Finding Relations from a Large Corpus using Generalized Patterns

Hyemin Kim[1], Heesoo Kim[1], Ikkyu Choi[1], and Minkoo Kim[2]

[1] Graduate School of Information and Communication, Ajou University,
Suwon, Kyonggido 443-749, Republic of Korea
[2] College of Information and Computer Engineering, Ajou University,
Suwon, Kyonggido 443-749, Republic of Korea

cho@ajou.ac.kr, minkoo@ajou.ac.kr

Abstract

We can find many valuable relations from a text document. Extracting these relations from a document plays an important role in Information Extraction. However, it is not easy to achieve relation extraction task from a large corpus. This is due to the many kinds of relation and it's appearing patterns. To solve these difficulties, we suggest a method which automatically finds the relations using generalized patterns. In this method, generalized patterns are created using the information at the initial time. The patterns are used to find new terms in the specific relation. The evaluation of our method has been done over a collection of more than 100,000 sentences. The result shows improved performance.
Keyword: Information extraction, Generalized pattern, Large corpus.

I. Introduction

The world consists of conceptual units and there exist many relations between them. These relations can be found in a text document with various forms. With rapid development of internet, amount of text document grows every day and it is not easy to find useful relations from them. Thus there is much interest in finding relation from a document automatically. The core of automatic relation acquisition system is a set of *patterns* which is used to extract relevant relation information from a document. We can say that *patterns* are realization of a relation in a real document. For example, in a sentence like "X is a Y", "is a" is a pattern for hyponymy relation. In a sentence like "X is a kind of Y", "is a kind of" is a pattern for hyponymy. Relation extraction system can find relation from a document using patterns. If the system use validate patterns, the result of the system will be correct. However, it is not easy to make validate patterns. One of the most obvious problems in making validate patterns is diversity of a relation. As we mentioned in the first sentence, there exist many relations between terms and patterns of each relation are different. To extract each relation, different patterns are needed. Patterns like "is a" or "kind of" is needed for hyponymy while "is located in" or "is in" is needed for organization-location relation. Making relevant patterns for each relation costs a lot of time and human labor. Other problem is realization forms of relation in a real document. Though semantics of sentence represent same relation, the way of representation is different in a

document. For example, the sentence which contains meaning of hyponymy of X and Y can be wrote like “X is a Y” or “X is a [adjective]Y” or “X is a kind of Y”. To solve this, we developed hybrid method which automatically makes patterns and finds relations with a minimal human labor.
In this paper, we present a method for the automatic acquisition of relations. We developed the system which generalize patterns and finds relation from a plain-text document. In our system, user provides small information about the relation what they want to find. With a small human labor-initial information- relevant patterns are generated according to the each relation. To recognize various forms of patterns, we simplified context of the sentence. Our system is built on the ideas of Snowball, which we describe next.

Snowball. Snowball[7] is a novel system which finds patterns between location and organization from a document collection. Snowball is initially given a handful of valid tuples of organization and location. For example, tuples like<Microsoft, Redmond> are given in initial time. In order to generate a patterns, Snowball group occurrences of the initial tuples in documents. Left, middle, right contexts associated tuples are expressed as a vector. And 5-tuple is generated. 5-tuple consists of left vector, tag1, middle vector, tag2, right vector. In this case, tag1 is organization and tag 2 is location. Then clusters these 5-tuple using a simple single-pass bucket clustering algorithm[10], using the Match function which calculate the similarity between the 5-tuples. The centroid of each cluster becomes patterns. Using these patterns, Snowball finds a sentence that includes an organization and location as determined by the named-entity tagger. For an occurrence of organization and location tuples, Snowball generates 5-tuple $t =< l_c, tag_1, m_c, tag_2, r_c >$ using left, middle, right context. A candidate tuple is generated if there is a pattern t_p such that $Match(t, t_p) \geq \tau_{sim}$ where τ_{sim} is the clustering similarity threshold. For each candidate tuple, snowball store the set of patterns that generated it, each with an associated degree of match. Snowball uses this information to select new tuples from candidate tuples.
The goal of Snowball is finding organization and location relation. They used named-entity tagger which identifies every organization and location in a text document. In our system, there is no bound of terms in relation. Our method can be applied in finding any relation. Due to the fact that Snowball uses contexts of a sentence as it is, Snowball cannot recognize various patterns of relation in a sentence. To recognize various forms of patterns in relation, we applied soft pattern matching method from SP+PRF system[8]. For the experiment, we focus on hyponymy relation which is the most basic relations of terms

SP+PRF.SP+PRF system[8] is a QA system that generate definition sentence from the web. They focus on identifying the definition sentences from relevant news articles for recent terms for which structured knowledge bases have no definition. They applied soft matching method to extract definition patterns. We applied simplified version of soft pattern method that we present in Section 2. We discuss our method of finding hyponymy by simplifying the context in the next section. Section 3 describes the experimental setting and result. In the last section, we conclude paper and describe future direction.

