

An Alignment-Based Approach to Semi-supervised Relation Extraction Including Multiple Arguments

Seokhwan Kim¹, Minwoo Jeong¹, Gary Geunbae Lee¹,
Kwangil Ko², and Zino Lee²

¹ Department of Computer Science and Engineering,
Pohang University of Science and Technology,
San 31, Hyoja-dong, Nam-gu, Pohang, 790-784, Korea
{megaup, stardust, gblee}@postech.ac.kr

² Alticast Corp., 15th Floor, Nara Investment Banking Corp. Bldg. 1328-3,
Seocho-Dong, Seocho-Gu, Seoul, 137-858, Korea
{kik, zino}@alticast.com

Abstract. We present an alignment-based approach to semi-supervised relation extraction task including more than two arguments. We concentrate on improving not only the precision of the extracted result, but also on the coverage of the method. Our relation extraction method is based on an alignment-based pattern matching approach which provides more flexibility of the method. In addition, we extract all relationships including two or more arguments at once in order to obtain the integrated result with high quality. We present experimental results which indicate the effectiveness of our method.

1 Introduction

During the past few years, we have been able to obtain a large amount of information about various topics through the Internet. However, the high accessibility of the Internet has caused the trend of information overflow which makes it difficult to obtain valuable information due to excessive amount of information rather than lack of it. In order to improve the efficiency of gathering valuable information, the information extraction task has been actively researched by many researchers, and it has grown into one of the most important topics of natural language processing field.

The area of information extraction is divided into several subtasks by the characteristic and range of target information, and most of them can be generalized by extracting the defined number of relevant arguments from natural language documents. Named entity recognition and binary relation extraction tasks can be considered as special cases of the above-mentioned generalized concept of information extraction, which define the number of extracted arguments as 1 and 2, respectively. Both subtasks are the most widely researched topics in

the information extraction tasks, and several researchers have shown that supervised machine learning based approaches are significantly effective ways to solve these problems. [1] [2]

However, supervised machine learning methods have a cost problem by requiring a considerable amount of training data for achieving good performance. In order to reduce the cost of building required resources with minimal performance loss, recently, semi-supervised machine learning methods have been attempted to solve the problem. Most of existing works for semi-supervised information extraction commonly concentrate on automatically creating context patterns guaranteeing high-precision by integrating statistical characteristics of target documents with grammatical induction methodologies. [3] [4] [5] [6] [7] [8]

In this paper, we will describe our semi-supervised information extraction approach with following two points of views which are little different from other existing works. The first issue is about the coverage of each context pattern. Our approach is based on the bootstrapping methods. In the case of bootstrapping, the high-precision is an absolutely important goal of the method, because even very small number of errors generated in earlier iteration can be enormously harmful to the overall performance due to error accumulation by iterating. Nevertheless, high-precision is not the only prerequisite for achieving high performance of the method. If it is guaranteed that the set of context patterns accumulated by iterating more than considerable times will have the sufficient coverage which is needed to extract all existing information, then reducing errors by improving the precision of context pattern induction is the best way to improve the overall performance. However, this assumption is far from realistic, because expressions indicating even the same information can be entirely different each other and each expression also can be derived into the totally new expressions as time goes by. It is difficult to keep up with the variety of the expressions only depending on the set of precise context patterns, and even if it is possible, it might require huge number of iteration which is limited by current computing power. In actuality, we should consider not only high-precision, but also the way of enhancing coverage of each context pattern for improving the overall performance. In order to encourage coverage of the method, we focused on the task of context pattern matching rather than context pattern induction, and we will present an alignment-based information extraction method as a pattern matching approach in our method.

The other issue is about the number of extracted arguments of the task. Although most of existing works have concentrated on the task of extracting individual named-entities or relationships between just two named-entities, in many cases, we should extract the relationship including more than two arguments. For extracting the n-ary relationships, we applied our alignment-based information extraction method to the task of extracting relationships including not only just two arguments, but also more than two arguments. Moreover, we will present a reinforcement scheme based on the result of bottom-up integration, starting with the result of binary relation extraction.

