Discourse Relations Detection via a Mixed Generative-Discriminative Framework

Jifan Chen, Qi Zhang, Pengfei Liu, Xuanjing Huang
Shanghai Key Laboratory of Data Science
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, P.R.China
{jfchen14, qz, pfliu14, xjhuang}@fudan.edu.cn

Abstract

Word embeddings, which can better capture the fine-grained semantics of words, have proven to be useful for a variety of natural language processing tasks. However, because discourse structures describe the relationships between segments of discourse, word embeddings cannot be directly integrated to perform the task. In this paper, we introduce a mixed generative-discriminative framework, in which we use vector offsets between embeddings of words to represent the semantic relations between text segments and Fisher kernel framework to convert a variable number of vector offsets into a fixed length vector. In order to incorporate the weights of these offsets into the vector, we also propose the Weighted Fisher Vector. Experimental results on two different datasets show that the proposed method without using manually designed features can achieve better performance on recognizing the discourse level relations in most cases.

Introduction

Discourse relations describe how two segments (e.g. clauses, sentences, and larger multi-clause groupings) of discourse are logically connected. These relations can be used to describe the high-level organization of text. Hence, various NLP applications, such as opinion mining (Somasundaran and Wiebe 2009; Heerschop et al. 2011; Taboada et al. 2011), summarization (Thione et al. 2004; Cristea, Postolache, and Pistol 2005), essay quality analysis (Attali and Burstein 2006), and event detection (Huang and Riloff 2012), can benefit from it.

Along with the increasing requirements, the discourse relation classification and discourse parsing tasks have received considerable attention in recent years. Existing researches have been conducted from different perspectives, including rich linguistic features (Soricut and Marcu 2003; Subba and Di Eugenio 2009; Feng and Hirst 2012), rule based methods (Polanyi et al. 2004), statistical methods (Baldridge and Lascarides 2005; Duverle and Prendinger 2009; Lin, Kan, and Ng 2009; Muller et al. 2012; Li et al. 2014; Ji and Eisenstein 2015), and deep learning based methods (Li, Li, and Hovy 2014). Because there is no discourse-level grammar analogous to sentence-level grammar, discourse relations are less straightforward to define and capture than sentence-level parsing. Most of the works mentioned above treated the task as a supervised classification problem and used linguistic features relating to words and other syntax-related cues to perform the task.

Recently, methods for learning continuous word representations have succeeded in capturing semantic and syntactic regularities using vector arithmetic (Pennington, Socher, and Manning 2014). Mikolov et al. (2013) introduced an interesting observation about word analogies. For example:

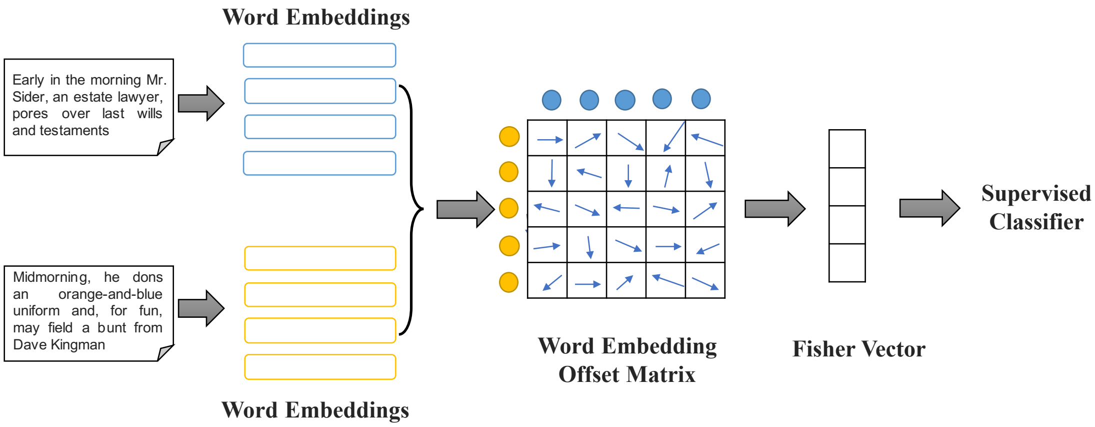
$$v(\text{king}) - v(\text{queen}) \approx v(\text{man}) - v(\text{woman})$$

$v(\cdot)$ denotes the embedding of a word. This indicates that vector offsets in embedding space can represent the shared semantic relations between word pairs. Many existing works also show that hidden relation between words can be represented by the vector arithmetic (Pennington, Socher, and Manning 2014; Fu et al. 2014). Thus, it motivates us to assume that offsets between embeddings of words in a pair of text segments can represent their relevant semantic relations.

