Neural Comput & Applic
DOI 10.1007/s00521-012-1109-9

ORIGINAL ARTICLE

SREC: Discourse-level semantic relation extraction from text

Mohammad-hadi Zahedi · Mohsen Kahani

Received: 7 August 2011 / Accepted: 23 July 2012

Abstract Semantic relation extraction is a significant topic in semantic web and natural language processing with various important applications such as knowledge acquisition, web and text mining, information retrieval and search engine, text classification and summarization. Many approaches such rule base, machine learning and statistical methods have been applied, targeting different types of relation ranging from hyponymy, hypernymy, meronymy, holonymy to domain-specific relation. In this paper, we present a computational method for extraction of explicit and implicit semantic relation from text, by applying statistic and linear algebraic approaches besides syntactic and semantic processing of text.

Keywords Semantic relation extraction · Semantic role labelling · Singular value decomposition

1 Introduction

Text sources and the rapidity information distribution is speedily increasing over time in all modern media such as internet. These phenomena make it complicated and problematic to identify meaningful links between intraday information flows. Many studies have concentrated on information extraction (IE) and relation extraction (RE) in this decade. Researchers have employed several methods from natural language processing (NLP), pattern recognition and classification into similarity measurement to distinguish relevant sources (information) from irrelevant ones and to extract information as well as relation that may be mentioned explicitly or even implicitly. This topic can be seen from different points of view such as computational linguistic, cognitive linguistic, morphology (linguistic). Many applications apply RE technique, such as ontology learning and alignment [1–3], ontology evolution and enrichment [4], IX and retrieval [5–7], text categorization and summarization.

Semantic relations are meaningful associations between two or more entities; these can be mentioned explicitly or implicitly. Associations between entities can be categorized into different types, abstracted or conceptualized. Semantic relations can refer to relations between concepts in the mind (conceptual relations), or relations between words (lexical relations) or text segments [8]. As is said in [9], "When I think in language, there aren't 'meanings' going through my mind in addition to the verbal expressions: the language is itself the vehicle of thought".

Semantic RE in domain-independent text is a challenging task. One problem is the identification of semantic relations that are at the core of NLP and may be latent and concealed among the elements of text such as phrases, sentences, and discourse. In some context, the RE aims to identify the semantic relations between terms in the texts. These include already existing relations between the terms in the knowledge base or completely new (latent) relations. As a result, extra information about terms can be acquired. By identifying new relation in the text and recognizing their features, the approach could also be applied to ontology population. Moreover, since it extracts new relations between terms, it could be applied for ontology learning. In addition, various methods have been presented to compute mappings between elements from different

M. Zahedi (✉) · M. Kahani
Department of Computer Engineering,
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: Zahedi@staff.um.ac.ir

M. Kahani
e-mail: kahani@um.ac.ir

ontologies. In many cases, they fail to achieve salient mappings due to poorly developed or heterogeneous ontology structures. In this case, semantic RE also plays a crucial role [1, 10].

However, entity and relations are inextricably bound with language and text, and it is difficult to analyse the meaning of concepts and relations apart from the language that expresses them. Halliday and Hasan have explained in [11]: "A text is not something that is like a sentence, only bigger; it is something that differs from a sentence in kind. A text is best regarded as a semantic unit: a unit not of form but of meaning".

With this background in our study, we focus on RE from the standpoint of computational linguistic, with emphasis on semantic. We demonstrate the effectiveness of semantic role in RE and indicate how this feature along with syntactic attributes and corpus-based statistical approaches such as tf–idf and singular value decomposition (SVD), which are linear algebra-based methods, will optimize the model performance. The experimental results show that the proposed model can achieve noticeable performance comparable with previous best studied models. Ontology independency, scalability and usability in various applications such as ontology population, text summarization, text classification, question/answering systems and search engines are the strengths of our method.

The rest of paper is organized as follows: in Sect. 2, we have briefly reviewed previous and related works, and Sect. 3 presents the methodological basis of SVD method. Then, we discuss some NLP on text and corpus in Sect. 4, while the proposed model is described in Sect. 5. Section 6 illustrates the experimental results and comparisons of our work with related works. Finally, Sect. 7 completes this paper with a brief conclusion.

2 Related work

Several approaches have been proposed for the relations extraction from text. Some of them rely on the mapping of syntactic dependencies onto semantic relations, such as SVO [12, 13], using either pattern matching [14] or other strategies, such as probabilistic parsing [15], or clustering of semantically similar syntactic dependencies, according to their selection restrictions [16]. The Asium system [17] learns semantic knowledge by using clustering features in the form of sub-categorization frames of verbs. Zelenko et al. [18] used support vector machines with a specialized kernel model based on instances representation and their position in a shallow parse tree, while Zhao et al. [19] used kernel methods and support vector machines to integrate and extend the individual kernels. Reichartz et al. [20] have presented new tree kernels over dependency parse trees automatically generated from natural language text. They also optimized kernel computations to improve the actual runtime. But kernel-based methods suffer from two weak points. First, in some cases, information contained in the SPT is not sufficient to determine relationship. Secondly, the current tree kernels may not be able to adequately capture the structural information in a tree structure. In other words, each sub-tree enumerated in the kernel computation does not consider the contextual information outside the sub-tree. Finally, while it is well known that semantic information plays a crucial role in semantic RE, it has not been well studied in [21–23]. In contrast, Zhou and Zhang in [23] employed diverse lexical, syntactic and semantic knowledge in feature-based RE using support vector machines. Their study demonstrated that the base phrase chunking information contributes to most of the performance improvement from syntactic aspect. Evaluation results show that their feature-based approach significantly outperforms tree kernel-based approaches.

