Final Results

In order to preserve the integrity of the test data, we do not apply any fine-tuning or measure performance improvement by iterating on the test set. We apply only our best performing systems on the training data, which are Simple Logistic trained on Google News and Wikipedia + Gigaword 5 for Task-1, and CNN for Task-2. The results are reported by the shared task evaluation script for the related pairs only (i.e. excluding 'FALSE' and 'RANDOM') and are shown in Table 6. We achieve 79.0% and 42.3% F-score For Task-1 and Task-2 respectively. Tables 7 and 8 present the detailed performance per label in Task-2 and the confusion matrix. We notice that synonyms are the hardest to distinguish among all other labels. This is reminiscent of the philosophical question of the non-existence of exact synonyms (Carstairs-McCarthy, 1994). By contrast, the system performs best in detecting hypernym and part-of relations.

6 Conclusion

In this paper we have presented our systems for identifying and classifying semantic relations between single words. We used linear regression trained only on the cosine distance between word embedding representations. This method gives better results for Task-1. For task2, multi-task CNN method performs better. Our system performs relatively well for the binary classification of similarity between pairs of words, but the performance significantly decreases for the multi-class classification of four semantic relations. This is probably due to the ambiguity in one-to-many and many-to-many relationships.

References

Y. Bengio, R. Ducharme, and P. Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, Colorado.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Andrew Carstairs-McCarthy. 1994. Inflection classes, gender, and the principle of contrast. *Language*, pages 737–788.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Fintan J Costello. 2007. Ucd-fc: Deducing semantic relations using wordnet senses that occur frequently in a database of noun-noun compounds. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 370–373. Association for Computational Linguistics.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

J. Elman. 1990. Finding structure in time. *Cognitive Science*, 14:179–211.

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. In *Proceedings of naacl-hlt*, pages 8–15.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics.

G.E. Hinton, J.L. McClelland, and D.E. Rumelhart. 1986. Distributed representations. *In: Parallel distributed processing: Explorations in the microstructure of cognition*, 1.

Yoshua Bengio Ian Goodfellow and Aaron Courville. 2016. Deep learning. Book in preparation for MIT Press.

David A. Jurgens, Saif M. Mohammad, Peter D. Turney, and Keith J. Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *First Joint Conference on Lexical and Computational Semantics (SEM)*, pages 356–364, Montreal, Canada.

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. 2014. Semeval-2014 task 3: Cross-level semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 17–26, Dublin, Ireland.

Katharina Kann and Hinrich Schütze. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. 2014. Linguistic regularities in sparse and explicit word representations. In *CoNLL*, pages 171–180.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *In Proceedings of International Conference on Learning Representations (ICLR) 2013. arXiv:1301.3781v3*, pages 746–751, Scottsdale, AZ.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT 2013*, pages 746–751, Atlanta, Georgia.

Jeffrey Pennington, Richard Socher, and Christopher D. 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, Doha, Qatar.

Reinhard Rapp and Michael Zock. 2014. The cogalex-iv shared task on the lexical access problem. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon*, pages 1–14, Dublin, Ireland.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Learning internal representations by backpropagating errors. *Nature. 323:533.536*.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems (NIPS 2004)*, November. This is a draft version from the NIPS preproceedings; the final version will be published by April 2005.

Ekaterina Vylomova, Laura Rimmel, Trevor Cohn, and Timothy Baldwin. 2015. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. *arXiv preprint arXiv:1509.01692*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.