In this paper, we introduce a method based on the idea of using vector offsets between word embeddings for discourse relation extraction. Each word in a discourse segment is first embedded into a d-dimensional vector space by a looking-up word embeddings table. Word embeddings can be learned in advance by a feed-forward neural network language model (Bengio et al. 2006), continuous skip-gram model (Mikolov et al. 2013), or other methods. Then vector offsets between word embeddings in segment pairs are calculated. As the number of words is variable in different segments, we propose to use the Fisher kernel framework (Jaakkola, Haussler, and others 1999) to aggregate these vector offsets into a fixed length vector. Finally, supervised methods are used to model the task based on the fixed length vectors.

The main contributions of this work can be summarized as follows:

- We proposed to use vector offsets between word embeddings to represent semantic relations between sentences or segments.


Early in the morning Mr. Sider, an estate lawyer, pores over last wills and testaments

Midmorning, he dons an orange-and-blue uniform and, for fun, may field a bunt from Dave Kingman

Figure 1: The processing framework of the proposed approach.

- The Fisher kernel framework is incorporated to convert a variable number of vector offsets into a fixed length vector, and in order to incorporate the weights of these offsets into the vector, we also propose the Weighted Fisher Vector.
- Experimental results on two datasets show that the proposed method can achieve comparable performance with the state-of-the-art methods using rich linguistic features.

The Proposed Approach

Inspired by the observations in word analogy of word embeddings, we in this work assume that vector offsets between word embeddings in each pair of text segments can represent the semantic relations between them. The processing flow of the proposed approach is shown in Fig. 1. Given a pair of text segments, first, through a lookup table, each word in the pair of text segments is represented by its corresponding word embedding. Then, the vector offsets between all of the word embeddings in the two text segments are computed. These vector offsets compose a *word embedding offset matrix*. Since the size of the matrix is depended on the lengths of the two text segments, it can not be directly used for supervised methods. Hence, we then use Fisher kernel framework to aggregate them into fixed-length vectors. Finally, we use a supervised classifier to predict the discourse relation based on the generated vectors. In the following of this section, we will illustrate the details of these steps of the proposed framework.

The Word Embedding Offset Matrix

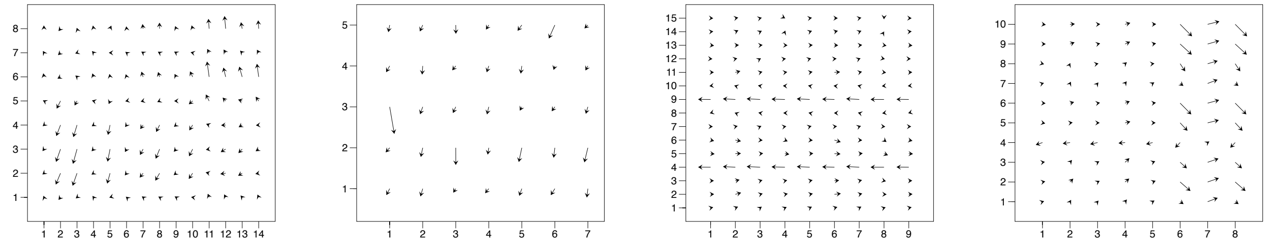
Distributed word representations (word embeddings) are usually designed to capture the attributional similarities between words, which is defined by Turney (2006). It means that words with the same context will be close in the embeddings spaces. Recently, various works also demonstrated that vector offset between word embeddings can present the hidden semantic relations between words. Based on these observations, in this work, we propose to use offsets between embeddings of words in a pair of text segments to learn the relations between them.