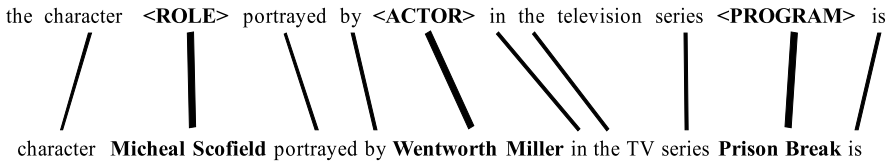


Fig. 1. Sentence alignment for extracting multiple relevant arguments

The remainder of this paper is organized as follows: In the next section, we present a detailed description of the alignment-based information extraction method. In section 3, the overall architecture of our method and detailed descriptions of each subtask are presented. We present the experimental results in section 4, and our conclusions are provided in section 5.

2 Alignment-Based Information Extraction

Kim *et al.* [9] presented an alignment-based named entity recognition method to solve the spoken language understanding problem. We modified the method to extract not individual named entities, but tuples including two or more relevant arguments, and applied this modified method to the task of n-ary relation extraction. As shown in Fig. 1, we align a raw sentence with a context pattern which is a part of sentence containing labels of target arguments. Then, from the result of the best alignment between them, we extract the parts of the raw sentence which are aligned to the argument labels in the context pattern, and incorporate the extracted arguments into a tuple which is a candidate of n-ary relation. In Fig. 1, a tuple (Prison Break, Michael Scofield, Wentworth Miller) is extracted as a candidate of ternary relationship, (PROGRAM, ACTOR, ROLE), which means that an ACTOR acts a role of ROLE on a PROGRAM.

In order to enhance the coverage of each context pattern, we should consider the flexibility of the alignment task. Accordingly, we adapted an alignment scheme based on the Smith-Waterman algorithm [10], which is a widely used biological sequence alignment algorithm providing a systematic way of controlling the flexibility of the task. We utilize this algorithm into the sentence alignment task by considering a word or a morpheme as a unit of alignment instead of biological residues.

The alignment algorithm is performed by computing the score for each word pair in the alignment matrix M . Each row in the matrix corresponds to a word in the context pattern, while each column in the matrix corresponds to a word in the raw sentence. Moreover, a point of crossing between a row and a column has the score of aligning the word in the raw sentence with the word in the context pattern.

The first step in the alignment method is to assign the initial value of each position in the matrix M with 0. And then, we find the maximum alignment score by starting on the upper left hand corner in the matrix M and continuing

| | | | | | | | | | | | | | | | |
|------------|-----|-----------|---------|----------|-----------|----|-----------|--------|----|-----|----|--------|--------|-------|----|
| | the | character | Michael | Scofield | portrayed | by | Wentworth | Miller | in | the | TV | series | Prison | Break | is |
| character | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| <ROLE> | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| portrayed | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| by | 1 | 1 | 2 | 2 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| <ACTOR> | 1 | 2 | 2 | 3 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| in | 1 | 2 | 2 | 3 | 3 | 4 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| the | 1 | 2 | 2 | 3 | 3 | 4 | 5 | 5 | 6 | 7 | 7 | 7 | 7 | 7 | 7 |
| television | 1 | 2 | 2 | 3 | 3 | 4 | 5 | 5 | 6 | 7 | 7 | 7 | 7 | 7 | 7 |
| series | 1 | 2 | 2 | 3 | 3 | 4 | 5 | 5 | 6 | 7 | 7 | 8 | 8 | 8 | 8 |
| <PROGRAM> | 1 | 2 | 3 | 3 | 4 | 4 | 5 | 6 | 6 | 7 | 8 | 8 | 9 | 9 | 9 |
| is | 1 | 2 | 3 | 3 | 4 | 4 | 5 | 6 | 6 | 7 | 8 | 8 | 9 | 9 | 10 |

Fig. 2. An example of computed alignment matrix

to find the maximum score $M_{i,j}$ for each position in the matrix according to the lower right direction. The maximum score $M_{i,j}$ is defined as

$$M_{i,j} = \max \begin{pmatrix} M_{i-1,j-1} + \text{sim}_{i-1,j-1} \\ M_{i-1,j} + gp \\ M_{i,j-1} + gp \\ 0 \end{pmatrix}, \quad (1)$$

where $\text{sim}_{i,j}$ is the value of similarity between the i -th word in the context pattern and j -th word in the raw sentence, and gp is the pre-defined penalty for a gap. Fig. 2 shows an example of computed matrix for the alignment which is shown in Fig. 1. In this example, we defined the similarity function as

$$\text{sim}_{i,j} = \begin{cases} 1, & \text{if PTN}_i \text{ and RAW}_j \text{ are identical} \\ & \text{or PTN}_i \text{ is an argument label} \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where PTN_i is the i -th word in the context pattern, while RAW_j is the j -th word in the raw sentence. And the value of gap penalty, gp , is ignored in this example.

