Relation Classification through Substring Representations Using Nonlinear Classifiers

Zhan Jin[*], Chihiro Shibata[†] and Kazuya Tago[‡]

School of Computer Science, Tokyo University of Technology
1404-1 Katakuramachi, Hachioji, Tokyo, JAPAN
Email: {d2113002a2,shibatachh,ktago}@edu.teu.ac.jp

Abstract—Semantic relation classification can be considered as a multiclass classification problem. Richer and higher quality feature sets lead to better performance when using traditional features. This tendency also increases the dimensions of the feature space, resulting in an increased processing time, and leads to lower classification accuracy when using nonlinear classifiers. We introduce an approach to build features for relation classification consisting of only low-dimensional vectors representing substrings between two words, called *substring vectors*. In this paper on substring vectors, we survey the relationship between the numbers of dimensions and the obtained accuracies when nonlinear classifiers are applied. Through experimental results, we found that our approach using relatively low-dimensional representations achieves a sufficiently high accuracy that is better than most existing approaches. Furthermore, we utilize autoencoders for reconstruction and decrease the number of dimensions; finally, we obtain better classification results than before.

1. Introduction

In the past few years, relation classification has attracted considerable research interest. It has widely served an important role in many applications such as machine translation. Although many approaches have been explored for relation classification, the most representative and general one is that of supervised classification, which has been shown to be reliable and yields good classification results in most cases [5][2]. These methods use a set of heuristic features that can effectively represent the relations between two nominals that must be determined after performing a textual analysis. Because the use of richer and higher-quality features leads to better performance for the existing approaches, various features such as part-of-speech (POS) tagging, syntactic patterns, and prepositions are frequently used, and external resources such as WordNet data, Wikipedia data, and Google n-grams are continually imported [1][8][6]. These approaches are effective because they leverage a large body of linguistic knowledge. However, this tendency also increases the dimensions of the feature vector space, resulting in an increased processing time. Thus, it is difficult to simplify the complexity of the features and improve the classification results simultaneously.

Recent research has tended to use distributed representations for words and neural network language models (NNLMs) to solve this problem [9][7]. In our opinion, obtaining an appropriate distributed representation is no less important for achieving highly accurate results and a low computational cost than learning methods over the vectors.

In this paper, we introduce new distributed representations for sequences of words between two words, called *substring vectors*, which have a much lower dimension than the feature vectors used in existing approaches. We reconstruct and decrease the number of dimensions utilizing an autoencoder such as an independent component analysis (ICA) and principal component analysis (PCA). By combining these representations with sophisticated nonlinear classifiers, the relations between pairs of nominals in our approach are classified efficiently. The experimental results demonstrate that our approach achieves a sufficiently high accuracy with a small computational cost.

2. Related Work

Relation classification is one of the most important topics in natural language processing (NLP). The approach of Bryan et al.(2010) won the relation classification contest called SemEval-2010 Task 8 [3] and uses various types of features that can be partitioned into eight groups, where five groups are taken from external resources. This shows that the combination of rich features and learning algorithms that are tolerant to high dimensions, such as a linear support vector machine (SVM), is one of the most effective approaches for relation detection. However, the performance of that approach strongly depends on the quality of the designed features and the amount of external resources.

With the recent revival of interest in deep neural networks (DNNs), many researchers have concentrated on the use of deep learning approaches to learn features. Socher (2012) proposed a new recursive neural network (RNN) to learn vectors for relation classification [7], and Zeng et al.(2014) used a convolutional DNN to extract lexical and sentence-level features [9]. These studies showed that the use of NNLMs improves the classification results more than approaches based on traditional features. Unfortunately, it is difficult to reduce the computational cost while maintaining the prediction accuracy because of the large number of dimensions. Typically, PCA and ICA are often utilized to learn a compressed, distributed representation from input data [4], and they can lower number of dimensions to some extent.

3. Proposed Method

We propose a distributed representation of substrings in text data as well as a method for classifying semantic relations. The advantages of our approach are that only one type of feature is used—*substring vectors*, and the construction method is very simple.

3.1. Preprocessing Input Data

Let $S_1, \cdots, S_M$ be the sentences in the data, and for each i, $S_i = w_{i1} \cdots w_{i|S_i|}$, where w_k represents the k-th word of S_i, and $|S_i|$ represents the length of the S_i. Let the set of all sentences $D = (S_1, \cdots, S_M)$. We assume that each sentence S_i has at most one pair of indices, e_{i1} and e_{i2}, where $w_{ie_{i1}}$ and $w_{ie_{i2}}$ are the pair of words to be classified with respect to the semantic relations. To avoid double subscripts, we let $w(e_{ik})$ denote $w_{ie_{ik}}$ for $k = 1, 2$. If S_i has no such pair of words, we let $e_{i1} = e_{i2} = 0$ and $w(0) = \lambda$, where λ denotes the empty word. Input data are represented by the sequence of triplets $\langle S_1, e_{11}, e_{12} \rangle, \cdots, \langle S_M, e_{M1}, e_{M2} \rangle$. We allow e_{i1} and e_{i2} to be chosen arbitrarily.