A text segment s with length N in a corpus D can be represented as a word sequence $w_1, w_2, ...w_N$. Through a lookup table T, s can be transformed to a sequence of word embeddings $e_s = e_{w_1}, e_{w_2}, ...e_{w_N}$ with each word in s been mapped into a d-dimensional vector space. Suppose there are two text segments s_p, s_q with length m and n, in order to construct an embedding offset matrix M, we first convert them to two embedding sequences e_{s_p} and e_{s_q}. Then we compute the vector offsets between all word embeddings in the two segments, these vector offsets fill an offsets matrix M, where M is a $m \times n$ matrix and $M = \{o_{ij} | o_{ij} = e_{w_i} - e_{w_j}, 0 \leq i \leq m, 0 \leq j \leq n\}$, e_{w_i} and e_{w_j} are the ith and jth word embeddings in e_{s_p} and e_{s_q}. As the number of words are variable in different segments, the size of the embedding offset matrix (EOM) is also different.

Fig. 2 visualizes four word embeddings offset matrixes, which are constructed based on the examples given in the PDTB annotation manual (Prasad et al. 2007). Fig. 2 (a) and (b) show examples about CONTINGENCY relation. Fig. 2 (c) and (d) show examples about COMPARISON relation. From the these examples, we can see that a large percent of offsets in Fig. 2 (a) and (b) point to the bottom-left directions. While, many vectors offsets in Fig. 2 (c) and (d) point to the right directions, except for the offsets related to the stop words (such as *to*, *the*, et al.). We can see that the EOMs constructed based on the sentence pairs with the same relations are similar with each other and EOMs of different relations are different.

Fisher Vector

Given an embedding offset matrix M, since we don't focus on the order of those vector offsets in the matrix, then M can be treated as a bag of vector offsets $M = \{o_t, 1 \leq t \leq N\}$, where N represents the size of M. We assume that the generation process of M can be modeled by a probability density function u_λ with parameter λ. Then the vector offset matrix M can be characterized using the following score



(a) CONTINGENCY: (S1) The Sunnyvale chip maker is worried about blackouts, (S2)a sudden surge or drop in electric power could ruin integrated circuits being built.

(b) CONTINGENCY: (S1) This is not the case. (S2) Some diaries simply aren't worth snooping in.

(c) COMPARISON: (S1) The drug seems to suppress ovulation for three to seven months after it is taken. (S2) Some women clearly have no trouble eventually conceiving again

(d) COMPARISON: (S1) The company intends to pay dividends from available cash flow, (S2) the amount may vary from quarter to quarter.

Figure 2: Examples of 2-dimensional PCA projections of word embedding offset matrixes.

function:

$$G_\lambda^M = \nabla_\lambda \log u_\lambda(M), \tag{1}$$

where G_λ^M is a vector whose size is only depended on the number of parameters in λ, not on the number of offset in the matrix. The gradient describes the contribution of each individual parameters to the generative process. In other words, it describes how the parameters of the generative model u_λ should be modified to better fit the data. The fisher kernel on these gradient is (Jaakkola, Diekhans, and Haussler 1999):

$$K_{FK}(M, \hat{M}) = G_\lambda^{M'} F_\lambda^{-1} G_\lambda^{\hat{M}}, \tag{2}$$

where F_λ is the Fisher information matrix of u_λ:

$$F_\lambda = E_{M \sim u_\lambda}\left[G_\lambda^M G_\lambda^{M'}\right]. \tag{3}$$

Since F_λ is symmetric and positive definite, it has a Cholesky decomposition $F_\lambda = L'_\lambda L_\lambda$, and $K_{FK}(M, \hat{M})$ can be rewritten as a dot-product between normalized vector $\mathscr{G}$ with:

$$\mathscr{G}_\lambda^M = L_\lambda G_\lambda^M = L_\lambda \nabla_\lambda \log u_\lambda(M), \tag{4}$$

where $\mathscr{G}_\lambda^M$ is referred to as the *Fisher Vector* of M.

We follow the work of (Perronnin and Dance 2007), and choose u_λ to be a Gaussian mixture model(GMM): $u_\lambda(x) = \sum_{i=1}^{k} w_i u_i(x)$. Thus $\lambda = \{w_i, \mu_i, \Sigma_i, 1 \leq i \leq K\}$, where w_i, μ_i and Σ_i are respectively the mixture weight, mean vector and covariance matrix of Gaussian u_i. We assume that the covariance metrics are diagonal as any distribution can be approximated with an arbitrary precision by a weighted sum of Gaussian with diagonal covariance, we use the notation $\sigma_i^2 = diag(\Sigma_i)$(Perronnin and Dance 2007). The GMM u_λ is trained on the whole set of embedding offset matrix through Maximum Likelihood(ML).

