CogALex-V Shared Task: CGSRC - Classifying Semantic Relations using Convolutional Neural Networks

Chinnappa Guggilla
chinna.guggilla@gmail.com

Abstract

In this paper, we describe a system (CGSRC) for classifying four semantic relations: synonym, hypernym, antonym and meronym using convolutional neural networks (CNN). We have participated in CogALex-V semantic shared task of corpus-based identification of semantic relations. Proposed approach using CNN-based deep neural networks leveraging pre-compiled word2vec distributional neural embeddings achieved 43.15% weighted-F1 accuracy on subtask-1 (checking existence of a relation between two terms) and 25.24% weighted-F1 accuracy on subtask-2 (classifying relation types).

1 Introduction

Discovering semantic relations and the corresponding relation types between word pairs is an important task in Natural Language Processing (NLP) with a wide range of applications, such as automatic Machine Translation, Question Answering Systems, Ontology Learning, Paraphrase Generation, etc. Corpus-driven automated methods for semantic relation identification have been promising an efficient and scalable solution in the recent past.

To discover semantic relations such as synonym, hypernym and antonym, most of the existing methods (Hearst, 1992; Snow et al., 2004) employed lexical patterns or distributional hypothesis, and suffer from sparsity and low accuracy problems. Moreover, many of these methodologies model individual semantic relations using external knowledge sources such as thesauri, WordNet, etc. Although semantic networks like WordNet[1] define semantic relations such as synonym, hypernym, antonym and part-of between word types, however they are limited in scope and domain.

Recently, few approaches based on distributional word embeddings (Shwartz et al., 2016; Baroni et al., 2012; Ono et al., 2015; Leeuwenberg et al., 2016) reported significant improvements in identifying various lexical semantic relations such as hypernymy, antonymy, synonymy etc. Distributional representations of words learned from a large corpus capture linguistic regularities and collapse similar words into groups (Mikolov et al., 2013b).

Inspired by these approaches, we propose a lexical semantic relation detection system using CNN-based deep neural networks by leveraging word2vec[2] distributional word embeddings as part of 5th edition of CogALex shared task . The shared task proposed two subtasks namely, relation detection and relation type identification. Subtask-1 aims at detecting a relation between two given terms and subtask-2 aims at identifying semantic relations such as synonym, hypernym, antonym, and part-of between two terms if a relation exists. This task is particularly challenging as local context for term pairs is not available in the training corpus.

The rest of the paper is organized as follows: in section 2, we describe related work and in section 3, we introduce deep learning-based supervised classification technique for identifying semantic relations. We describe datasets and the experimental results in section 4. In section 5, we analyze various types of errors in relation classification and conclude the paper.

[1] http://wordnetweb.princeton.edu/perl/webwn/

[2] https://code.google.com/archive/p/word2vec/

Proceedings of the Workshop on Cognitive Aspects of the Lexicon,
pages 104–109, Osaka, Japan, December 11-17 2016.

2 Related Work

For discovering semantic relations between term pairs, several researchers have employed various methods such as pattern-based, distributional, unsupervised and supervised approaches. Several methods that have been developed for synonym extraction employed distributional hypothesis (Saveski and Trajkovski, 2010; Pak et al., 2015) approach. Van der Plas and Tiedemann (2006) combined distributional word similarity, and word-alignment context for synonym extraction in Dutch.

More recently, Leeuwenberg et al. (2016) proposed minimally supervised synonym extraction approach based on neural word embeddings that are compiled using continuous bag-of-words model (CBoW) and the skip-gram model (SG). They analyzed word categories that are similar in the vector space using various combinations of similarity measures with part of speech (POS) information for extracting synonyms from the corpus.

Shwartz et al. (2016) proposed an integrated approach based on deep neural networks by combining path-based and distributional methods for hypernymy detection. Initially, authors experimented with path-based model using dependency paths as embedding features and reported good improvement over prior path-based methods and comparable performance with the superior distributional methods. Later, they extended deep neural networks with distributed signals and showed significant improvement over state-of-the-art approaches. Our proposed approach is similar to this approach in employing deep neural networks and uses word2vec embeddings instead of dependency-based embeddings and also models other semantic relations synonymy, meronymy and antonymy along with hypernymy relation.

Most of the existing approaches (Yih et al., 2012; Zhang et al., 2014) for antonym extraction leveraged thesauri information for distinguishing antonyms from synonyms. Ono et al. (2015) proposed a word embedding-based approach using supervised synonym and antonym information from thesauri, and distributional information from large-scale unlabeled text data and reported improved results.