After matrix computation, we trace back the matrix to find the best alignment with the maximum score and extract the relevant arguments from the result of alignment. The traceback task is started at the position with maximum score on the alignment matrix. For each current position $[i, j]$, the next position is determined by the following policies in order:

1. if $M_{i,j} = M_{i,j-1} + gp$, then the next position is $[i, j - 1]$.
2. if $M_{i,j} = M_{i-1,j-1} + \text{sim}_{i,j}$, then the next position is $[i - 1, j - 1]$.
3. if $M_{i,j} = M_{i-1,j} + gp$, then the next position is $[i - 1, j]$.

The order of applying the policies should be preserved. Although most of sequence alignment methods based on the Smith-Waterman algorithm consider the diagonal advancement corresponding to our second policy as a prior direction, we give a preference to the left direction by applying the policy of the left position first, because we should make it possible to align each argument label

in the context pattern with two or more words in the raw sentence in order to extract the arguments which consist of multiple words.

In Fig. 2, the sequence of positions with gray color indicates the best alignment with maximum alignment score. From the result of the alignment, we can extract the words “Michael Scofield”, “Wentworth Miller”, and “Prison Break” as relevant arguments which have types of ROLE, ACTOR, and PROGRAM respectively.

3 Semi-supervised Relation Extraction Including Multiple Arguments

Most of existing works about relation extraction have concentrated on the task of extracting relationships including just two arguments, regardless of supervised or semi-supervised approach. However, the binary relation extraction might not be sufficient in some circumstances.

Firstly, we consider the case that we should extract a relationship which includes more than two arguments. In the example which is mentioned in the previous section, the binary relationship between ACTOR and ROLE arguments has to be specified by another argument about the corresponding PROGRAM, because an actor can be related to various roles according to the performed programs. In order to extract relationships with multiple arguments, we can consider an approach of integrating several binary relationships into an n-ary relationship. For example, for obtaining relationships including three arguments which are PROGRAM, ACTOR, and ROLE, we should extract the following three binary relationships, (ACTOR,ROLE), (PROGRAM,ACTOR), and (PROGRAM,ROLE) and integrate them together. However, we have a problem that the errors originated from each task of binary relation extraction are accumulated into the integrated result. Added to that, this problem is getting worse as the number of arguments of the target relationship is increased.

The second problem is caused by the tendency of relevant arguments to be contiguously located each other. Although we want to deal with a complete binary relationship which has a specified meaning with just two arguments, other relevant arguments located closely to the target arguments might interfere the task of binary relation extraction. In the case of the (PROGRAM,ROLE) relationship in the previous example, another argument, ACTOR, tends to be closely located to both PROGRAM and ROLE. For example, we consider a context pattern of the (PROGRAM,ROLE) relationship extracted from Fig. 1, ‘character ROLE portrayed by *Wentworth Miller* in the TV series PROGRAM is’. In this context pattern, ‘Wentworth Miller’ is the part of an ACTOR argument, and the coverage of the context pattern is dramatically weakened by this interposed argument.

In order to solve these problems which are caused by depending only on the binary relation extraction, we propose a new semi-supervised relation extraction method including multiple arguments. An overview of our method is shown in Fig 3.