We consider the gradient with respect to the mean and the diagonal covariance matrix (the gradient with respect to the weight parameters brings little additional information). Let D denote the dimensionality of o_t in M, let $\mathscr{G}_{\mu,i}^M$ be the gradient with respect to the mean μ_i and $\mathscr{G}_{\sigma,i}^M$ be the gradient with respect to σ_i of Gaussian i. Mathematical derivations lead to:

$$\mathscr{G}_{\mu,i}^M = \frac{1}{N\sqrt{w_i}} \sum_{t=1}^{N} \gamma_t(i) \left(\frac{o_t - \mu_i}{\sigma_i}\right), \tag{5}$$

$$\mathscr{G}_{\sigma,i}^M = \frac{1}{N\sqrt{2w_i}} \sum_{t=1}^{N} \gamma_t(i) \left[\frac{(o_t - \mu_i)^2}{\sigma_i^2} - 1\right], \tag{6}$$

where $\gamma_t(i)$ is the soft assignment of o_t to Gaussian i, which is also known as the posterior probability or responsibility:

$$\gamma_t(i) = \frac{w_i u_i(o_t)}{\sum_{j=1}^{N} w_j u_j(o_t)}, \tag{7}$$

and where N is the size of offset matrix M, the division and exponentiation of vectors should be understood as term-by-term operations. The final gradient vector $\mathscr{G}_\lambda^M$ is the concatenation of the $\mathscr{G}_{\mu,i}^M$ and $\mathscr{G}_{\sigma,i}^M$ vectors for $i = 1, ..., K$ and is therefore 2KD-dimensional.

Weighted Fisher Vector One weakness of the *Fisher Vector* described above is that when training GMM, the contributions of all the offsets in the Offset Matrix are equal. However, since every word in one text segment has its own weight (e.g. tf-idf), then the offset between different word embeddings in one text segment pair should have different weights to the relation of the pair. Based on this assumption, each offset in the Offset Matrix has its own weight α when training GMM, and the Offset Matrix M then becomes a weighted Matrix $M_w = \{\alpha_t o_t, 1 \leq t \leq N\}$. With these weighted matrices, we evaluate the parameters of GMM and generate fisher vectors as described above, we name these generated vectors *Weighted Fisher Vector*.

Experiment

We evaluated the proposed method on two datasets: the Penn Discourse Treebank (Miltsakaki et al. 2004) and explanatory relations in product reviews (Zhang et al. 2013).

Table 1: The performances of different approaches on the PDTB. “FV” represents our approach using *Fisher Vector*, and “WFV” represents our approach using *Weighted Fisher Vector*. “ADD” is additive vector composition and “PWM” is point-wise multiplicative vector composition (Mitchell and Lapata 2010), “RAE” is (Socher et al. 2011)’s recursive auto-encoder mentioned above. *CON* means to use the concatenation of compositional text segment vectors as features, *SUB* denotes using the subtraction of compositional text segment vectors as features.

	Comparison	Contingency	Expansion	Temporal
(Pitler, Louis, and Nenkova 2009)	21.96%	47.13%	76.42%	16.76%
(Zhou et al. 2010)	31.79%	47.16%	70.11%	20.30%
(Park and Cardie 2012)	31.32%	49.82%	79.22%	26.57%
(McKeown and Biran 2013)	25.4%	46.94%	75.87%	20.23%
(Ji and Eisenstein 2015)	**35.93**%	52.78%	80.02%	**27.63**%
ADD+CON	26.58%	40.03%	69.72%	12.03%
PWM+CON	25.01%	41.31%	66.03%	14.28%
RAE+CON	18.83%	44.49%	71.96%	13.31%
ADD+SUB	26.30%	39.52%	67.42%	11.18%
PWM+SUB	24.09%	41.56%	65.87%	11.60%
RAE+SUB	19.46%	43.69%	69.88%	12.32%
FV	29.75%	51.86%	80.50%	18.28%
WFV	30.21%	**53.57%**	**80.90%**	20.24%