Shoemaker and Ganapathi (2005) system for automatically discovering meronyms (part-whole) from text corpora using supervised SVM classifier based on empirical distribution over dependency relations as features. vor der Brück and Helbig (2010) proposed semantic-oriented approach for meronymy relation extraction based on semantic networks using automated theorem prover.

3 Methodology

Deep neural networks, with or without word embeddings, have recently shown significant improvements over traditional machine learning–based approaches when applied to various sentence- and relation-level classification tasks.

Kim (2014) have shown that CNNs outperform traditional machine learning-based approaches on several tasks, such as sentiment classification, question type classification, etc. using simple static word embeddings and tuning of hyper-parameters. Zhou et al. (2016) proposed attention-based bi-directional LSTM networks for relation classification task. More recently, (Shwartz et al., 2016) proposed LSTM-based integrated approach by combining path-based and distributional methods for hypernymy detection and shown significant accuracy improvements.

3.1 CNN-based Relation Classification

Following Kim (2014), we present a variant of the CNN architecture with four layer types: an input layer, a convolution layer, a max pooling layer, and a fully connected softmax layer for term pair relation classification as shown in figure 1. Each term pair (sentence) in the input layer is represented as a sentence(relation) comprised of distributional word embeddings. Let $v_i \in \mathbb{R}^k$ be the k-dimensional word vector corresponding to the ith word in the term pair. Then a term pair S of length ℓ is represented as the concatenation of its word vectors:

$$S = v_1 \oplus v_2 \oplus \cdots \oplus v_\ell. \tag{1}$$

In the convolution layer, for a given word sequence within a term pair, a convolutional word filter P is defined. Then, the filter P is applied to each word in the sentence to produce a new set of features. We use

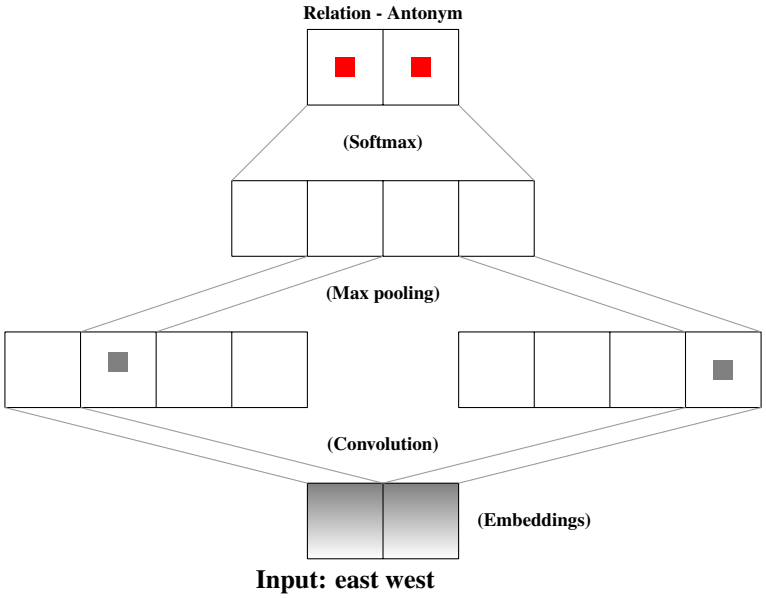

Figure 1: Illustration of an example term pair relation classification using convolutional neural networks

a non-linear activation function such as rectified linear unit (ReLU) for the convolution process and max-over-time pooling (Collobert et al., 2011; Kim, 2014) at pooling layer to deal with the variable sentence size. After a series of convolutions with different filters with different heights, the most important features are generated. Then, this feature representation, Z, is passed to a fully connected penultimate layer and outputs a distribution over different relation labels:

$$y = \text{softmax}(W \cdot Z + b), \tag{2}$$

where y denotes a distribution over different relation labels, W is the weight vector learned from the input word embeddings from the training corpus, and b is the bias term.

3.2 Experimental Setup

We model the relation classification as a sentence classification task. We use the CogALex-V 2016 shared task dataset in our experiments which is described in the next section. This dataset consisting of term pairs is tokenized using white space tokenizer. We performed both binary and multi-class classification on the given data set containing two binary and five multi-class relations from subtask-1 and subtask-2 respectively. We used Kim's (2014) Theano implementation of CNN[3] for training the CNN model. We use word embeddings from word2vec which are learned using the skipgram model of Mikolov et. al (2013a,b) by predicting linear context words surrounding the target words. These word vectors are trained on about 100 billion words from Google News corpus. As word embeddings alone have shown good performance in various classification tasks, we also use them in isolation, with varying dimensions, in our experiment. We performed 10-fold cross-validation (CV) on the entire training set for both the subtasks in random and word2vec embedding settings. We initialized random embeddings in the range of $[-0.25, 0.25]$. We did not use any external corpus for training our model but used pre-compiled word2vec embeddings trained on about 100 billion words from Google News corpus. We used a stochastic gradient descent-based optimization method for minimizing the cross entropy loss during the training with the Rectified Linear Unit (ReLU) non-linear activation function.