Maedche and Staab [24] proposed an algorithm based on statistical techniques and association rules of data mining technology for detecting relevant relationships between ontological concepts. Imsombut [4] presented a statistical approach for learning the semantic relations between concepts of ontology in the agricultural domain. In [25], mainly statistical methods based on frequency information have been employed over linguistic dependencies in order to establish relations between entities from a corpus of the biomedical domain without labelling the discovered relations. DODDLE [26] extracts the non-taxonomic relationships by analysing the co-occurrence of lexical relations, based on word space, while Kavalec and Svatek [27] proposed the technique for relation labelling by selecting verbs frequently occurring in the context with each concept association. Karoui et al. [3] combined a verb-centred method, lexical analyses, syntactic and statistic ones to extract multi-type relations from the text analyses and the existent relations. Focusing on the particular task of ontology development, RelExt system [28] extracts relevant verbs and their grammatical arguments from a domain-specific text collection and computes corresponding relations through a combination of linguistic and statistical processing. Yan et al. [29] proposed an unsupervised approach that uses linguistic information to reduce surface and noisy surface patterns generated from large corpus, and used Web frequency information to smooth the sparseness of linguistic information.

This paper focuses on studying syntactic and semantic information to determine a more effective method of extracting semantic relations between text semantic units. We will further explore the linear algebra approach with syntactic and semantic information. We also show how semantic information such as semantic role of each term can be equipped to further improve the performance. Evaluation of the ACE corpora shows that our system outperforms other previous methods.

3 Singular value decomposition

Latent semantic analysis (LSA) applied SVD [30] to reveal semantic patterns in textual data. The fundamental idea in LSA explains that linguistic knowledge contains a large number of weak correlations between semantic concepts. LSA claims that using SVD and choosing correct choice of dimensionality can reveal the semantic patterns and latent meaning of the text. SVD was found to be adopted along with measuring similarity or clustering techniques in most domains.

3.1 Structural SVD design

In LSA, a corpus of linguistic data is represented as a word-by-document matrix. Rows represent words and columns represent documents in which words appear. The matrix entries are a measure of the number of times a word appears in a specific document (Fig. 1).

Singular value decomposition takes an occurrence matrix A with m rows and n columns and then decomposes it into a product of three matrices, $A = USV^T$, where $U(m \times m)$ and $V^T(n \times n)$ are the left and right orthogonal matrices and $S(m \times n)$ is a rectangular matrix with non-negative singular values on the diagonal in order of decreasing magnitude. We choose special orthonormal bases $V = (v_1, v_2, \ldots, v_r)$ for the row space and $U = (u_1, u_2, \ldots, u_r)$ for the column space, such that Av_i is in the direction of u_i. s_i provides the scaling factor where $Av_i = s_i u_i$. In matrix form, this becomes $AV = US$ or $A = USV^T$.

3.1.1 Dimension reduction

In the next step, SVD performs a dimensionality reduction over matrix A. It keeps the first k values of S and produces a k-reduced truncated approximation of A as $\dot{A}(m \times n) = \dot{U}(m \times k) * \dot{S}(k \times k) * \dot{V}^T(k \times n)$. Different approximations will be produced for different values of k. The entries in A will change to lower or higher values in $\dot{A}$, producing a linear least squares approximation of matrix A. The singular values preserve the most important associative relationships of the matrix A in decreasing order of magnitude. The principal claim in LSA is that at some optimal dimensionality, the k-approximation will cut out the "noise" or irrelevant relationships and will induce important implicit or latent relationships that exist between the words and documents and cannot be observed directly from the original matrix.

	d_1	d_2	d_3	...	d_n
w_1	n_{11}	n_{12}	n_{13}	...	n_{1n}
w_2	n_{21}	n_{22}	n_{23}	...	n_{2n}
w_3	n_{31}	n_{32}	n_{33}	...	n_{3n}
...	...	...	...	...	...
w_m	n_{w1}	n_{m2}	n_{m3}	...	n_{mn}

Fig. 1 LSA word-by-document example

The complexity of SVD is $O((N + M)^2 \times k^3)$, where N is the number of documents, M is the number of unique words, and k is the number of dimensions in the concept space. According to Kontostathis et al. [31], the value of k can be a small number ranging from 25 to 200 for the small collection and 50–500 for the larger collection, depending on the collection of documents.

3.2 What does SVD do in LSA?

The conceptual explanation for this mathematical process is that linguistic knowledge contains a large number of weak interrelations. Although words are invariant, the way in which they come together in sentences is unique. Thus, the original data matrix is sparse, and this does not adequately capture implicit "meaning". The implicit relations are determined by the SVD and dimensionality reduction process, because of the capacity to employ the linear combination of each matrix entry with all the others in producing the U and the V and then observing this in a reduced number of dimensions. Singular value decomposition creates a "global" space from "local" occurrences of words in documents, where the association strengths of words and documents with each other are a statistical measure of how each one relates to all the others, whether or not they appeared together explicitly in the data set.

4 Natural language processing task

As it was mentioned earlier, we have applied some NLP task over corpus. Parts of them are pre-processing such as sentence splitting, word segmentation and term extraction, stop word elimination and stemming. Then, we use a POS tagger, which is a preliminary and syntactic processing stage in language processing. POS tagger is the process of identifying the parts of speech corresponding to each word in a given text. It is clear that some words may have different parts of speech depending on its location in a sentence; for example, “book” can be a noun (“the book on the table”) or verb (“to book a flight”), or the word “set” can be a noun, verb or adjective.

In the next step, we apply semantic role labelling (SRL), the computational identification and labelling of arguments in text, which consists of, given a sentence, detecting basic event structures such as “who” did “what” to “whom”, “when” and “where”. Typical roles used in SRL are labels such as agent, patient, location, temporal and manner. The first three labels are used for entities participating in an event while the later two show characterization of other aspects of the event [32].