II. Simplifying Contexts

In this section we present our method, which finds terms in hyponymy from the document collection. Like other relations, hyponymy exist in various forms in sentences. To recognize various forms in hyponymy we augment relevant context and makes patterns from them. For simplified patterns, some unnecessary terms are removed and some of terms are generalized into the other terms. To find the entities that related to hyponymy we applied the method advocated by Agichtein

and Gravano[7]. We find patterns in hyponymy and using the patterns, find new hyponymy by simplifying the context near the hyponymy terms. Fig.1 describes our simplifying pattern system.

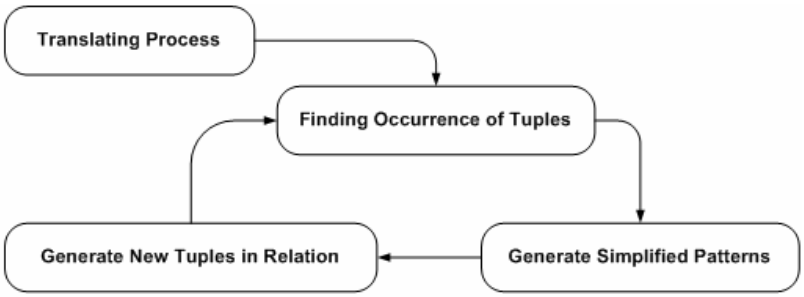

Fig. 1. This is the main process of simplifying patterns. Translating process is operated only once in the first time.

A. Translating Process

For simplified patterns, we only consider core of a sentence. Some of terms in a sentence are removed and some of them are translated into other words. Rules that we used are listed in Table 1.

Table 1. Categorization of Terms

Terms/POS tag	Category	Example
is, are, am, was, were	BE	Is ->BE
Noun, Noun phrases	Not translated	Kodak-> Kodak
Adjectival and Adverbal modifiers	To be deleted	
Determiner	DT	the ->DT

For POS tag, we used Brill Tagger[9]. The rule is a simple version of substitution heuristics from soft patterns method by Cui and Chua[8]. The main difference is we do not translate Noun or Noun phrases into NP because NP may be the target we want to find. Every Adjectival and Adverbal modifiers are deleted because these words do not play important role in hyponymy. After translating process is ended, sentences contain essence information. For example, a sentence like “Insignia is a privately-held firm” is translated into “Insignia BE DT firm”

B. Generalizing Simplified Patterns

To find patterns in hyponymy, valid seeds in hyponymy are provided as described in the first section. Initial seeds are listed in Table2.
The system finds occurrence of sentences which contain initial seeds. Then left context of tag_1 translated into left vector l_c, contexts between tag_1 and $tag2_2$ becomes vector m_c and right context of $tag2_2$ becomes vector r_c. A length of each vector is normalized as 1. The weight of a element in each vector is a function of the frequency of the term in the corresponding context. Then, 5-tuple $t =< l_c, tag_1, m_c, tag_2, r_c >$ is generated. These 5-tuple are clustered according to a simple single-pass clustering algorithm[10], using Match function[7].

Table 2. Initial Seed of Hyponymy

Hypernym	Hyponym
Protocols	Transmission Control
Program	Gauss
Telecommunication company	Bell
Firm	Insignia
Software	Network Computing System

The degree of match $Match(tp,ts)$ between two 5-tuples $t_p =< l_p, tag_1, m_p, tag_2, r_p >$ and $t_s =< l_s, tag_1, m_s, tag_2, r_s >$ is defined as:

$$Match(tp,ts) = \begin{cases} l_p \cdot l_s + m_p \cdot m_s + r_p \cdot r (\text{if the tags match}) \\ 0 \ (\text{otherwise}) \end{cases} \quad (1)$$

The pattern is the centroid of each cluster. The pattern is dropped if the number of 5-tuple which support the patterns is less than $\tau_{\sup}$.