Tuning Hyper Parameters. The hyper-parameters we varied are the drop-out, batch size, embedding dimension and hidden node sizes for training our models in cross-validation setting for finding

[3] https://github.com/yoonkim/CNN_sentence

	TRUE	FALSE	Total
Train	826	2228	3054
Test	1201	3059	4260
Total	2027	5287	7314

Table 1: Training and test data sets: Subtask-1 of CogALex-V shared task

	ant	hyper	part_of	random	syn	Total
Train	241	255	163	2228	167	3054
Test	360	382	224	3059	235	4260
Total	601	637	387	5287	402	7314

Table 2: Training and test data sets: Subtask-2 of CogALex-V shared task

CNN	Relation	Precision	Recall	F1
random emb.	true	41.23	37.26	39.00
	false	77.45	80.26	78.80
weighted		41.23	37.26	**39.00**
word2vec emb.	true	58.15	52.69	54.99
	false	82.94	85.96	84.37
weighted		58.15	52.69	**54.99**

Table 3: Avg. 10-fold cross-validation results on subtask-2 with rand. & word2vec embeds.

	Relation	Precision	Recall	F1
CNN	true	35.21	55.70	43.15
	flase	77.46	59.76	67.47
weighted		35.21	55.70	**43.15**
Rand.baseline	true	28.33	50.29	36.24
	flase	71.95	50.05	59.03
weighted		8.33	50.29	**36.24**

Table 4: Subtask-1 test set results word2vec embedding setting Vs. Random baseline.

the optimal model using training set. We performed grid search over these value ranges for the mentioned hyper parameters: drop out{0.1,0.2,0.3,0.4,0.5,0.6}, batch size{12,24,32,48,60}, embedding dimension{50,100,150,200,250,300} and hidden node sizes{100,200,300,400,500}. Optimal results are obtained using drop out-0.5, batch size-32,embedding size-300 and hidden node size-300 for subtask-1 and dropout-0.5, batchSize-24, embedding size-300 and hidden node size-400 for subtask-2 in cross validation setting as shown in tables 3 and 5. We used fixed context-window sizes set at [1,2] as max length of the term pair in given corpus is 2 for both the tasks. We also used fixed number of 25 iterations with default learning rate (0.95) for training our models.

4 Datasets and Evaluation Results

In this section, we describe CogALex-V 2016 shared task data sets and the experimental results.

Datasets. We used a dataset extracted from EVALution 1.0 (Santus et al., 2015), which was developed from WordNet and ConceptNet, and which was further filtered by native speakers in a CrowdFlower task. This data set is split into training and test sets. The distribution of training and test splits are shown in tables 1 and 2. The samples in the subtask-1 and subtask-2 test set are unbalanced and majority of relation classes are "FALSE" and "random" in the given train and test sets.

Evaluation and Results. We evaluated results on test sets using trained models with the optimal parameters for both the tasks and compared results against random baseline results as shown in tables 4 and 6. CogALex-V shared task results are evaluated using weighted-F1 measure on both the tasks. Weighted F-1 values for all the relations except for "random" relation are computed and reported on subtask-2. On subtask-1, i.e. for relation detection, in the cross-validation setting, it is shown that CNN with word2vec embedding setting performed (16%F1) better than the random embeddings. On test set, CNN with word2vec embeddings outperformed (13%) the random baseline results. On subtask-2, i.e for relation type detection, in the cross-validation setting, it is shown that CNN with word2vec embedding setting performed (14.63%F1) better than the random embeddings learned from the training set. On the test set, CNN with word2vec embeddings outperformed (14.64%) the random baseline results. These results suggest that word2vec-based distributional embeddings significantly contributed in improving the relation classification performance.

CNN	Rel.type	Precision	Recall	F1
random emb.	syn	22.61	16.46	18.01
	ant	17.84	13.09	14.67
	hyper	14.05	32.80	35.63
	part_of	40.87	27.98	32.08
weighted		27.82	20.01	**22.68**
	random	69.65	77.15	73.15
word2vec emb.	syn	21.36	12.30	15.03
	ant	39.10	33.08	35.30
	hyper	51.89	48.89	49.61
	part_of	49.24	39.11	42.63
weighted		42.09	34.93	**37.31**
	random	81.89	87.84	84.66

Table 5: Avg. 10-fold cross-validation results on subtask-2 with rand. & word2vec embeds.