Implicit Discourse Relation Detection with Penn Discourse Treebank

Experiment Protocols The dataset we used in this work is Penn Discourse Treebank 2.0 (Prasad et al. 2008), which is one of the largest available annotated corpora of discourse relations. It contains 40,600 relations, which are manually annotated from the same 2,312 Wall Street Journal (WSJ) articles as the Penn Treebank. We followed the recommended section partition of PDTB 2.0, which is to use sections 2-20 for training and sections 21-22 for testing (Prasad et al. 2008). For comparison with the work of Pitler et al. (2009), Zhou et al. (2010), Mckeown et al. (2013), and Ji (2015) we trained four binary classifiers to identify each of the top level relations. For each classifier, we used an equal number of positive and negative samples as training data, because each of the relations except *Expansion* is infrequent (Pitler, Louis, and Nenkova 2009). The negative samples were chosen randomly from training sections 2-20. In our experiment, due to the high cost of computing word embeddings, we used the embeddings trained by us on the WSJ Corpus as well as the publicly available embeddings provided by Collobert et al. (2011)[1], Turian et al. (2010)[2], Mikolov (2012)[3] and Mikolov et al. (2013)[4].

We used a 10-fold cross-validation of the training set to select the optimal word embeddings as well as the number of Gaussian densities in the Gaussian Mixture Model (GMM). 300-dimensional vectors pre-trained by Mikolov (2013) achieve the best performance. The optimal number of Gaussian densities in GMM is 16. As for the weights in *Weighted Fisher Vector*, it is reported in the work of Pitler et al.(2009) that the nouns, verbs and adjectives in the pair contribute more to the detection of its relation. In this experiment, we simply set the weight of the offset between nouns, verbs and adjectives to 2, and the others to 1. For the binary classifier, we trained a Random Forest Classifier based on the Fisher Vectors.

For comparing with the proposed method, we also conducted an experiment in which we used the other methods to combine word embeddings of the two text segments to compose their text segments embeddings. These methods are widely used to capture syntactic and semantic meanings of text segments (Mitchell and Lapata 2010). We obtained the text segments vector from its word embeddings by using the following methods:

- **Vector Addition (ADD)**: It was defined as constructing the text segment vector by simply sum the word embeddings in that text segment (Mitchell and Lapata 2010).

$$SentVec(+) = \sum_{\forall e_{w_i} \in sent} e_{w_i} \qquad (8)$$

- **Point-wise Multiplication (PWM)**: Mitchell and Lapata (2010) proposed to construct the text segment vector by using point-wise multiplication for every word embedding in that text segment.

$$SentVec(\odot) = \prod_{\forall e_{w_i} \in sent} e_{w_i} \qquad (9)$$

- **Recursive Autoencoder (RAE)**: Socher (2011) used the parser tree of a sentence as the basis for a RAE. The aim is to construct a vector representation for the tree’s root bottom-up where the leaves contain word vectors.

[1] http://ml.nec-labs.com/senna/

[2] http://metaoptimize.com/projects/wordreprs/

[3] http://rnnlm.org/

[4] https://code.google.com/p/word2vec/

We then performed either concatenation or subtraction on the two text segments embeddings to generate a new vector. After that, we trained supervised classifiers to predict the discourse relations based on the generated vectors.

Results The performances of the four binary classifiers on the top level relations are shown in Table 1. The first highlight for this table is that our approach achieved better performance than previous methods on Contingency and Expansion relations as well as achieved a comparable result on Comparison and Temporal. This proves our assumption that vector offsets between word embeddings in each discourse segment represent the semantic and syntactic meanings of the discourse segments. Also, setting different weight to some important offsets can obviously improve the performance. Furthermore, compared with previous works (Pitler et al. (2009), Zhou et al. (2010), etc.), which used either a lot of complex textual features and contextual information about the two text segments or a larger unannotated corpus to do the prediction, the proposed approach is quite simple and elegant. We only used the information of the two text segments themselves, no complex features and contextual information are needed. We do not even require parsing of the two text segments. With so little information required, we still achieved even better results on the same dataset than previous works did, thus showing that our method is powerful in modeling discourse relations. We can also observe that the performance on temporal relation is not so good as other relations, we believe it is mainly because the training samples of temporal relation are much less than other relations, maybe those samples are inadequate to train our model.