The example of pattern from our system is described Table3. Patterns include concise information.

Table 3. Example of pattern from our system

Left vector	Middle vector	Right vector
<hardware, 0.354>	<DT,0.604>	<software, 0.354>
<and, 0.354>	<systems, 0.250>	<protocol, 0.354>
<support, 0.354>	<including, 0.604>	<tcp/ip, 0.354>
	<server,0.250>	

C. Generating New Tuples in Relation

Using the clusters from previous steps, the system finds new tuples in hyponymy. We used concept list which extracted from C-value/NC-value method[11] instead of named-entity tag for hyponymy. C-value/NC-value method generates the important word list-concept list from the document collection based on frequency of terms. The C-value, that aims to improve the extraction of nested multi-word terms. The NC-value, that incorporates context information to the C-value method, aiming to improve multi-word term extraction in general.

Table 4. Some New Tuples Founded by Our System

Hypernym	Hyponym
Europe	UK
Computer Firm	Texas instruments
Remote users	Portable pcs
Applications developer	Lotus development corp
Word processors	Microsoft word
Laser printer	HP laserjet III

Our system checks the occurrence of the word in the concept list from C-value /NC-value. Then 5-tuple $t =< l_c, tag_1, m_c, tag_2, r_c >$ is generated using left, middle, right context. A candidate tuple is generated if there is a pattern t_p such that $Match(t, t_p) \geq \tau_{sim}$ where τ_{sim} is the clustering similarity threshold. Snowball evaluated confidence of a pattern P[7], which becomes higher when a pattern P generate candidate tuples which exactly match with a candidate tuple in the previous candidate tuples list. The value becomes lower if part of them matched with a candidate tuples in a previous list. If a confidence value of candidate tuple T is higher than τ_t T becomes new tuple and used in next iteration. With new tuples, the process of 2.1 and 2.2 are repeated. New tuples in hyponymy are listed in Table 4. We will discuss our experiment result in the next section.

III. Experiment

We describe the document collection that we used for experiments and parameter values we used and compare the experimental result.

A. Experimental Setting

We use Ziff document set(Information from Computer Select disk,1989-1990, copyrighted by Ziff Davis) offered by TREC. Ziff document set contains 785 files, and about 800MB size. Among them, we selected about 101,000 sentences which contain 268,610 words.

Table 5. A Sample of Ziff document

<DOC>
<DOCNO> ZF32-244-004 </DOCNO>
<DOCID>09 754 449</DOCID>
<JOURNAL>Computerworld Jan 14 1991 v25 n2 p1(2)</JOURNAL>
<TITLE>3Com cuts back net plans. (3Com Corp.) (abandoning network
operating system business)</TITLE>
<AUTHOR>Keefe, Patricia; Nash, Jim.&M;</AUTHOR><TEXT>
<ABSTRACT>3Com Corp abruptly announces its intention to withdraw from the
LAN operating systems market and focus its efforts on multivendor connectivity products.
</ ABSTRACT></TEXT>
<DESCRIPT>
Company: American Telephone and Telegraph Co. (Communication systems)
Ticker: COM
Topic: Fiber optics
Data Communications
T3 Communications
Feature: illustration
Caption: Higher in fiber.</DESCRIPT>

Parameter value used for this experiment is listed in Table 6. τ_{sim} is minimum degree of match of Match function in 2.1. If τ_{sim} is too high, a lot of cluster generated and dropped because $\tau_{\sup}$ value. If τ_{sim} is too low, patterns loss their identity. If the number of element in a cluster is smaller than $\tau_{\sup}$, pattern (centroid of the cluster) is also dropped. W_{middle} is the weight of the middle context. Because the context between two tags is more important than left and right context of tags, W_{middle} is higher than weight of other context. These weight values are applied when we calculate match degree in Match function. Window(m) is the maximum number of terms in middle vector. Window(m) is bigger than other window size to consider the case that distance between tag1 and tag2 is far.

Table 6. Parameter Values for Experiments

Parameters	Value	Description
τ_{sim}	0.6	Minimum degree of match
τ_t	0.7	Minimum tuple confidence
$\tau_{\sup}$	2	Minimum pattern support
$I_{\max}$	3	Number of iterations of Snowball
W_{middle}	0.6	Weight for the middle context
W_{left}	0.2	Weight for the left context
W_{right}	0.2	Weight for the right context
Window(m)	7	Maximum number of terms in middle vector
Window	2	Maximum number of terms in other vectors

B. Experimental Result

In this evaluation, we used concept list from C-value/NC-value method. We selected top 10,000 concept list which is arranged by the value of importance. Simplifying pattern system finds occurrence of concepts in the list instead of named-entity tagger.