	Rel.type	Precision	Recall	F1
CNN	syn	06.96	13.62	09.21
	ant	20.21	31.39	24.59
	hyper	30.71	40.84	35.06
	part_of	25.20	27.68	26.38
weighted		21.89	30.22	**25.24**
	random	77.40	62.93	69.42
Rand.baseline	syn	05.89	20.85	09.18
	ant	07.77	19.17	11.06
	hyper	08.83	20.42	12.33
	part_of	05.31	20.09	08.40
weighted		07.28	20.07	**10.60**
	random	71.57	18.93	29.94

Table 6: subtask-2 results in word2vec embedding setting vs Random baseline.

		Predicted	
		true	**false**
Actual	**true**	669	532
	false	1231	1828

Table 7: Confusion matrix of subtask-1 test set results

		Predicted				
		random	**syn**	**ant**	**hyper**	**part-of**
Actual	random	1925	336	363	280	155
	syn	118	32	47	30	8
	ant	190	39	113	12	6
	hyper	145	35	31	156	15
	part_of	109	18	5	30	62

Table 8: Confusion matrix of subtask-2 test set results

5 Discussion and Conclusion

We can assess the degree of confusion between various relation classes from the confusion matrix of CNN classification model as shown in tables 7 and 8. On subtask-1, 44% of the term pairs are false-negatives and 40% of the term pairs are reported as false-positives. On subtask-2, the "synonym" relation is mostly confused with the "antonym" and "hypernym" and less confused with the "part_of" relation. We also observe a significant amount of confusion between "part_of" and the "hypernym" relations. The relations– "antonym" and "hypernym" are less confused with the "meronym" relation but both are confused with the "synonym" relation. We also observe that majority of the identified relation classes largely confused with the majority "random" class.

In our proposed approach, our system showed that distributional embeddings learned from the large corpus improve relation classification. There are a number of potential directions to improve relation classification accuracy. One possible future work might be to compile the common vocabulary among most confusing relation classes and for the vocabulary compile embeddings from large, unlabeled relation corpora using neural networks, and encode both syntactic and semantic properties of words in the network representation.

Learning embeddings from sense-annotated larger relation corpus might improve the relation detection and relation-type classification accuracy further. Incorporation of dependency embeddings might also improve the relation classification as syntactic contexts can help in distinguishing different terms for identifying appropriate relation type on subtask-2. As antonyms and synonyms fall on the same side in the vector space due to the frequent co-occurrences in the similar contexts, embeddings learned from extra contexts can also improve the relation-type classification performance.

References

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

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.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Artuur Leeuwenberg, Mihaela Vela, Jon Dehdari, and Josef van Genabith. 2016. A minimally supervised approach for synonym extraction with word embeddings. *The Prague Bulletin of Mathematical Linguistics*, 105(1):111–142.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proc. of NAACL*.

Alexander Alexandrovich Pak, Sergazy Sakenovich Narynov, Arman Serikuly Zharmagambetov, Sholpan Nazarovna Sagyndykova, Zhanat Elubaevna Kenzhebayeva, and Irbulat Turemuratovich. 2015. The method of synonyms extraction from unannotated corpus. In *Digital Information, Networking, and Wireless Communications (DINWC), 2015 Third International Conference on*, pages 1–5. IEEE.

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. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications, Association for Computational Linguistics, Beijing, China*, pages 64–69.

Martin Saveski and Igor Trajkovski. 2010. Automatic construction of wordnets by using machine translation and language modeling. In *13th Multiconference Information Society, Ljubljana, Slovenia*.

Austin Shoemaker and Varun Ganapathi. 2005. Learning to automatically discover meronyms.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.

Lonneke Van der Plas and Jörg Tiedemann. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 866–873. Association for Computational Linguistics.

Tim vor der Brück and Hermann Helbig. 2010. Meronymy extraction using an automated theorem prover. *JLCL*, 25(1):57–81.

Wen-tau Yih, Geoffrey Zweig, and John C Platt. 2012. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1212–1222. Association for Computational Linguistics.

Jingwei Zhang, Jeremy Salwen, Michael R Glass, and Alfio Massimiliano Gliozzo. 2014. Word semantic representations using bayesian probabilistic tensor factorization. In *EMNLP*, pages 1522–1531.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 207.