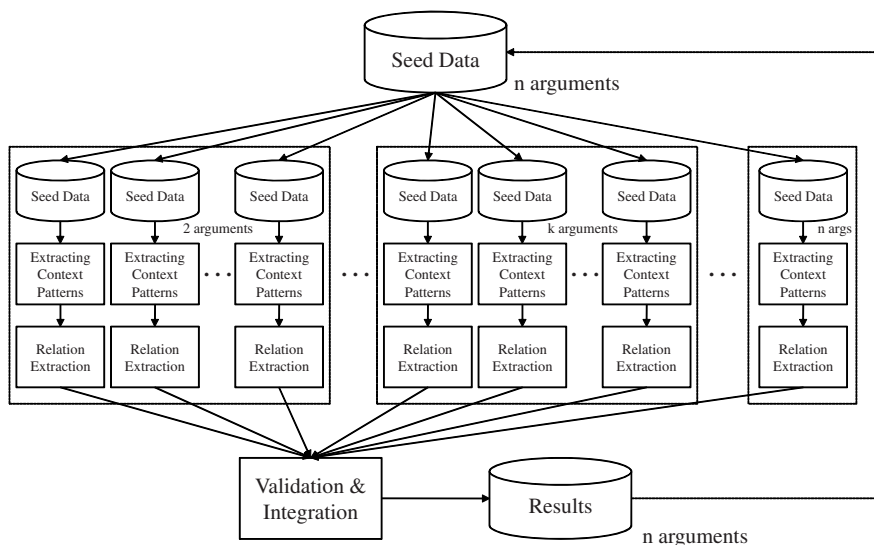


Fig. 3. Overall Architecture of the Method

If the number of arguments in the target relationship is n , we define the variable k which has all values between two and n . For each value of k , we consider all subsets which are organized by combinations of any k relevant arguments among overall n arguments in seed data. For example, if the target relationship is organized by four arguments, CHANNEL, PROGRAM, ACTOR, and ROLE, the value of k can be two, three, or four. The subsets which include two arguments are (CHANNEL,PROGRAM), (PROGRAM,ACTOR), and (ACTOR,ROLE), and the subsets are (CHANNEL,PROGRAM,ACTOR), (CHANNEL,PROGRAM,ROLE), and (PROGRAM,ACTOR,ROLE) for the k value of three. For the k value of four, we use seed data themselves, because k is equivalent to the value of n . For each individual subset, we execute the following series of tasks. Firstly, we separate the sub-seed data including only corresponding arguments from the original seed data. By using these separated data as seeds, we extract context patterns from the source documents. And then, based on the context patterns, we extract new tuples including k relevant arguments from source documents. These series of tasks are executed parallel for all derived subsets. The set of results in each parallel execution is verified and integrated by post-processing methods, and we can obtain the integrated result with n number of arguments which is equal to the number of the arguments in the original seed data. The above-mentioned tasks are performed in iterations of bootstrapping, and the results from the iteration are added to the seed data and affect to the next iteration.

Because our method extract not only binary relationships, but also all intermediate k -ary relationships up to the ultimate n -ary ones, we can reduce the accumulated errors in the integration of extracted tuples with relatively less arguments

through the cross-validation process between intermediate k -ary relationships. Also, the problem of interfering by other closely located relevant arguments can be solved by extracting more than two relevant arguments at once.

We present the detailed descriptions about subtasks in our method on the following subsections.

3.1 Context Patterns Extraction

Since, we use an alignment-based approach between raw sentences and context patterns for extracting relevant arguments as stated in section 2, each context pattern should take the form that can be aligned to raw sentences, hence we incorporate context patterns starting from the sentences in source documents.

For each tuple in the seed data, we search for the sentences containing all arguments of the tuple in source documents. Although we can directly utilize the full sentence as a context pattern, we segment out subpart of the sentence which densely contains the arguments for enhancing coverage of the context patterns. The range of subpart is determined by locations of arguments and the value of margin size m . We extract the subpart from the m -th word on the left hand of the leftmost argument to the m -th word on the right hand of the rightmost argument in the sentence. And then, we make a context pattern by replacing the parts of arguments in the sub-sentence with corresponding argument labels.

For example, for a seed tuple (Prison Break, Michael Scofield, Wentworth Miller) of the ternary relationship (PROGRAM, ACTOR, ROLE), we can extract a context pattern, ‘character (ROLE) portrayed by (ACTOR) in the TV series (PROGRAM) is’ from the raw sentence in Fig. 1 and the margin size m of one.

3.2 Relation Extraction Based on Pairwise Alignment

Each extracted context pattern is aligned pairwise with the sentences in the source documents for extracting candidate tuples containing relevant arguments. We can compute the alignment score for each alignment based on the alignment matrix M which is introduced in section 2. The alignment score is based on the maximum value in the alignment matrix M , and the position which has the maximum value is the start position of the trace-back task. Since we set the matching reward to 1 and both mismatching and gap penalty to 0, the maximum value on the matrix M means the number of equally aligned words in the best alignment. This value can be normalized by the length of the context pattern, and we define the alignment score as

$$score(PTN, RAW) = \frac{\max(M(PTN, RAW))}{\text{length}(PTN)}, \quad (2)$$

where PTN is a context pattern, RAW is a raw sentence, and $M(PTN, RAW)$ is the matrix which is computed by the task of alignment between them. We regard this score as a measure of reliability of extracted candidates, and we select only candidates with higher score than a threshold value as a result of extraction.

