Building a Business Knowledge Base by a Supervised Learning and Rule-Based Method

저자
(Authors)

Sungho Shin, Hanmin Jung, Mun Yong Yi

출처
(Source)

KSII Transactions on Internet and Information Systems(TIIS) 9(1), 2015.1, 407-420 (14 pages)

발행처
(Publisher)

한국인터넷정보학회
Korean Society For Internet Information

URL

http://www.dbpia.co.kr/Article/NODE06139374

APA Style

Sungho Shin, Hanmin Jung, Mun Yong Yi (2015). Building a Business Knowledge Base by a Supervised Learning and Rule-Based Method. KSII Transactions on Internet and Information Systems(TIIS), 9(1), 407-420.

Building a Business Knowledge Base by a Supervised Learning and Rule-Based Method

Sungho Shin[1, 2], Hanmin Jung[1] and Mun Yong Yi[2]
[1] Department of Computer Intelligent Research, Korea Institute of Science and Technology Information
Daejeon, 305-806 - South Korea
[e-mail: {maximus74, jhm}@kisti.re.kr]
[2] Department of Knowledge Service Engineering, Korea Advanced Institute of Science and Technology
Daejeon, 305-701 - South Korea
[e-mail: munyi@kaist.ac.kr]
*Corresponding author: Mun Yong Yi

Received October 26, 2014; revised December 5, 2014; accepted December 9, 2014; published January 31, 2015

Abstract

Natural Language Question Answering (NLQA) and Prescriptive Analytics (PA) have been identified as innovative, emerging technologies in 2015 by the Gartner group. These technologies require knowledge bases that consist of data that has been extracted from unstructured texts. Every business requires a knowledge base for business analytics as it can enhance companies' competitiveness in their industry. Most intelligent or analytic services depend a lot upon on knowledge bases. However, building a qualified knowledge base is very time consuming and requires a considerable amount of effort, especially if it is to be manually created. Another problem that occurs when creating a knowledge base is that it will be outdated by the time it is completed and will require constant updating even when it is ready in use. For these reason, it is more advisable to create a computerized knowledge base. This research focuses on building a computerized knowledge base for business using a supervised learning and rule-based method. The method proposed in this paper is based on information extraction, but it has been specialized and modified to extract information related only to a business. The business knowledge base created by our system can also be used for advanced functions such as presenting the hierarchy of technologies and products, and the relations between technologies and products. Using our method, these relations can be expanded and customized according to business requirements.

Keywords: Information extraction, business knowledge base, structural support vector machine, named entity recognition, relation extraction

This work was supported by the IT R&D program of MSIP/KEIT. [2014-044-024-002, Developing On-line Open Platform to Provide Local-business Strategy Analysis and User-targeting Visual Advertisement Materials for Micro-enterprise Managers]
A preliminary version of this paper was presented at APIC-IST 2014 and was selected as an outstanding paper. This version includes a concrete analysis and supporting implementation results on building a business knowledge base.

http://dx.doi.org/10.3837/tiis.2015.01.025 **ISSN : 1976-7277**

1. Introduction

Information extraction is highly helpful in detecting useful information presented in texts by collecting, storing, and analyzing them. For this purpose, texts are subject to a series of processes, such as splitting them into sentences and tokens, analyzing the meaning of each token to recognize useful named entities, and extracting the relation between these entities. Some exemplary technologies used for information extraction include language processing, text mining, data mining, and machine learning. Natural Language Question Answering (NLQA) and Prescriptive Analytics (PA) are the latest information extraction technologies that recently appeared in the hype cycle for emerging technologies, prepared by Gartner. The process of information extraction involves sentence splitting, tokenization, Part of Speech (PoS) tagging, parsing, feature extraction, machine learning (or rule-based), Named Entity Recognition (NER), and Relation Extraction (RE). Through these processes, a knowledge base for in-depth data analysis and intelligent services such as NLQA and PA can be built. However, conventional information extraction has been used for NER that mainly focused on person name, location name, organization name, and RE, especially in the biological field. Only few companies have used this technology to build their business knowledge base to provide data for intelligent services. Many researchers in this field have made efforts to find new information extraction methods and improve the performance of algorithms developed by them. Because of this, limited interest has been shown in the extraction of useful information related to businesses, including competition between products manufactured by companies, competition between technology, and relation between products and technologies. Thus, it is necessary to study how information extraction can be applied to the analytics of product or technology. This study focuses on extraction of information related to businesses, and applies supervised learning and rule-based methods to create a business knowledge base. In particular, the types of named entities include product name and technology name as well as person name, location name, and organization name. For RE, seven relations that exist between product name and technology name are extracted for business purposes.