The results of using compositional text segment vectors on the four top relations are also shown in Table 2. As can be seen from the table, each of these compositional methods has its own strengths and weaknesses. For example, RAE (Socher et al. 2011) performs much better than other compositional methods on the $Contingency$ relation, but it has a weaker performance on the $Comparison$ relation. Also, using concatenation often gets better results than subtraction. In general, the results based on text segment vectors are less than satisfactory, and the performance of our approach far exceeds these results. One possible explanation for this phenomenon is much of the word analogy information cannot be held when constructing the text segment vectors. It demonstrated that the proposed word embedding offset matrix has carried as much information of the word analogy as possible, so that it can represent the semantic relations between two text segments.

Parameter Sensitivity Finally, we conducted another experiment to show how the hyperparameters (i.e. the kind of word embedding and the number of Gaussian densities in GMM) affect the effectiveness of our proposed method. In that experiment, we first fixed the number of Gaussian densities and change the word embedding used in our method. Then we fixed the word embedding and modified the number of Gaussian densities so that we could see how the hyperparameter alone affects the effectiveness of the proposed method.

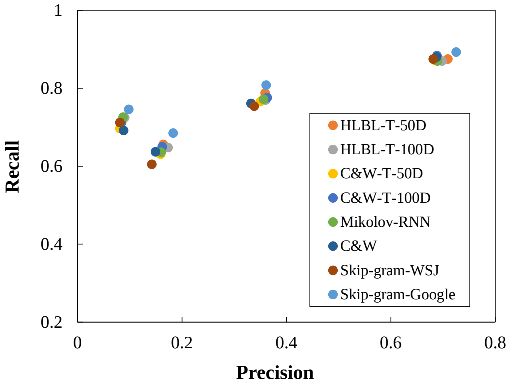


Figure 3: The Precision and Recall on the four top level relations of all the embeddings mentioned above. The number of Gaussian densities is fixed to 16.

Fig. 3 shows the results of using different word embeddings. From the figure, we can observe that although different embeddings were used, the points of the same relation gathered together into four clusters which correspond to the top four relations. In each cluster, the points are very close to each other, which means the Precision and Recall are almost the same under different word embeddings. Taking a deeper look at each cluster, we observe that the skip-gram-Google embedding get slightly better performance than other embeddings, whereas the performance of the Skip-Gram-WSJ and C&W embeddings were less than satisfactory. We believe this is mainly because the Skip-Gram-Google embedding was trained on Google News, which is one of the largest corpora used for embedding training, whereas the Sip-Gram-WSJ embedding was trained by us using a corpus much smaller than all of the other embeddings used. The C&W embedding was trained on the Wikipedia corpus, which is quite different from the news corpus, so it is reasonable for its unsatisfied results on PDTB.

In summary, when fixing the number of Gaussian densities in GMM, the change of word embeddings has minor effect on the performance of our proposed method; all of the embeddings can achieve fairly good performance. Also, training the word embedding on a large corpus may help improve performance.

Fig. 4 illustrates the results via the number of Gaussian densities K, which is used for Fisher kernel. From the figure, we can find that with the increasing number of Gaussian densities, the four curves fluctuate little. One possible explanation for this phenomenon is once the number of Gaussian densities is enough to model the vector offsets, this hyperparameter will have little effect on the performance of our proposed method. Based on the experiments given above, we conclude that the hyperparameters for our proposed method are very easy to choose; no special skills or empirical knowledge are needed.