3.3 Alignment-Based Verification

Most of candidate tuples which are extracted parallel for each subset of seed data are still erroneous. The primary factor of the errors is the redundant attachment problem. This problem is caused by aligning not only argument itself but also contiguous words to the argument with an argument label in the context pattern. For example, if a context pattern, ‘character ⟨ROLE⟩ portrayed by ⟨ACTOR⟩ in ⟨PROGRAM⟩ is’ is aligned with the raw sentence in Fig. 1, we will obtain not ‘Prison Break’, but ‘the TV series Prison Break’ as the argument of PROGRAM. The redundant attachment problem is more serious in case that propositional words and particles are frequently omitted, or morpheme-based processing is required, such as in Korean language.

In order to solve the redundant attachment problem, we propose a verification approach which is also based on the alignment method. We consider the alignment score as a measure of similarity between two candidate arguments, and the similarity is defined as

$$similarity(A, B) = \frac{\max(M(A, B)) \times 2}{\text{length}(A) + \text{length}(B)} \quad (3)$$

where both A and B are candidate arguments.

For verifying the candidate tuples, firstly, we organize the clusters of similar tuples based on the tuple similarity measure defined as

$$sim(tuple1, tuple2) = \frac{\sum_{i=1}^{\#args} similarity(tuple1_i, tuple2_i)}{\# \text{ of arguments}} \quad (4)$$

where *tuple1* and *tuple2* are candidate tuples being compared each other, and *tuple_i* is the *i*-th argument in the tuple. We consider tuples which pairwise have higher similarity than a threshold value as a cluster.

And then, we perform the task of pairwise alignment of each argument in a cluster of tuples. For each argument, we replace it with the argument which has the maximum summation of similarities. For example, if there are considerable number of (Prison Break, Michael Scofield, Wentworth Miller) and a few (the TV series Prison Break, Michael Scofield, Wentworth Miller) in a cluster of the tuples for the relationship (PROGRAM, ACTOR, ROLE), ‘the TV series Prison Break’ might be replaced by ‘Prison Break’ which has the maximum summation of similarities. By this alignment verification, the distribution of candidates is reflected to the final result, and it plays the important role of reflecting the statistical characteristics of data in the overall method.

3.4 Bottom-Up Integration

From the extracted and verified tuples including relatively small number of arguments, we can integrate new tuples with more arguments. For example, we can make a new tuple of (PROGRAM, ACTOR, ROLE) by integrating a tuple of (PROGRAM, ROLE) relationship and a tuple of (ACTOR, ROLE) relationship which have the common argument of ROLE. However, these new integrated

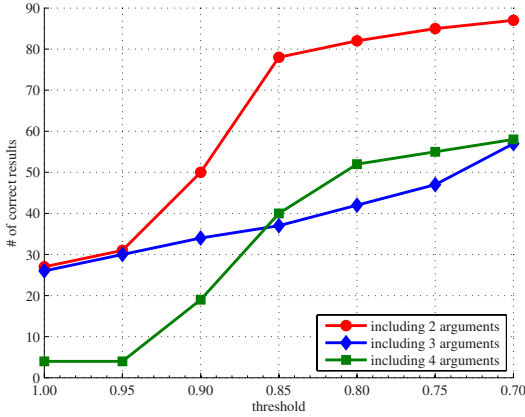


Fig. 4. Comparison of the coverage for various threshold values

tuples can come into collision with existing tuples which are extracted for the consisted arguments at once. In order to resolve the collision, we performed the previously mentioned verification task again to the set of the tuples which contains both existing and integrated tuples.

4 Experiments

We evaluated our method on 930 Korean news documents about TV series which consist of 13,175 sentences. Only a tuple with 4 arguments about the relationship (CHANNEL, PROGRAM, ACTOR, ROLE) is used as a seed information. Each result is collected after the first iteration and evaluated manually.

Firstly, we performed the method for various values of the threshold which affect to the acceptance rate of extracted candidates in the task of relation extraction. We selected all values of the threshold from 1.0 to 0.7 by downing the difference of 0.05. And the results were evaluated by manually counting of correctly extracted argument pairs after the verification task. The tuples with more than two arguments were separated into several pairs of arguments according to the corresponding binary relationships for evaluation. The experimental results are shown in Fig. 4. As the value of threshold decreases, the number of correctly extracted relevant arguments increases regardless of the number of target arguments. It means that the coverage of context patterns can be enhanced by the flexible setting of the threshold in the alignment-based approach.