2. Related Work

Information extraction can be defined as the task of automatically extracting useful information from unstructured or semi-structured documents. It has many subtasks with the most general ones being NER and RE. NER has been implemented using Conditional Random Fields [1] and Averaged Perceptron [2]. Most studies in NER are recently about how to add global features [3]. Many researches on RE have focused on how to use Maximum Entropy (ME) and Support Vector Machine (SVM). They have also explored how to use the non-linear kernel of SVM [4]. A recent study [5] discuses about a distant supervision method of automatically building training data to use machine learning in order to reduce the cost of building training data. Various learning algorithms and learning speed improvements have also been part of the study. The results of these studies have been published and applied in many fields. For the structural SVM used in this study [6], the 1-slack structural SVM and the cutting-plane algorithm have been modified and applied together to enhance learning speed.

There are many rule-based information extraction systems that are available for building business knowledge bases [7-8]. They were used for the extraction process from the very beginning of information extraction study, but some systems have recently taken advantage of

merging rules and machine learning methods. The state-of-the-art of performance in relation extraction is about F1 score of 72.1 which is achieved by using a composite kernel that consists of an entity kernel and a convolution parse tree kernel [9].

Recently, many researchers have begun focusing on resolving technical and content characteristics issues in this area such as context generalization to reduce data sparsity, long context, disambiguate fine-grained types, and parsing errors. Until the mid-2000, researchers have mainly used MUC (Message Understanding Conference) and ACE (Automatic Content Extraction) corpus that contains 5 relation types and 24 subtypes. This extracts various relations among person, location, and organization [10]. The 5 relations are: At, Near, Part, Role, and Social. However, research on what actually needs to be extracted has been very limited, especially in areas such as business relations for more practical purposes. This paper focuses on information extraction for business relations between technology name and product name. This is an area that researchers have not studied in-depth, but is important for business analytics.

3. Process and Data

3.1 Process

The proposed system is based mainly on a supervised learning method for information extraction. This method analyzes word features, positional features, and lexical features on each keyword in documents and classifies them into predefined types. This method follows a two-step process: learning for creating statistic information required by the correct answer collection and recognition by using a learning model to extract information from documents.

The training data is subject to textual analysis, for example, morphological analysis and structural analysis of sentences. Feature extraction is then performed to extract word features, morphological features, and syntactic features to use them as features for NER and RE. In the learning process, the extracted feature values are used to generate entity models and relation models. In the NER and RE process, the extracted feature values and the models built through learning are used to recognize a specific keyword as an entity, or to extract relations between entities. After information extraction, filtering is performed to enhance the results. Filtering is performed by applying rules specialized to business information.

Fig. 1 below shows the overall process of building a business knowledge base through information extraction.