In the final step of this stage, we apply tf–idf weighting (term frequency–inverse document frequency), which is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases in proportional to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents. Till now, various (mathematical) forms of the tf–idf term weight have been proposed, but the general form of tf–idf is as below:

$$\text{tf-idf}_{ij} = \text{tf}_{i,j} \times \text{df}_i \tag{1}$$

5 Method

As we know, semantic meaning is represented explicitly and locally through syntax, as well as implicitly and globally through the same syntax. Therefore, semantic meaning is not addressed completely by syntax; it is also a result of the interaction and correlation of semantic units (entities) as determined by the syntax. The patterns of occurrence (words in sentences) capture inherent “meaning” in natural language. Singular value decomposition manages to bring out implicit (“latent” in the LSA terminology) or explicit empirical syntactic patterns of occurrences of concept or occurrences in their context where the co-occurrence of the concepts (words) themselves defines these sentence (context). In our method, combination of SVD, with dimensionality reductions and text semantic processing task, and similarity measurements and pattern recognition could reveal semantic patterns from the text.

As is said in Hofstadter’s book [33]:

> semantic properties are connected to open-ended searches because, in an important sense, an object’s meaning is not localized within the object itself… Thus, another way of characterizing the difference between “syntactic” and “semantic” properties is that the syntactic ones reside unambiguously inside the object under consideration, whereas semantic properties depend on its relations with a potentially infinite class of other objects, and therefore are not completely localizable. There is nothing cryptic, or hidden, in principle, in syntactic properties, whereas hiddenness is of the essence in semantic properties.

Regarding this definition and respect of linguistic theory, it can be inferred that in addition to words, semantic relations can occur at higher levels of text-between phrases, clauses, sentences and larger text segments, as well as between documents and sets of documents [8]. Important steps of semantic relations extraction are the following:

5.1 Syntactic and semantic task

Some kinds of syntactic task that have been applied in this paper are sentence breaking, word segmentation and term extraction, stop word elimination and stemming. Stemming is a vital step in the process of determining term frequencies to reduce the number of terms. Without stemming, the term frequencies will give deceptive results. The stemmer used in this paper is a version of Porter[1] [34]. Then we use a SNoW-based POS tagger[2] for identifying the parts of speech corresponding to each word in a given text. In the next step, we apply Illinois semantic role labeller[3] for labelling semantic role of each entity. Core of this system is SNoW learning architecture and the relational feature extraction language. Finally, we apply the normalized weight model that helps to increase the equality between terms [35] for calculating tf–idf.

$$w_{ij} = \frac{\text{tf}_{i,j}}{\max_{\text{tf}_{i,j}}} \times \log \frac{N}{\text{df}_i} \tag{2}$$

where $\text{tf}_{i,j}$ is frequency of term i in document j; $\max_{\text{tf}_{i,j}}$ is maximum number among all of tf values in corpus; N is the number of documents; df_i is the number of documents in which term_i occur.

5.2 Create occurrence matrix

The next step is creation of an occurrence matrix. All transformed textual documents are converted to a term matrix. Here, we adopt some changes; instead of creating document–word matrix, we make a sentence–word matrix for each document separately; Also, instead of one matrix, we create two matrices: one for words frequency with POS tag of NN(noun) and V(verb), called PM; another one for words with semantic label of agent, patient and source,

[1] Available at: http://tartarus.org/~martin/PorterStemmer/.

[2] Available at: http://cogcomp.cs.illinois.edu/page/software_view/3.

[3] Available at: http://cogcomp.cs.illinois.edu/page/software_view/12.

called SM. We apply this modification for three main purposes: firstly, to eliminate the terms that have less effect and role in a sentence and document; secondly, to increase the impact of syntactic and semantic roles of terms; finally, to reduce the size of matrix and consequently to increase the computation performance.

The matrix is nothing more than an $N \times M$ table, where N represents the number of sentences, and M is the total number of words remaining after the preparation stage. Each matrix cell contains a number representing the weight of word i in document j. During this step, we use the normalized weight algorithm that was mentioned former.

5.3 Apply singular value decomposition method

The next major step is to apply the SVD algorithm to decompose the created matrices in the order we described in Sect. 3. It is clear that as the k values are further increased and varied, the approximations $\dot{A}$ will be increasingly accurate to the original occurrence matrix A. At $k = r$, the matrix will become the same again, $A = \dot{A}$.

Our observation shows that implicit relationships are returned using the largest singular values; as the k values are increased, the approximations begin to return only the explicit relationships as we expected. This result is relevant to the finding best choice of k. Different k values will bring out different level of relationships between the terms and the sentences, and indicate different implicit relationships.

As mentioned earlier, $AV = US$ and $A^T U = SV$. Let terms (occurrence) space $T = US$ and sentence space $S = SV^T$. Then term space T $(m \times k)$ has m rows, each having k components. Thus, we can rewrite: $T = (t_1, t_2, \ldots, t_m)$, where each $t_i = \{t_{i1}, t_{i2}, \ldots, t_{ik}\}z$. Similarly, $S(k \times n)$ has n columns, each having k components. Thus, we can rewrite: $S = \{s_1, s_1, \ldots, s_{1n}\}$, where each $s_j = (s_{j1}, s_{j2}, \ldots, s_{jk})$.