Each sentence S_i is divided into three substrings by splitting S_i at e_{i1} and e_{i2}. Let $\texttt{Substr}_{i1}$, $\texttt{Substr}_{i2}$, and $\texttt{Substr}_{i3}$ be these substrings in order. The substring that we mainly map into the vector space is $\texttt{Substr}_{i2}$ because $\texttt{Substr}_{i2}$ is the most informative about the semantic relation between $w(e_{i1})$ and $w(e_{i2})$. For instance, suppose that

$$S_1 = \text{The}_1 \ \text{eye}_2 \ \text{works}_3 \ \text{using}_4 \ \text{the}_5 \ \text{retina}_6 \ \text{as}_7 \ \text{a}_8 \ \text{lens}_9 \ .,$$

$e_{11} = 2$, and $e_{12} = 6$, i.e., $w(e_{11})$ = "eye" and $w(e_{12})$ = "retina". S_1 can be considered a sentence that describes the relations between e_1 and e_2. Then, we have

$$\begin{aligned}
\texttt{Substr}_{11} &= (w_{11}) = (\text{"the"}),\\
\texttt{Substr}_{12} &= (w_{12}, w_{14}, w_{15}) = (\text{"works"}, \text{"using"}, \text{"the"}),\\
\texttt{Substr}_{13} &= (w_{17}, w_{18}, w_{19}) = (\text{"as"}, \text{"a"}, \text{"lens"}),
\end{aligned}$$

and $\texttt{Substr}_{12}$ appears to explain the relation between "eye" and "retina" best among the three.

All sentences in D are used as training data by algorithms that give distributed representations. The word vectors are in $\mathbb{R}^N$, where N is arbitrarily chosen but is usually approximately 10–100. Although we believe that the word vectors have potentially sufficient information about the semantic relations, it is still necessary to introduce another distributed representation for the sentence itself, such as substring vectors. Before that, we can extract important information about the semantic relations from the word vectors utilizing PCA or ICA, which also lowers the number of dimensions and simplifies the process.

3.2. Constructing a Substring Representation

The construction process is shown in Figure 1. First, we create the weight dictionary on the basis of the word frequencies. Then, we use this dictionary for weighting and normalizing each word in the substrings and obtain the substring vectors by averaging them.

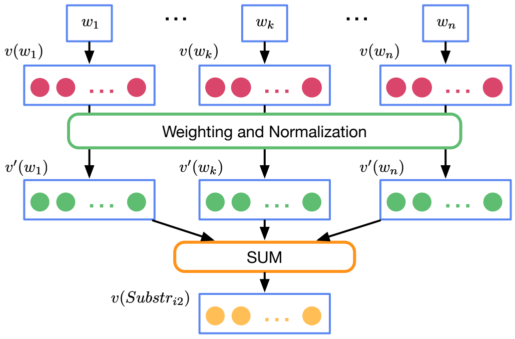

Figure 1: Construction process of substring vectors.

To construct a reasonable representation for a substring from word vectors, we define the weights for each word that represent the degrees of importance of the relations between pairs of nominals. For instance, suppose that a dataset D includes only one sentence S_1, i.e., $D = (S_1)$. Among "works", "using", and "the", which are the elements of $\texttt{Substr}_{12}$, "the" appears in both $\texttt{Substr}_{11}$ and $\texttt{Substr}_{12}$, whereas "works" and "using" only appear in $\texttt{Substr}_{12}$. Thus, we observe that, as compared to "the", "works" and "using" are more informative for the semantic relation between "eye" and "retina". In other words, if a word w mainly appears in $\texttt{Set}_2$, we believe that w frequently represents some semantic relation.

For a word w and substring s, we define $\texttt{Cnt}(w, s)$ as the number of occurrences of w in s. In addition, for a multiset of substrings $\mathcal{S}$, let $\texttt{Cnt}(w, \mathcal{S}) = \sum_{s \in \mathcal{S}} \texttt{Cnt}(w, s)$. The weight for a word w is defined as

$$a(w) = \frac{\texttt{Cnt}(w, \texttt{Set}_2)}{\texttt{Cnt}(w, \texttt{Set}_1) + \texttt{Cnt}(w, \texttt{Set}_2) + \texttt{Cnt}(w, \texttt{Set}_3)}. \tag{1}$$

In order to prevent the length of each substring vector from being too large or small or having the center of gravity for the weighted word vectors, we normalize the weights in Eq. 1 with respect to each substring $\texttt{Substr}_{ij}$:

$$\bar{a}_{ij} = \frac{a(w_{ij})}{\sum_{k=e_{i1}+1}^{e_{i2}-1} a(w_{ik})}. \tag{2}$$

Using the normalized weights and word vectors, we define the substring vectors $v(\texttt{Substr}_{i2})$ for each substring $\texttt{Substr}_{i2}$ as

$$v(\texttt{Substr}_{i2}) = \sum_{k=e_{i1}+1}^{e_{i2-1}} \bar{a}_{ij} v(w_{ij}), \tag{3}$$

where $v(w)$ is the word vector of w.