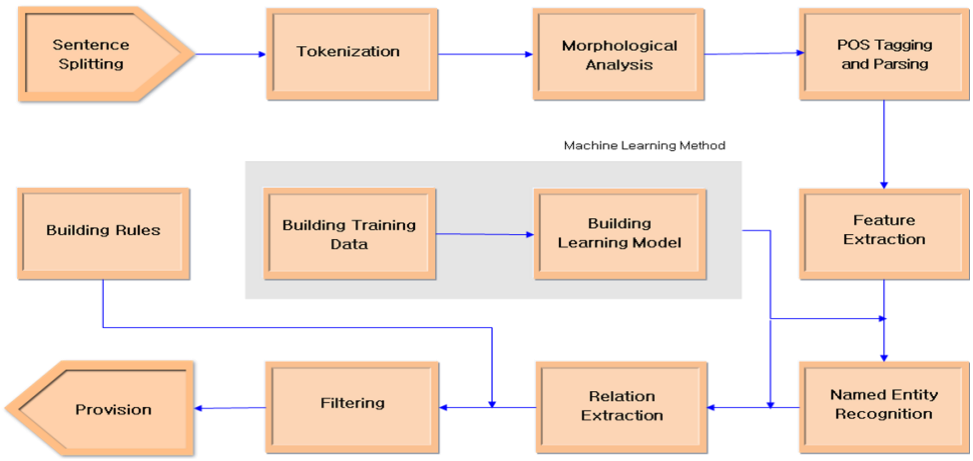

Fig. 1. Process of building the knowledge base

3.2 Input Data

The information extraction system in this study aims at extracting useful information from unstructured text documents. Unstructured text documents can be classified into various categories, but the input documents of this study are limited only to papers, patents, and web articles as listed in **Table 1**. These documents belong to the science and technology domains. The fields of science and technology encompass the entire disciplines except for humanities, social studies, art, and sports. For paper and patent, the documents are collected from KISTI-NDSL providing an information service for science and technology papers and patents. The web articles are collected from the science and technology category. These articles have been collected from popular websites such as the New York Times, Thomson Reuters, and BBC. However, articles from blogs or personal homepages are excluded. These documents express personal views of writers, who are not responsible for their contents. Therefore, the contents of the web articles that we have used are quite reliable.

Table 1 illustrates the size and type of documents that we want to analyze. Information on the type of documents we want to analyze is one of the important factors in designing an information extraction system. This factor plays a vital role because the extraction environment needs to be changed according to the characteristics of the documents.

Table 1. Summay of input data

Document Type	Domain	Part	Period (year)	# Document
Paper	Science and Technology	Title & Abstract	2001~2012	4,093,516
Patent	Science and Technology	Title & Abstract	2001~2012	8,486,300
Web article	Science and Technology	Title & Body	2001~2013	5,261,883

3.3 Output Data

We aim to extract 5 types of named entities and 7 types of relations. The types of named entities are Person Name, Location Name, Organization Name, Technology Name, and Product Name. Location Name is divided into Nation Name and City Name. Organization Name has also subtypes such as Company Name, Institution Name, and University Name.

Since the definition of named entity types can vary from person to person, the exact definition for each type of entity is required. In particular, technology name, because it appears similar to product name, but the two entities have different meanings and hence, must be differentiated.

Table 2 lists the definitions and examples of Person name, Location Name, Organization Name, Technology Name, and Product Name which are mainly addressed as output data types in this study [7]. Our system is somewhat different from other information extraction systems as it covers business terms such as Product Name and Technology Name. These are not general named entities that are extracted in information extraction, but are specialized for our business knowledge base.

Table 3 lists the definitions for 7 relations between product name and technology name. Each relation contains arguments and constraint as listed in **Table 4**. For example, productConsistTechnology relation has product name as the subject and technology name as the object. In this relation, the constraint is ‘directional’, which means that technology name should not be the subject and product name should not be used as the object as least in this relation.

Table 2. Entity types and description

Types	Description	Example
Person	People who work for organizations or do activities (including research) related to products or technology production.	Barack Obama, Steve Jobs, Eric Schmidt
Location	Countries and regions where organizations are located.	South Korea, California
Organization	Organizations of producing and selling technology, products, etc., organizations or institutions established for roles and goals.	Hynix, Apple Inc.
Technology	Method of developing tools, machines or materials people need, and producing processes or products to use them.	Smartphone, Mobile device, Fuel cell, Java, E-book, Tablet PC
Product	Articles, for example, models or series implemented by using technology in corporations.	iPad, iPad 2

Table 3. Description on each relation names

Relation name	Description
partOfProduct	Product A is one of different products used for producing product B.
competeProduct	Products A and B have similar purpose and functions, and are competing each other in the market. They can replace each other.
similarProduct	Although products A and B have similar features in the same type of business in the market, they do not compete each other. They cannot replace each other, and are used independently.
elementOfTechnology	Technology A is one of detailed technologies which are components of Technology B.
competeTechnology	Technology A and B have similar purpose and functions, and are competing in the market. They can replace each other.
similarTechnology	Although technologies A and B have similar features in the same field of the market, they do not compete each other. They can not replace each other, and are used independently.
productConsistTechnology	There are different technologies used to make product A, and technology B is one of them.