5.4 Measuring semantic similarity

The next step is computing similarity between pair of sentence–sentence, term–term and sentence–terms, in the k-reduced approximation space to measure the strength of the associative relationship between them. Here in this paper, we use cosine similarity. Cosine similarity incorporates the information provided in the previous stage to create a numerical value to describe the similarity of each pair of compared items. The value itself means little, but a group of such values creates a natural ordering of comparisons in which the highest values are the most similar and the lowest values are the least. The general form of cosine similarity is as below:

$$\mathrm{Cosine}(D_1, D_2) = \Sigma\left(\frac{w_{D_1(j)} * w_{D_2(j)}}{\mathrm{norm}(D_1) * \mathrm{norm}(D_2)}\right) \tag{3}$$

Formally, interaction strength between term i and term j is defined as the cosine between the vectors representing term i and term j in k-dimensional space as follows:

$$C_T = t_i^{\mathrm{T}} t_j / \|t_i\| \|t_j\| \tag{4}$$

Interaction strength between sentence i and sentence j is defined as the cosine between the vectors representing sentence i and sentence j in k-dimensional space as follows:

$$C_S = s_i^{\mathrm{T}} s_j / \|s_i\| \|s_j\| \tag{5}$$

Also, interaction strength between term i and sentence j is defined as the cosine between the vectors representing term i and sentence j in k-dimensional space as in Eq. 7:

$$C_{T-S} = t_i^{\mathrm{T}} s_j / \|t_i\| \|s_j\| \tag{6}$$

5.5 Create similarity matrix

After computing cosine similarity between all pairs of term–term, term–sentence and sentence–sentence, we create three matrices: $X^k_{m\times m}, Y^k_{n\times n}$ and $Z^k_{m\times n}$, which contain all the interaction strength measurements C_T, C_S and C_{T-S}, respectively.

5.6 Similarity matrix combination

As has been mentioned earlier, we applied all the previous stages for both PM and SM matrices. In the next step, we construct combined matrices of both corresponding similarity matrices as follows:

$$C_{TF} = \alpha_1 . C_T^{PM} + \beta_1 \cdot C_T^{SM} \tag{7}$$

$$C_{SF} = \alpha_2 . C_S^{PM} + \beta_2 \cdot C_S^{SM} \tag{8}$$

$$C_{TF} = \alpha_3 . C_T^{PM} + \beta_3 \cdot C_T^{SM} \tag{9}$$

where $\alpha_1 + \beta_1 = \alpha_2 + \beta_2 = \alpha_3 + \beta_{13} = 1$. Setting any one of these factors to 0 means that we do not include that matrix. Setting both of the factors to 0.5 means we consider them equally important.

5.7 Extracting the patterns

In this step, we reorder the similarity matrices in decreasing order by keeping the corresponding rows and columns index. Then, a greedy algorithm is applied to extract blocks from matrices when the entire cell blocks (sub-matrix) have a value higher than a threshold. Each block illustrates patterns that indicate a semantic relation between different parts of text. These approaches have made various patterns. An appropriate pattern should be expressive enough to

The settling companies would also assign their possible claims against the underwriters to the investor plaintiffs.

DT The VBG settling NNS companies MD would RB also VB assign PRP$ their JJ possible NNS claims IN against DT the NNS underwriters TO to DT the NN investor NNS plaintiffs . .

agent, assigner [A0] The settling companies, general modifier [AM-MOD] would, discourse marker [AM-DIS] also, V: assign assign, thing assigned [A1] their possible claims against the underwriters, assigned to [A2] the investor plaintiffs.

Fig. 2 Example of semantic role labeller and POS tagger

represent the information that is to be extracted from text without being overly complex.

5.8 Graph representation

For better visualizing and understanding, the results have been shown in the form of graphs, wherein each node represents terms (or sentences) and edges show the relation between terms (or sentences). Each edge has a label that indicates the strength of relation between two nodes.

5.9 Graph operations

We can apply different functions over created graphs. We can find the number of strongly connected components in the directed graph to show which nodes are strongly connected with each other. It is possible to navigate the reaction of a system when a specific node (terms or sentence) was eliminated. Even we can calculate the shortest paths from each node to all other nodes by specifying a well-defined function cost. Minimal spanning tree could be extracted from the graph. When we have two graphs, each one can show the relation between sentences and terms; we can find isomorphism between the two graphs that may be usable in some application.

6 Experiment and comparison

6.1 Experiment

We mainly use the Microsoft Research Paraphrase Corpus data set, which consists of 5,801 pairs of sentences gleaned over a period of 18 months from thousands of news sources on the web [36]. Accompanying each pair is judgment reflecting whether multiple human annotators considered the two sentences to be close enough in meaning to be considered close paraphrases. We show the execution of the proposed model step by step. First, we applied POS tagger and semantic role labeller over the text. The results are in the form shown in Fig. 2:

Here, A0 is a subject, A1 is an object and A2 is an indirect object in semantic role labeller, DT is a determiner, IN is a preposition, JJ is a adjective, NN is a singular noun, NNS is a plural noun, NNP is a proper singular noun, NNPS is a proper plural noun, PRP is a personal pronoun, RB is a adverb, TO—to, VB is a verb base form, VBD is a verb past tense, VBG is a verb gerund/present participle, VBN is a verb past participle, VBP is a verb non-third ps. sing. Present, VBZ is verb third ps. sing. Present is POS tagger.

Then, we create two occurrence matrices: one for terms with POS tag, called PM matrix, and another one for terms with semantic label, called SM matrices. PM matrix contains terms with NN* and VB* terms only, while SM matrix contains terms with A0, A1, A2 and V label.

Then, we applied SVD and adopt dimension reduction. After several executions, we gain that best value for K (reduces dimension size) is $lt/4$, where lt is the number of sentence in document. After that, we compute the cosine similarity, and X, Y, Z are created. The similarity matrices are reordered by keeping the corresponding row and column index as shown in Fig. 3, which shows an example of X matrix.

In the next step, we try to extract block of sub-matrix where all of its cell have a value higher than the threshold value. For X (term–term) similarity matrix, we choose threshold $\tau = 0.75$, and for Y (sentence–sentence) similarity matrix, we choose threshold $\tau = 0.55$. Figure 4 shows pattern extractions from matrix Y (it can be mentioned that the first row and first column show the index of terms or sentences). These blocks show semantic relations between semantic units of text. For better visualizing, we show the result in the form of graph as shown in Fig. 5.