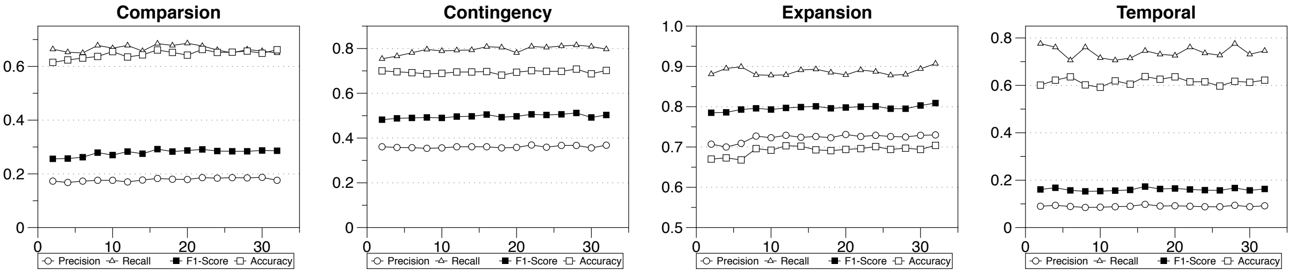


Figure 4: The Precision, Recall, F-score and Accuracy curves of different number of Gaussian densities on the four top level relations, with word embedding fixed to Skip-Gram-Google Embedding.

Explanatory Relation in Product Reviews

The explanatory relation dataset (Zhang et al. 2013) contains a number of reviews about digital cameras crawled from Buzzillions[5], which is a product review site and contains more than 16 million reviews. It contains 1,137 sentences, which are composed of 1,665 clauses. 694 clauses are labeled subjective, and 478 clauses explain other ones. More than 56.1% opinion expressions are explained by their corresponding explanatory sentences. Given this dataset, the aim is to decide whether the opinion clause and nearby clauses hold an explanatory relation or not. To make our results comparable to Zhang et al. (2013), we followed the protocol they used to divide the dataset (i.e. we used 80% of the reviews as training set and the others as test set). All the hyperparameters are the same as we used in the last experiment.

We illustrate the results of the proposed method and the results achieved by Zhang et al. (2013) using other methods in Table 2. Since we used the same training and testing data, we listed the results reported in their literature. From the results, we observe that the proposed method achieved significant improvements over all of the other previous methods. We achieved 19.00% absolute improvement (33.92% relative improvement) over the previous best accuracy and 15.80% absolute improvement (24.88% relative improvement) over the previous best F1-score. Such dramatic improvements show that our proposed approach is effective in modeling the discourse level relations.

In summary, we can see that the proposed method achieved satisfactory results on both datasets, showing that our method is not designed for a specific dataset; instead, it has great abilities of generalization. Moreover, the hyperparameters used for the two datasets are also same. It shows that the proposed method can be easily adopted for other tasks.

Table 2: Performance comparisons between the proposed method and other state-of-the-art methods implemented by Zhang et al. (2013)

Methods	Accuracy	F1
RAE-Subj+PDTB-Rel	28.5%	32.8%
RAE-Subj+SVM-Rel	32.4%	47.6%
MLN	56.2%	63.5%
ADD+CON	67.6%	68.7%
PWM+CON	62.3%	60.5%
RAE+CON	69.4%	68.8%
ADD+SUB	61.0%	61.5%
PWM+SUB	57.5%	58.0%
RAE+SUB	62.0%	63.9%
FV	73.4%	77.4%
WFV	**75.2%**	**79.3%**

Conclusions

In this work, we introduced a novel method to model the relations between discourse level relations between text segments. Motivated by the observation of offsets between word embeddings, we proposed to use vector offsets between words in the embedding space. Since the length of text segments is different, the offsets between word embeddings cannot be directly integrated to perform the task. We incorporated the Fisher kernel framework to convert a variable number of vector offsets into a fixed length vector, and in order to incorporate the weights of these offsets into the vector, we also propose the Weighted Fisher Vector. To demonstrated the effectiveness of the proposed method, we evaluated it on two different datasets. Experimental results demonstrate that the performances of the proposed method are better than pervious best results and other representation methods for text segment in most cases.

Acknowledgement

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 61473092, 61472088, and 61532011), the National High Technology Research and Development Program of China (No. 2015AA015408), and Shanghai Science and Technology Development Funds (13dz2260200, 13511504300).

[5] www.buzzillions.com

References

Attali, Y., and Burstein, J. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment* 4(3).

Baldridge, J., and Lascarides, A. 2005. Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*.

Bengio, Y.; Schwenk, H.; Senécal, J.-S.; Morin, F.; and Gauvain, J.-L. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*. 137–186.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

Cristea, D.; Postolache, O.; and Pistol, I. 2005. Summarisation through discourse structure. In *Computational Linguistics and Intelligent Text Processing*.