Table 4. Arguments and constraint of relations

Relation name	Subject Type	Object Type	Constraint
partOfProduct	Product	Product	Directional (A→B)
competeProduct	Product	Product	Bi-directional (A↔B)
similarProduct	Product	Product	Bi-directional (A↔B)
elementOfTechnology	Technology	Technology	Directional (A→B)
competeTechn-ology	Technology	Technology	Bi-directional (A↔B)
similarTechno-logy	Technology	Technology	Bi-directional (A↔B)
productConsistTechnology	Product	Technology	Directional (A→B)

4. Machine Learning Environment

4.1 Textual Analysis

Textual analysis is the step before information extraction. It consists of sentence splitting, tokenization, morphological analysis, and parsing.

a. Sentence splitting and tokenization
Before the textual analysis, each document is split sentence by sentence. The sentence separation is done by using "new line" in a document. Patterns are made based on:

- The head of a sentence is starting with capitals or double quotation marks or their combination).
- The end of a sentence is starting with period, or question marks, or exclamation marks, or double quotation marks, or their combination).
- Exceptions of sentence separation are considered such as periods which come after Mr, Mt, and Dr.

b. Morphological analysis and parsing
A morpheme is the smallest grammatical unit in a sentence. Morphological analysis breaks down each word into morphemes and analyzes which PoS each morpheme belongs to. In English, morphemes are divided by spaces. Morphological analysis analyzes to which PoS each morpheme belongs among nouns, verbs, adjectives, etc.

Parsing analyzes the entire structure of a sentence, its elements such as subject, object, etc., and their relation. For the analysis, the result of morphological analysis is integrated into phrases such as phrasal nouns and phrasal verbs. This is done to analyze the dependence between phrases on the basis of the morphological analysis results. In this study, the Stanford Tagger and the Stanford Parser, which are open sources, are used for morphological analysis and parsing. The results are used as basic information in the feature extraction step that is used as features.

4.2 Feature Extraction

Features normally consist of the following:

- The morphological analysis results
- Syntactic analysis that have been obtained through the textual analysis
- The result of word information from each keyword in sentences

The features are classified into entity features for recognizing named entities and relation features for relation recognition between entities. For NER, 37 features including word features, local features, and external features are used. For example, the current token starts with capital, digit pattern, uppercase, token length, ngram character, and so on as listed in **Table 5**. For RE, 24 features are used including context features around entities and syntactic structural features for relation instances as listed in **Table 6**.

Table 5. Features for NER

Criteria	Features
Word features	If starting with capital
	If being expressed in all capital
	If consisting of both uppercase letters and lowercase letters
	If ending with a period
	If having period(s) between letters
	If having apostrophe(s) between letters
	If having hyphen(s) between letters
	Normalized digit letters in a row
	Ordinal numbers
	If consisting of both alphabetical letters and digit letters
	If having possessive expressions of the possessive
	First person pronoun
	Stem for current token
	Lemma for current token
	If ending with clue expressions useful for assuming certain entity type, for example –ist and -ish.
	If extracting only alphabet letters
	If extracting non-alphabet letters
	N-grams
	Expression after converting to lowercase letters
	Expression after converting to uppercase letters
	Expression after normalization (allowing duplication of letters)
	Expression after converting to normalizing letters (not allowing duplication of letters)
	Length of current token
	POS
Local features	Length of the phrase containing current token
	Lists of two tokens before and after current token
	If previous token is ‘from’
	If previous token is ‘by’
	If previous token is ‘and’
External features	If included in the stop-word dictionary
	If included in the corporation dictionary
	If included in the institution dictionary
	If included in the nation dictionary
	If included in the person dictionary
	If included in the product dictionary
	If included in the technology dictionary
	If included in the university dictionary