6.2 Comparison

In all experiments, this paper employed the ACE RDC 2003 provided by the linguistic data consortium.[4] The ACE RDC corpora are gathered from various newspapers, newswires and broadcasts. In the ACE RDC 2003 corpus, the training set consists of 674 documents and 9,683 positive relation instances, while the test set consists of 97 documents and 1,386 positive relation instances. The ACE RDC 2003 corpus defines five entity types, five major relation types (see Table 1) and 24 relation sub-types. All document entities and the relations between them were

[4] http://www.ldc.upenn.edu/.

0	1	7	10	3	9	2	4	266	267	80	77	78	79	76	126	127	128
1	1	0.999726	0.999683	0.999683	0.999683	0.999683	0.969496	0.959627	0.959627	0.942264	0.698108	0.698108	0.698108	0.698108	0.703678	0.703678	0.703678
7	0.999726	1	0.99887	0.99887	0.99887	0.99887	0.973976	0.965818	0.965818	0.949651	0.713776	0.713776	0.713776	0.713776	0.71724	0.71724	0.71724
10	0.999683	0.99887	1	1	1	1	0.96313	0.95241	0.95241	0.933705	0.680228	0.680228	0.680228	0.680228	0.68589	0.68589	0.68589
3	0.999683	0.99887	1	1	1	1	0.96313	0.95241	0.95241	0.933705	0.680228	0.680228	0.680228	0.680228	0.68589	0.68589	0.68589
9	0.999683	0.99887	1	1	1	1	0.96313	0.95241	0.95241	0.933705	0.680228	0.680228	0.680228	0.680228	0.68589	0.68589	0.68589
2	0.999683	0.99887	1	1	1	1	0.96313	0.95241	0.95241	0.933705	0.680228	0.680228	0.680228	0.680228	0.68589	0.68589	0.68589
4	0.969496	0.973976	0.96313	0.96313	0.96313	0.96313	1	0.993851	0.993851	0.989835	0.843019	0.843019	0.843019	0.843019	0.856365	0.856365	0.856365
266	0.959627	0.965818	0.95241	0.95241	0.95241	0.95241	0.993851	1	1	0.998329	0.867999	0.867999	0.867999	0.867999	0.859332	0.859332	0.859332
267	0.959627	0.965818	0.95241	0.95241	0.95241	0.95241	0.993851	1	1	0.998329	0.867999	0.867999	0.867999	0.867999	0.859332	0.859332	0.859332
80	0.942264	0.949651	0.933705	0.933705	0.933705	0.933705	0.989835	0.998329	0.998329	1	0.895238	0.895238	0.895238	0.895238	0.88427	0.88427	0.88427
77	0.698108	0.713776	0.680228	0.680228	0.680228	0.680228	0.843019	0.867999	0.867999	0.895238	1	1	1	1	0.973044	0.973044	0.973044
78	0.698108	0.713776	0.680228	0.680228	0.680228	0.680228	0.843019	0.867999	0.867999	0.895238	1	1	1	1	0.973044	0.973044	0.973044
79	0.698108	0.713776	0.680228	0.680228	0.680228	0.680228	0.843019	0.867999	0.867999	0.895238	1	1	1	1	0.973044	0.973044	0.973044
76	0.698108	0.713776	0.680228	0.680228	0.680228	0.680228	0.843019	0.867999	0.867999	0.895238	1	1	1	1	0.973044	0.973044	0.973044
126	0.703678	0.71724	0.68589	0.68589	0.68589	0.68589	0.856365	0.859332	0.859332	0.88427	0.973044	0.973044	0.973044	0.973044	1	1	1
127	0.703678	0.71724	0.68589	0.68589	0.68589	0.68589	0.856365	0.859332	0.859332	0.88427	0.973044	0.973044	0.973044	0.973044	1	1	1
128	0.703678	0.71724	0.68589	0.68589	0.68589	0.68589	0.856365	0.859332	0.859332	0.88427	0.973044	0.973044	0.973044	0.973044	1	1	1
130	0.703678	0.71724	0.68589	0.68589	0.68589	0.68589	0.856365	0.859332	0.859332	0.88427	0.973044	0.973044	0.973044	0.973044	1	1	1
129	0.389805	0.396001	0.376068	0.376068	0.376068	0.376068	0.534595	0.488769	0.488769	0.505716	0.577323	0.577323	0.577323	0.577323	0.728487	0.728487	0.728487
175	0.060374	0.060432	0.053737	0.053737	0.053737	0.053737	0.15109	0.094517	0.094517	0.101269	0.141348	0.141348	0.141348	0.141348	0.310792	0.310792	0.310792
177	0.060374	0.060432	0.053737	0.053737	0.053737	0.053737	0.15109	0.094517	0.094517	0.101269	0.141348	0.141348	0.141348	0.141348	0.310792	0.310792	0.310792
178	0.060374	0.060432	0.053737	0.053737	0.053737	0.053737	0.15109	0.094517	0.094517	0.101269	0.141348	0.141348	0.141348	0.141348	0.310792	0.310792	0.310792
134	0.023689	0.024744	0.02053	0.02053	0.02053	0.02053	0.067189	0.046738	0.046738	0.050695	0.074314	0.074314	0.074314	0.074314	0.144864	0.144864	0.144864