Duverle, D. A., and Prendinger, H. 2009. A novel discourse parser based on support vector machine classification. In *Proceedings of the ACL-IJCNLP*, 665–673.

Feng, V. W., and Hirst, G. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of ACL*.

Fu, R.; Guo, J.; Qin, B.; Che, W.; Wang, H.; and Liu, T. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of ACL*.

Heerschop, B.; Goossen, F.; Hogenboom, A.; Frasincar, F.; Kaymak, U.; and de Jong, F. 2011. Polarity analysis of texts using discourse structure. In *Proceedings of CIKM 2011*.

Huang, R., and Riloff, E. 2012. Modeling textual cohesion for event extraction. In *Proceedings of AAAI 2012*.

Jaakkola, T.; Diekhans, M.; and Haussler, D. 1999. Using the fisher kernel method to detect remote protein homologies. In *ISMB*, volume 99, 149–158.

Jaakkola, T.; Haussler, D.; et al. 1999. Exploiting generative models in discriminative classifiers. *Proceedings of NIPS* 487–493.

Ji, Y., and Eisenstein, J. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics* 3:329–344.

Li, S.; Wang, L.; Cao, Z.; and Li, W. 2014. Text-level discourse dependency parsing. In *Proceedings of ACL*.

Li, J.; Li, R.; and Hovy, E. 2014. Recursive deep models for discourse parsing. In *Proceedings of EMNLP*, 2061–2069.

Lin, Z.; Kan, M.-Y.; and Ng, H. T. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of EMNLP*.

McKeown, K., and Biran, O. 2013. Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of ACL*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111–3119.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, 746–751.

Mikolov, T. 2012. *Statistical language models based on neural networks*. Ph.D. Dissertation, Ph. D. thesis, Brno University of Technology.

Miltsakaki, E.; Prasad, R.; Joshi, A. K.; and Webber, B. L. 2004. The penn discourse treebank. In *LREC*.

Mitchell, J., and Lapata, M. 2010. Composition in distributional models of semantics. *Cognitive science* 34(8):1388–1429.

Muller, P.; Afantenos, S.; Denis, P.; Asher, N.; et al. 2012. Constrained decoding for text-level discourse parsing. In *Proceedings of COLING-24th*.

Park, J., and Cardie, C. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. *Proceedings of EMNLP 2014*.

Perronnin, F., and Dance, C. 2007. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of CVPR'07*.

Pitler, E.; Louis, A.; and Nenkova, A. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of ACL-IJCNLP*.

Polanyi, L.; Culy, C.; Van Den Berg, M.; Thione, G. L.; and Ahn, D. 2004. A rule based approach to discourse parsing. In *Proceedings of SIGDIAL*, volume 4.

Prasad, R.; Miltsakaki, E.; Dinesh, N.; Lee, A.; Joshi, A.; Robaldo, L.; and Webber, B. L. 2007. The penn discourse treebank 2.0 annotation manual.

Prasad, R.; Dinesh, N.; Lee, A.; Miltsakaki, E.; Robaldo, L.; Joshi, A. K.; and Webber, B. L. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

Socher, R.; Huang, E. H.; Pennin, J.; Manning, C. D.; and Ng, A. Y. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, 801–809.

Somasundaran, S., and Wiebe, J. 2009. Recognizing stances in online debates. In *Proceedings of ACL-IJCNLP*.

Soricut, R., and Marcu, D. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings NAACL-2003*.

Subba, R., and Di Eugenio, B. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of NAACL-HLT*.

Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; and Stede, M. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* 37(2):267–307.

Thione, G. L.; Van Den Berg, M.; Polanyi, L.; and Culy, C. 2004. Hybrid text summarization: Combining external relevance measures with structural analysis. In *Proceedings of the ACL-04*.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 384–394.

Turney, P. D. 2006. Similarity of semantic relations. *Computational Linguistics* 32(3):379–416.

Zhang, Q.; Qian, J.; Chen, H.; Kang, J.; and Huang, X. 2013. Discourse level explanatory relation extraction from product reviews using first-order logic. In *Proceedings of EMNLP*.

Zhou, Z.-M.; Xu, Y.; Niu, Z.-Y.; Lan, M.; Su, J.; and Tan, C. L. 2010. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*.