Table 6. Features of RE

Criteria	Features
Context features	Expression word for each token
	Lemma word for each token
	Part of speech for each token
	Expression words for tokens existing between entities
	Word bigram for all terminal nodes of the path-enclosed tree
	POS bigram for all prior terminal nodes of the path-enclosed tree
	Information on the path connecting two entities in the parsing tree
	If existence of two entities in the same NP
	If existence of two entities in the same PP
	If existence of two entities in the same VP
	Word collection (bag-of-words) of extracted entities
	Word bigram of each entity in a sentence
Syntactic structural features	Combine entity 1 with entity 2
	Combine the type of entity 1 with the type of entity 2
	If no word between two entities
	Specifying the word where there is only one word between two entities
	Specifying the first word among the words where the number of words between two entities is not less than 2
	Specifying the last token among the words where the number of tokens between two entities is not less than 2
	Token that appears before the first entity
	Token that appears next to the second entity
	Dependency tree token bigram
	Bigram in a dependency tree format
	Information on a path connecting two entities in the mixed tree
	Clue words that appear in the sentence

4.3 Machine Learning Algorithm

The machine algorithm used for information extraction in this study is structural SVM [6]. This algorithm extends the existing SVM algorithm. While the existing SVM implements binary classification and multiclass classification, the structural SVM implements a more general structure. For example, sequence labeling and syntactic analysis. In this study, Pegasos algorithm that is applied to the SVM for high performance and fast learning speed is selected from Stochastic Gradient Decent (SGD) methods, extended, and used for structural SVM learning. **Fig. 2** shows the Pegasos algorithm modified for structural SVM learning. This algorithm receives algorithm iteration frequency T and learning data number k as input for calculating a sub-gradient. The vector w1 is initially set as any vector value less than $1/\sqrt{\lambda}$. For iteration frequency is t, size of At is k selected from entire learning data (row 4) and the most violated named entity tag is then obtained from the learning data in At (row 5). After establishing a learning rate (row 6), wt+1/2 is then obtained (row 7) to set the vector for

projecting wt+1/2 onto the collection {w:|w| than $1/\sqrt{\lambda}$ } as wt+1 (row 8). The result of the algorithm is wT+1 and the average vector waveraged (row 11).

1: Input: S, λ, T, k

2: Initialize: Choose $\mathbf{w}_1$ s.t. $\|\mathbf{w}_1\| \le 1/\sqrt{\lambda}$, $\mathbf{v} = 0$

3: For $t = 1,2,\ldots,T$

4: Choose $A_t \subseteq S$, where $|A_t| = k$

5: $\forall (\mathbf{x}_i, \mathbf{y}_i) \in A_t : \hat{\mathbf{y}}_i = \arg\max_{\mathbf{y}} \{L(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}^T \mathrm{f}(\mathbf{x}_i, \mathbf{y})\}$

6: $\eta_t = 1/\lambda t$

7: $\mathbf{w}_{t+1/2} = (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in A_t} \{\mathrm{f}(\mathbf{x}_i, \mathbf{y}_i) - \mathrm{f}(\mathbf{x}_i, \hat{\mathbf{y}}_i)\}$

8: $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+1/2}\|}\right\} \mathbf{w}_{t+1/2}$

9: $\mathbf{v} = \mathbf{v} + \mathbf{w}_{t+1}$

10: $\mathbf{w}_{\text{averaged}} = \mathbf{v}/T$

11: Output: $\mathbf{w}_{T+1}$ and $\mathbf{w}_{\text{averaged}}$

Fig. 2. A modified Pegasos algorithm for NER

4.3 Building Training Data

Training data is a collection of document in which named entities to be recognized are tagged for learning the models. The test collections such as MUC and ACE, for evaluating the performance of information extraction systems, are generally used for research, or training data can be built for a specific purpose by researchers.