Fig. 3 Example of similarity matrix X (term–term) derived form SM

0	1	2	87	88	24	40	39	23	25	26	21	56	44	55	43	33	34	22
1	1	0.998966	0.968897	0.964342	0.768245	0.742865	0.712995	0.585182	0.54082	0.486412	0.107322	-0.03035	-0.00227	-0.00227	0.011617	-0.06646	-0.05077	0.091014
2	0.998966	1	0.956951	0.951608	0.739047	0.71224	0.681607	0.548964	0.539291	0.483195	0.108134	-0.02876	-0.00424	-0.00424	0.009324	-0.05592	-0.04197	0.092118
87	0.968897	0.956951	1	0.999826	0.897733	0.871222	0.842549	0.757215	0.529854	0.479691	0.100505	-0.03646	0.003356	0.003356	0.015694	-0.11517	-0.08896	0.074483
88	0.964342	0.951608	0.999826	1	0.905774	0.879483	0.851009	0.769202	0.529268	0.479638	0.101348	-0.03601	0.004649	0.004649	0.017037	-0.11742	-0.09112	0.075973
24	0.768245	0.739047	0.897733	0.905774	1	0.982651	0.961559	0.967338	0.469457	0.437441	0.114262	-0.02072	0.036165	0.036165	0.049235	-0.15432	-0.13097	0.111012
40	0.742865	0.71224	0.871222	0.879483	0.982651	1	0.994921	0.958004	0.433807	0.428285	0.085609	-0.03566	0.044426	0.044426	0.065164	-0.19124	-0.16545	0.0974
39	0.712995	0.681607	0.842549	0.851009	0.961559	0.994921	1	0.943225	0.41259	0.4213	0.074111	-0.04128	0.048558	0.048558	0.0728	-0.20612	-0.18009	0.089298
23	0.585182	0.548964	0.757215	0.769202	0.967338	0.958004	0.943225	1	0.396834	0.37866	0.114943	-0.00857	0.053875	0.053875	0.066838	-0.1604	-0.14265	0.129303
25	0.54082	0.539291	0.529854	0.529268	0.469457	0.433807	0.41259	0.396834	1	0.915996	0.412867	0.074459	0.137464	0.137464	0.186314	-0.00818	-0.0648	0.373823
26	0.486412	0.483195	0.479691	0.479638	0.437441	0.428285	0.4213	0.37866	0.915996	1	0.537579	0.230861	0.346072	0.346072	0.405328	0.095088	0.044994	0.460574
21	0.107322	0.108134	0.100505	0.101348	0.114262	0.085609	0.074111	0.114943	0.412867	0.537579	1	0.911342	0.906228	0.906228	0.906409	0.86308	0.821456	0.886772
56	-0.03035	-0.02876	-0.03646	-0.03601	-0.02072	-0.03566	-0.04128	-0.00857	0.074459	0.230861	0.911342	1	0.941683	0.941683	0.920822	0.91505	0.91127	0.842189
44	-0.00227	-0.00424	0.003356	0.004649	0.036165	0.044426	0.048558	0.053875	0.137464	0.346072	0.906228	0.941683	1	1	0.995161	0.838279	0.836712	0.737778
55	-0.00227	-0.00424	0.003356	0.004649	0.036165	0.044426	0.048558	0.053875	0.137464	0.346072	0.906228	0.941683	1	1	0.995161	0.838279	0.836712	0.737778
43	0.011617	0.009324	0.015694	0.017037	0.049235	0.065164	0.0728	0.066838	0.186314	0.405328	0.906409	0.920822	0.995161	0.995161	1	0.803973	0.79492	0.747138
33	-0.06646	-0.05592	-0.11517	-0.11742	-0.15432	-0.19124	-0.20612	-0.1604	-0.00818	0.095088	0.86308	0.91505	0.838279	0.838279	0.803973	1	0.98277	0.763773
34	-0.05077	-0.04197	-0.08896	-0.09112	-0.13097	-0.16545	-0.18009	-0.14265	-0.0648	0.044994	0.821456	0.91127	0.836712	0.836712	0.79492	0.98277	1	0.69349
22	0.091014	0.092118	0.074483	0.075973	0.111012	0.0974	0.089298	0.129303	0.373823	0.460574	0.886772	0.842189	0.737778	0.737778	0.747138	0.763773	0.69349	1
82	0.085868	0.10099	-0.02175	-0.02478	-0.07171	-0.06127	-0.06467	-0.07671	0.264237	0.234786	0.326706	0.165042	0.106609	0.106609	0.137045	0.328113	0.185385	0.549643
81	0.083097	0.099492	-0.03254	-0.03618	-0.09584	-0.08051	-0.07914	-0.10609	0.272612	0.216511	0.173561	0.006217	-0.04085	-0.04085	-0.00462	0.170403	0.029775	0.414199
6	-0.01192	-0.01544	-0.02615	-0.02238	0.086873	0.143639	0.151397	0.162384	-0.11559	-0.11997	-0.04005	0.008807	-0.0331	-0.0331	-0.02454	0.018161	-0.03584	0.26319
5	-0.04425	-0.04678	-0.05688	-0.05358	0.042971	0.091483	0.102196	0.110369	-0.09867	-0.11607	-0.027	0.026876	-0.07783	-0.07783	-0.06391	0.017618	-0.0461	0.332957
27	0.019315	0.005463	0.062904	0.069545	0.231625	0.293704	0.301232	0.322638	-0.23152	-0.20008	-0.10545	0.005362	0.038079	0.038079	0.030313	-0.0443	-0.03673	0.067509
28	0.021461	0.006959	0.068758	0.075623	0.241944	0.303706	0.311126	0.334349	-0.23035	-0.19637	-0.09295	0.020668	0.052676	0.052676	0.044302	-0.03475	-0.02543	0.077731

Fig. 4 Example of extracting block of sub-matrix from the matrix Y (sentence–sentence) derived from combined matrixes

annotated by humans' annotators. Finally, Table 2 compares our system with other well-known systems on the ACE RDC 2003 corpora.

6.2.1 Evaluation metrics

We use the standard evaluation for measuring precision, recall and f-measure, defined as follows:

$$\mathrm{Prec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

$$\mathrm{Rec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$F = \frac{2 \cdot \mathrm{Prec} \cdot \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}} \tag{12}$$

where TP is the number of true positive, FP is the number of false positive, and FN is the number of false negative.