And then, we evaluated the differences between before and after the verification task for a fixed threshold value, 0.85, which is determined empirically. The compared result shown in Table 1 indicates that the verification task contributes to dramatically improve the precision of the extracted results. From the analysis on the errors, we discovered that only 16.28% of the errors in the verified results are caused by the redundant attachment problem, while 82.0% of the errors occurred by the redundant attachment before verification. We can confirm that the

Table 1. Result of the verification

| types of relations | before verification | | after verification | |
|------------------------------|---------------------|-----------|--------------------|-----------|
| | # of tuples | precision | # of tuples | precision |
| (ACTOR,ROLE) | 249 | 36.55 | 79 | 73.42 |
| (PROGRAM,ROLE) | 19 | 52.63 | 17 | 58.82 |
| (PROGRAM,ACTOR) | 10 | 60 | 10 | 60 |
| (CHANNEL,PROGRAM) | 12 | 33.33 | 6 | 66.67 |
| (PROGRAM,ACTOR,ROLE) | 7 | 42.86 | 5 | 60 |
| (CHANNEL,PROGRAM,ROLE) | 18 | 55.56 | 16 | 81.25 |
| (CHANNEL,PROGRAM,ACTOR) | 8 | 62.5 | 8 | 75 |
| (CHANNEL,PROGRAM,ACTOR,ROLE) | 15 | 60 | 14 | 85.71 |

Table 2. Result of the integration

| types of relations | with only binary relations | | with all intermediates | |
|------------------------------|----------------------------|-----------|------------------------|-----------|
| | # of tuples | precision | # of tuples | precision |
| (PROGRAM,ACTOR,ROLE) | 9 | 77.78 | 9 | 88.89 |
| (CHANNEL,PROGRAM,ROLE) | 11 | 81.82 | 16 | 87.5 |
| (CHANNEL,PROGRAM,ACTOR) | 12 | 58.33 | 9 | 77.78 |
| (CHANNEL,PROGRAM,ACTOR,ROLE) | 8 | 87.5 | 16 | 87.5 |

alignment-based verification elevates the performance of the relation extraction by solving the redundant attachment problem.

As the last experiment, we compared the result of the bottom-up integration using all intermediate sub-tuples with the result of integration which depends on only binary relationships. As shown in Table 2, by using not only binary tuples, but also intermediate sub-tuples with more than two arguments, we can obtain more precise integrated results with wider coverage than the cases of depending on only binary relationships.

5 Conclusions

We have presented an alignment-based approach to semi-supervised relation extraction including multiple arguments. Using the alignment-based pattern matching approach, we improved the coverage of context patterns. And we solved the redundant attachment problem which causes the critical precision loss, by introducing the alignment-based verification method. In the integration phase, we considered not only binary relationships, but also all k -ary intermediate relationships, which produced more improved results than binary relationship-based integration.

On the other hand, there are still more rooms to be improved in our approach such as lack of statistical and linguistic features. Although, the statistical characteristic of the data is used for verification in the current method, it should also be utilized for extracting context patterns in order to obtain more reliable context patterns. In the case of linguistic information, we expect that it can be

reflected by defining more systematic policies of the alignment method. Refining the method and applying it to more sophisticated problems such as automatic ontology population are our future works.

Acknowledgements

This research was supported by the Invitation for Proposals on Internet Services Theme funded by Microsoft Research Asia.

References

1. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of The Seventh Conference on Natural Language Learning (CoNLL-2003) (2003)
2. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. *Journal of Machine Learning Research* 3, 1083–1106 (2003)
3. Riloff, E., Jones, R.: Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 474–479 (1999)
4. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web - an experimental study. *Artificial Intelligence Journal* 165, 91–134 (2005)
5. Lee, S., Lee, G.G.: Exploring phrasal context and error correction heuristics in bootstrapping for geographic named entity annotation. *Information Systems* 32, 575–592 (2007)
6. Agichtein, E., Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of the fifth ACM conference on Digital libraries, pp. 85–94 (2000)
7. Yangarber, R., Grishman, R., Tapanainen, P., Huttunen, S.: Automatic Acquisition of Domain Knowledge for Information Extraction. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), pp. 940–946 (2000)
8. Sudo, K., Sekine, S., Grishman, R.: An improved extraction pattern representation model for automatic IE pattern acquisition. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), pp. 224–231 (2003)
9. Kim, S., Jeong, M., Lee, G.G.: A semi-supervised method for efficient construction of statistical spoken language understanding resources. In: Proceedings of the Interspeech 2007-Eurospeech (2007)
10. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, 195–197 (1981)