4. Dataset and Classifiters

To evaluate the performance of our proposed method, we used the SemEval-2010 Task 8 dataset [3]. The dataset is freely available and contains 10,717 annotated examples, including 8,000 training instances and 2,717 test instances. It distinguishes nine semantic directed relations, such as Entity–Origin, Component–Whole, and Cause–Effect. In addition, it has another special undirected relation called Other.

We learn from the training data and obtain F1 scores from the test data for 10 relations (including Other). The average of the F1 scores for nine relations (excluding Other) is called the macro-averaged F1 score. In addition, we remove the instances of Other from the training and test data and obtain the average of the F1 scores for nine relations, which is called micro-averaged F1-score. To compare results of our proposed method, we adopted the macro-averaged F1-score as a measure of the prediction accuracy (same as previous studies). However, we adopted the micro-averaged F1 score in other experiments because the classification results will be more stable when excluding occurrences of Other.

We applied four multiclass classifiers: a random forest (RF), an SVM with the Gaussian radial basis function kernel (SVM-RBF), a polynomial function kernel (SVM-POL), and the linear SVM (SVM-LIN). For the parameter settings of the RF, we let the number of trees be 120. For parameters of the SVMs, we let the cost for incorrect classification be 60.

5. Experiments

In this section, we conduct four sets of experiments. In the first set of experiments, we compare the F1 scores and computing times of three classifiers in different dimensions. The second set of experiments tests the validity of our weighting method. In the third set of experiment, we compare the F1 scores of the polynomial kernel classifier at different degrees. Finally, we reduce the dimensions of the word vectors using PCA and compare the performance of classification in different dimensions.

System Name	micro-averaged F1-score	# of External Resources
ECNU-SR-7	75.21	2
ISI	77.57	3
FBK_IRST_12VBCA	77.62	1
UTD	82.19	5
CDNN	82.7	1
Proposed (RF)	**77.18**	**0**
Proposed (SVM-RBF)	**78.10**	**0**

Table 1: F1 scores of all systems for relation classification.

To evaluate the performance of our proposed method, we compared five methods with our method, as summarized in Table 1. The first four are the best of the existing approaches that are not NNLMs, and the following were proposed in current studies using NNLMs. We demonstrate that our approach significantly outperforms existing approaches [3] that are not NNLMs when using an SVM with Gaussian kernels. Crucially, unlike existing approaches, our method does not use any external resources.

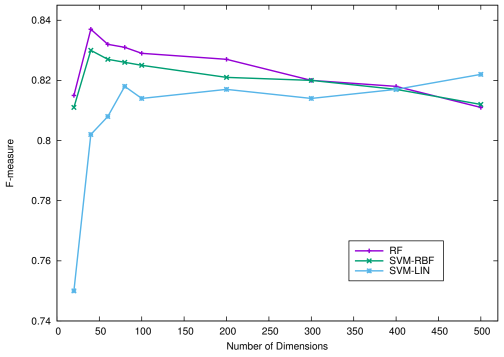

Figure 2: F1 scores for each classifier as a function of the dimension of the substring vectors.

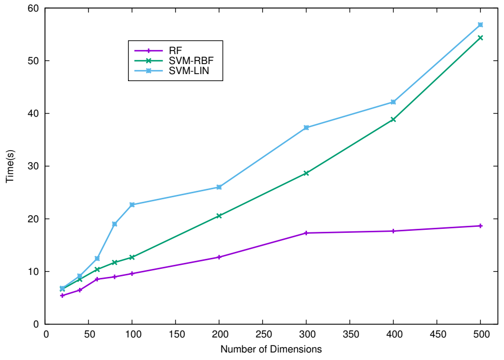

Figure 3: Computing time for each classifier as a function of the dimension of the substring vectors.

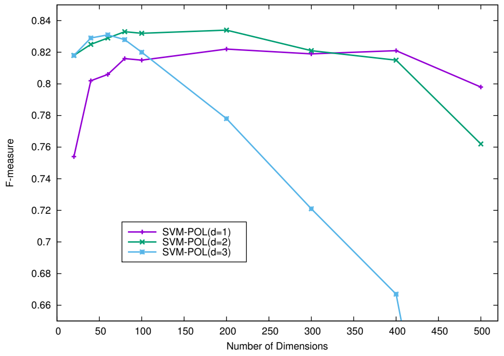

Figure 4: F1 scores for each degree of the polynomial function kernel SVM as a function of the dimension of the substring vectors.