6.2.2 Results

Table 2 shows that our model greatly outperforms previous systems, and the results are comparable with those of Zhou et al. [37] composite kernel methods. The SREC outperforms the feature vector-based kernel by 5.9 % precision and 8.8 % F-measure. The SREC also beats the dependency kernel by 16 % precision and 31 % F-measure.

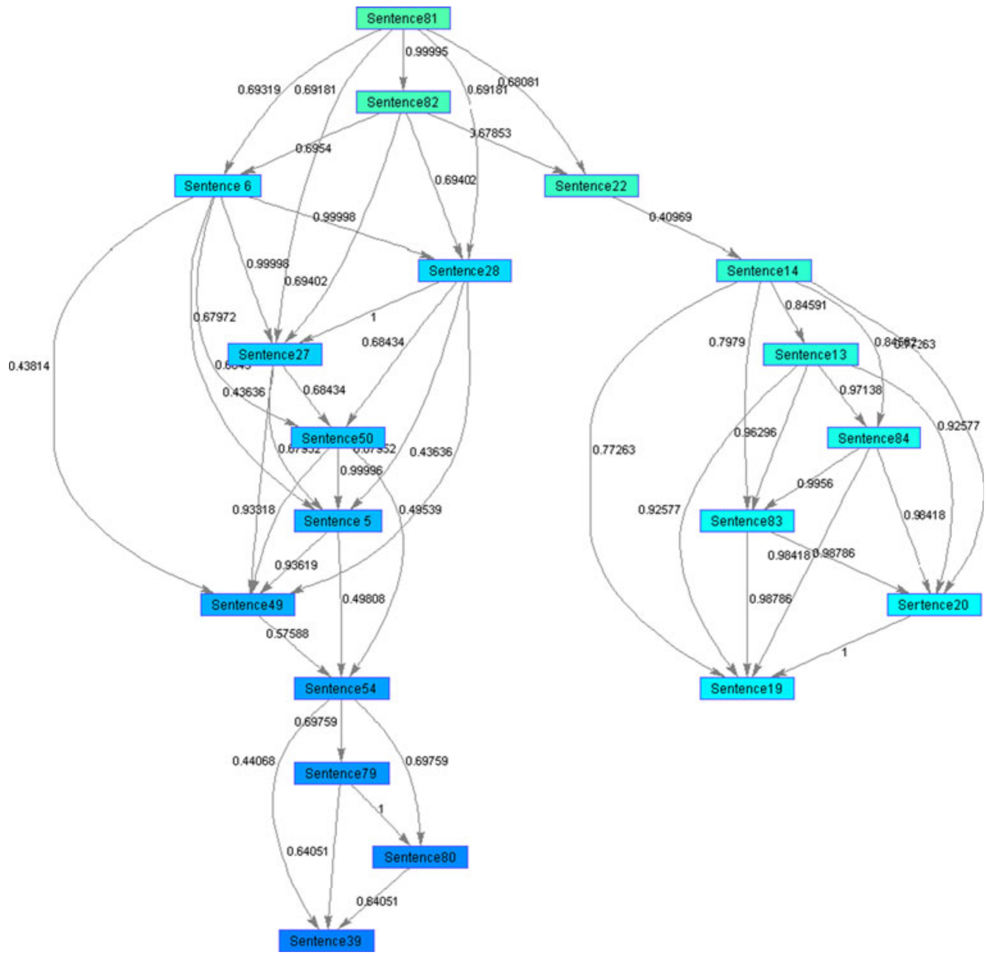

Fig. 5 Example of displaying strong connected sentence

Table 1 A list of relation types and the respective frequencies in the ACE-2003 corpus

Relations	At	Near	Part	Role	Social
No. of training	1,602	220	749	2,927	611
No. of test	389	70	163	712	112

Results indicate that SREC has significance different in terms of precision and *F*-measure in comparison with shortest-path kernel.

7 Conclusion and future work

In this paper, we have presented an approach for extracting semantic relation from corpus where syntactic and

Table 2 Comparison of the different systems on the ACE RDC 2003 corpus over the five relation types

ACE RDC 2003	Method	Precision	Recall	*F*-measure
Zhang et al. [21, 22]	Composite kernel	77.3	65.6	70.9
Zhang et al. [21, 22]	Standard CTK	76.1	62.6	68.7
Bunescu and Mooney [38]	Shortest path kernel	65.5	43.8	52.5
Culotta and Sorensen [39]	Dependency kernel	67.1	35.0	45.8
Zhou et al. [23, 40]	Feature vector-based	77.2	60.7	68.0
Zhou et al. [37]	Composite kernel	**82.3**	70.1	**75.7**
Zhou et al. [37]	Context-sensitive	82.1	67.2	73.9
SREC	Corpus-based and SVD	**83.1**	71.4	**76.8**

The bold represent that our method results clearly outperforms best results until now semantic knowledge along linear algebra method of SVD is employed, instead of exploring the full parse tree information directly and using dependency parse trees or applying WordNet with feature-based method. Although tree kernel-based approaches facilitate the exploration of the explicit relation, yet the current technologies are expected to be further advanced to effectively extract implicit relation. Evaluation of the ACE RDC corpora shows that our approach of combining statistic and semantic information can achieve better results. The experimental result also shows that our approach significantly outperforms the tree kernel-based and feature-based approaches. However, we think that adopting some techniques such as co-reference resolution (occurs when multiple expressions in a sentence or document refer to the same thing) and word sense disambiguation approaches may play a crucial role in detecting all implicit and explicit relations in accurate order.

Acknowledgments Authors thank referees for their careful reading of the manuscript and suggestions.