Table 7. A Sample of Concept List

Hard disk
San jose calif
Microsoft window
Lan manager
Apple computer inc
Personal computer
…

To measure the precision, we asked 5 persons who are familiar with computer domain, and calculated average precision value of their answer

As shown in Table 8 generalized pattern method finds more tuples than Snowball. This is because that our method simplified patterns and these patterns contain concise information. This effects the calculation of match degree. Thus more tuples are selected than Snowball.

Table 8. Number of New Tuples (τ_t = 0.7, the number of sentences = 100,000, the first iteration)

	Snowball	Generalized Pattern
Number of New Tuples	39	203
Precision	30.76 %	36.84%

Though our system is designed to iterate several times, performance became worse as it iterates. This is because that the initial tuples are generally accurate and makes valid patterns at the first iteration. After the first iteration, wrong tuples are also input to the system and generate invalid patterns. The performance will be better if we refine the result of the first iteration.

IV Conclusion

Variation of patterns of the relation and many kinds of relations makes difficulties in finding relations from a text document. To enfeeble this, we have presented generalizing pattern approach. In particular, we have generalized the context between terms in hyponymy and made patterns more general and concise. Our contribution is to use generalized patterns to recognize various context forms of the relation in a sentence. The patterns are automatically generated according to the initial information from a user. Experimental result show that our method outperforms in finding terms in the relation.

In further work, we plan to apply general constraints for the relations to increase precision. It also remains further work to find ideal initial tuples and defining kinds of terms in relations. The one of important parts of this method is initial condition. In Snowball, the initial tuples are very general and occur several times in sentence within location and organization relationship. In case of other relations, it is not easy to find initial tuples which occur in a sentence within specific relation. Due to the fact that the kinds of terms related to a relation are not limited, precision is not good. If we apply general constraints for the relations, the performance will be improved.

References

[1] Hetzler,B., Beyond word relations. SIGIR Forum, 21/2, 28-33, 1997
[2] M.A.Hearst, Automatic acquisition of hyponyms from large text corpora , in Proceedings of the 14th international Conference on Computational Linguistics, 1992
[3] D. Alan Cruse, Hyponymy and its variety, The Semantics of Relationships An Interdisciplinary Perspective, 3-21, 2002
[4] Philipp Cimiano, Lars Schmidt-Thieme, Aleksander Pivk, Steffen Staab, Learning Taxonomic Relations from Heterogeneous Evidence, in Proceedings of the ECAI2004 Ontology Learning and Population Workshop, 2004
[5] F.Ciravegna, Adaptive information extraction from text by rule induction and generalization, in Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCA I2001), (2001)
[6] Lyons, J., Introduction to Theoretical Linguistics, Cambridge University Press, 1968
[7] Eugene Agichtein and Luis Gravano. “Snowball: Extracting Relations from Large Plain-Text Collections”, In Proceedings of the ACM International Conference on Digital Libraries (DL’00), 2000.
[8] Hang Cui, Min-Yen Kan, Tat-Seng Chua, Unsupervised Learning of soft patterns for generating definition, in Proceedings of 13th International World Wide Web Conference, 2004
[9] Eric Brill, Brill Tagger http://www.ling.gu.se/~lager/Home/brilltagger_ui.html
[10] William, B.Frakes, Ricardo Baeza-Yates, Information Retrieval: Data Structures and Algorithms. Prentice-Hall. 1992

[11] Katerina T. Frantzi, Sophia Ananiadou, and Junichi Tsujii, The C-value/NC-value Method of Automatic Recognition for Multi-word Terms, Research and Advanced Technology for Digital Libraries: Second European Conference, ECDL'98, 1998
[12] Bagga, Amit and Breck Baldwin, Lingpipe, http://www.alias-i.com/lingpipe/index.html
[13] H. Kim, S.-g. Lee, Discovering Taxonomic Relationships from Textual Documents, Proceedings in IASTED Applied Informatics, 2002
[14] Sharon A.Caraballo, Automatic construction of hypernym-labeled noun hierarchy from text, in Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 120-126, 1999