Figures 2 and 3 show the F1 scores and computing time for the first set of experiments. If the dimension of the substring vectors is less than approximately 400, the nonlinear classifiers obtained better classification results than the linear one. However, when the dimension is greater than approximately 400, the linear classifier has better performance than the others. This phenomenon appears to be

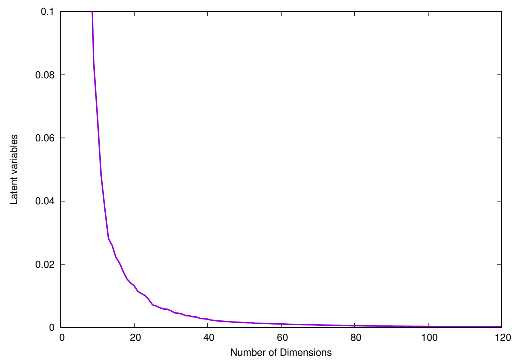

Figure 5: Latent variables for the PCA of the original word vectors.

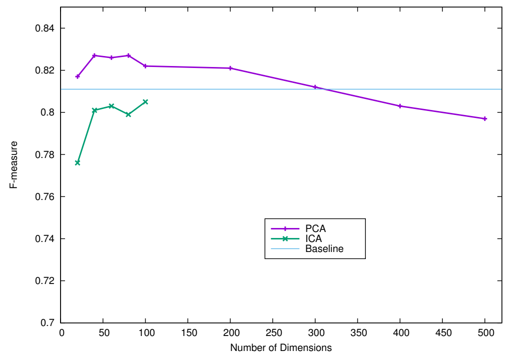

Figure 6: F1 scores for each dimension of the transformation word vectors. The classifier is the RF, and the baseline is F1 scores of the original 500 dimensions.

due to the overfitting caused by the high degree of freedom that the SVM-RBF has as a classifier. This implies that a sufficiently low-dimensional representation of the data is required in order to use nonlinear classifiers efficiently. In the first experiment, we obtained the best classification results using nonlinear classifiers for approximately 50 dimensions of the substring vectors, and the RF obtained better performance than the other classifiers with respect to the computing time.

In the second set of experiments, we verify that our weighting method is effective. After processing the weighting, the F1 scores of the classification results can be improved by 3%–4%. This is because the words that frequently appear in the substrings between pairs of nominals to be classified (e_{i1} and e_{i2}) are expected to carry much information about the semantic relations.

The results of the third set of experiments are shown in Figure 4. We ensure that overfitting occurs more easily for a higher degree of freedom for the classifiers, which is a similar result from the first set of experiments. We have the best combination when the degree of the kernel function is not one and the number of dimensions is approximately 40–60.

In the fourth set of experiments, we reduce the dimensions of the word vectors by PCA and ICA. The latent values for processing are shown in Figure 5. We can understand that most classification information exists within 100 dimensions of the reconstructed vectors. We extract several groups of reconstructed vectors in different dimensions. The classification results of each group are shown in Figure 6. The F1 score of the original 500 dimensions is 0.811 as a baseline for comparison. If the dimension of reconstructed vectors is less than 300, the processed data obtained better classification results than the original 500 dimensions. When the dimension is greater than 300, the F1 scores decreased because overfitting occurs. This experiment also demonstrated that we can improve our substring vectors by PCA.

6. Conclusion

In this paper, we introduce substring vectors, which represent the relationship between pairs of nominals. We show that dimensional reduction of substring vectors largely affects the classification results, especially when the classifiers have a high degree of freedom. The base performance is obtained for relatively lower dimensions, i.e., approximately 40–60, using nonlinear classifiers such as an RF and SVM-RBF. In our experiments, we found that our approach yields better results than almost all of the existing approaches and can be applied to nonlinear classifiers. However, there are many ways to improve our approach, such as the utilization of external resources to optimize the substring vectors.

References

[1] Yuan Chen, Man Lan, Jian Su, Zhi Min Zhou, and Yu Xu. Ecnu: Effective semantic relations classification without complicated features or multiple external corpora. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 226–229, 2010.

[2] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on ACL*, pages 427–434, 2005.

[3] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, 2010.

[4] Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, and Andrew Ng. Building high-level features using large scale unsupervised learning. In *International Conference in Machine Learning*, 2012.

[5] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL*, volume 2, pages 1003–1011, 2009.

[6] Bryan Rink and Sanda Harabagiu. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010.

[7] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Conference on EMNLP*, pages 1201–1211, 2012.

[8] Kateryna Tymoshenko and Claudio Giuliano. Fbk-irst: Semantic relation extraction using cyc. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 214–217, 2010.

[9] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, 2014.