References

1. Beisswanger E (2010) Exploiting relation extraction for ontology alignment. In: Presented at the 9th international semantic web conference (ISWC2010), Shanghai
2. Iria J, Ciravegna F (2005) Relation extraction for mining the semantic web. In: Presented at the machine learning for the semantic web, Dagstuhl
3. Karoui L et al (2007) Analyses and fundamental ideas for a relation extraction approach. In: IEEE 23rd international conference on data engineering (ICDEW '07)
4. Imsombut A (2009) A statistical approach for semantic relation extraction. In: Eighth international symposium on natural language processing, Bangkok
5. Khoo CSG et al (2001) Using cause–effect relations in text to improve information retrieval precision. Inf Process Manag Int J arch 37(1):119–145
6. Had M et al (2009) Relation extraction for monitoring economic networks. In: 14th international conference on applications of natural language to information systems (NLDB 2009), Saarbrücken
7. Manine A-P et al (2009) Learning ontological rules to extract multiple relations of genic interactions from text. Int J Med Inf 78:e31–e38
8. Khoo CSG, Na J-C (2006) Annual review of information science and technology, vol 40. Wiley, New York, pp 157–228
9. Wittgenstein L (1953) Philosophical investigations. MacMillan, New York
10. Reynaud C, Safar B (2007) Exploiting wordnet as background knowledge. In: The 6th international semantic web conference and the 2nd Asian semantic web conference, Busan
11. Halliday MAK, Hasan R (1976) Cohesion in English. Longman, London
12. Yangarber R (2003) Counter-training in the discovery of semantic patterns. In: 41st annual meeting of the association for computational linguistics (ACL-03), Sapporo, pp 343–350
13. Stevenson M, Greenwood MA (2005) Asemantic approach to IE pattern induction. In: 43rd annual meeting of the association for computational linguistics, Ann Arbor, pp 379–386
14. Sudo K et al (2003) An improved extraction pattern representation model for automatic IE pattern acquisition. In: 41st annual meeting of the association for computational linguistics (ACL-03), Sapporo, pp 224–231
15. Miller S et al (2000) A novel use of statistical parsing to extract information from text. In: Sixth applied natural language processing conference, pp 226–233
16. Gamallo P et al (2002) Mapping syntactic dependencies onto semantic relations. In: Presented at the ECAI workshop on machine learning and natural language processing for ontology engineering, Lyon
17. Nedellec C (2000) Corpus-based learning of semantic relations by the ILP system, ASIUM. Learn Lang Log Lect Notes Comput Sci 1925:259–278
18. Zelenko D et al (2003) Kernel methods for relation extraction. Mach Learn Res 3:1083–1106
19. Zhao S, Grishman R (2005) Extracting relations with integrated information using kernel methods. In: 43rd annual meeting on association for computational linguistics. Ann Arbor
20. Reichartz F et al (2009) Dependency tree kernels for relation extraction from natural language text. In: Presented at the European conference on machine learning and knowledge discovery in databases(ECML PKDD '09), Bled
21. Zhang M et al (2006) A composite kernel to extract relations between entities with both flat and structured features. In: International conference on computational linguistics and the annual meeting of the association for computational linguistics (COLING-ACL'2006), Sydney, pp 825–832
22. Zhang M et al (2008) Exploring syntactic structured features over parse trees for relation extraction using kernel methods. Inf Process Manag 44:687–701
23. Zhou GD, Zhang M (2007) Extracting relation information from text documents by exploring various types of knowledge. Inf Process Manag 43:969–982
24. Maedche A, Staab S (2001) Ontology learning for the semantic web. IEEE Intell Syst 16(2):72–79
25. Reinberger ML et al (2004) Automatic initiation of an ontology. In: Move to meaningful internet systems, Napa, pp 600–617
26. Yamaguchi T (2001) Acquiring conceptual relations from domain-specific texts. In: IJCAI 2001 workshop on ontology learning (OL'2001), Seattle
27. Kavalec M, Svatek V (2005) A study on automated relation labelling in ontology learning. In: Buitelaar P, Cimiano P, Magnini B (eds) A study of automated relation labelling in ontology learning, ontology learning from text: methods, evaluation and applications. IOS Press, Amsterdam, pp 44–58
28. Schutz A, Buitelaar P (2005) RelExt: a tool for relation extraction from text in ontology extension. In: 4th international semantic web conference, Galway
29. Yan Y et al (2009) An integrated approach for relation extraction from Wikipedia texts. In: Presented at the content analysis in web 2.0(CAW2), Madrid
30. Landauer TK, Dumais ST (1997) A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychol Rev 104(2):211–240
31. Kontostathis A et al (2005) Identification of critical values in latent semantic indexing. Found Data Min Knowl Discov 6:333–346
32. Marquez L et al (2008) Semantic role labeling: an introduction to the special issue. Comput Linguist 34:1–16
33. Hofstadter DR (1999) Gödel, escher, bach: an eternal golden braid. Art of Mentoring Series, Basic Books, New York

34. Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137
35. Greengrass E (2000) Information retrieval: a survey. Technical report, Department of defence, Washington
36. Quirk C et al (2004) Monolingual machine translation for paraphrase generation. In: Conference on empirical methods in natural language processing, Barcelona
37. Zhou GD et al (2010) Tree kernel-based semantic relation extraction with rich syntactic and semantic information. Inf Sci 180:1313–1325
38. Bunescu R, Mooney RJ (2005) A shortest path dependency kernel for relation extraction. In: Human language technology conference and conference on empirical methods in natural language (HLT/EMNLP'2005), Vancouver, pp 724–731
39. Culotta A, Sorensen J (2004) Dependency tree kernels for relation extraction. Annual meeting of the association for computational linguistics (ACL'04), Barcelona, pp 423–429
40. Zhou GD et al (2005) Exploring various knowledge in relation extraction. Annual meeting of the association for computational linguistics (ACL'2005), Michigan, pp 427–434