For building training data for a special purpose, either domain experts can be hired for building it manually or automated methods can be used to save time and cost. In this study, simplified distant supervision method is used to automatically build initial training data (silver standard) [11-12]. We built this training data by collecting sentences. In order to gather the sentences, seed data containing named entities and their relation were listed in advance. This list of seed data was used as keywords for searching the web and extracting sentences that include the relevant seed words. After building the silver standard, domain experts are hired to enhance the accuracy through manual verification and finally build the gold standard. A supporting tool for manual annotation was provided to the domain experts during the verification stage to improve the verification efficiency. The training data was built for NER and RE. The number of sentences in the training data was 31,273 for named entities and 8,382 for relations. We divided them into two groups: for training and for test. Training was conducted using 90% of the data and test using the remaining 10%.

5. Result and Discussion

We used F1 score to evaluate the accuracy of our system. F1 score is commonly used to evaluate information extraction systems. The score is the harmonic mean of precision and recall, ranging from 0 to 100. A high score indicates high accuracy. The result of the evaluation is shown in **Table 7** and **8**. The overall F1 score of NER is 74.61 and that of business RE is 70.92. The scores of each type are distributed around the average (**Fig. 3** and **4**).

The highest score in NER is for the sub-type university name and the lowest is for product

name. It is clear that the system performs well for the extraction of general entity types such as Person, Location, and Organization, as their average F1 score is over 81. However, the new entity types, Technology and Product, were less likely to be extracted correctly. The F1 score for both entities is about 65. To the best of our knowledge, this is because the two entities are very similar and hence, difficult to distinguish from one another. In a sentence, there can be a high degree of similarity in the feature values and clue words of these entities. For example, "mobile operating system" is a technology name and "android" is a product name. They both belong to a technology and product hierarchy. The top of this hierarchy can be "computer system." "Operating system" is a "computer system" and likewise "mobile operating system" is an "operating system." They are all technology names based on the definition (**Table 2**). There are many kinds of "mobile operating systems" and "android" is one of them. We define "computer system" and "mobile operating system" as technology name in the hierarchy, while "android" is defined as a product name. This is obtained from the definition provided in **Table 2** that states that a product is implemented using a technology in corporations. In order to distinguish between them accurately, highly sophisticated training corpus is required for machine learning. Machine learning requires such training data to classify these entities because it is not as accurate as humans for intelligent information extraction. However, it requires a considerable amount of time and labor. Here, we simply add some rules specialized for this task. Building a sophisticated training corpus is part of our future work.

Table 7. Performance of NER

Type	Sub-Type	Precision	Recall	F1 score
Person	Person	75.74	84.72	79.98
Location	Nation	77.58	81.60	79.54
Organization	University	91.15	90.45	90.80
	Corporation	82.56	77.64	80.03
	Institution	74.11	74.21	74.16
Business	Technology	75.10	57.67	65.24
	Product	72.26	58.71	64.79
Total		79.21	70.51	**74.61**

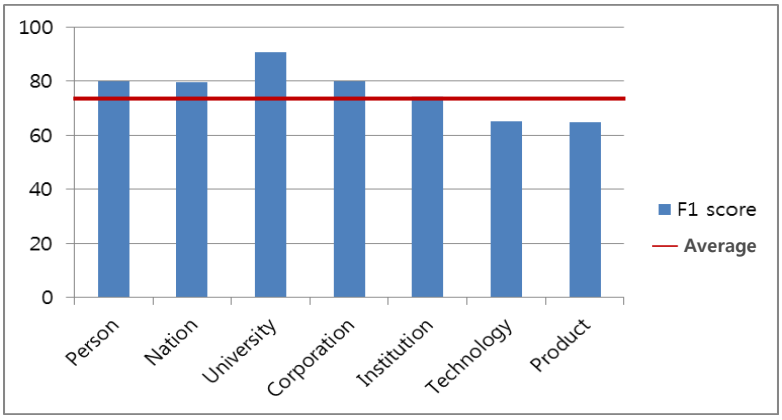

Fig. 3. Comparison of each type in NER task

Overall F1 score of the relation extraction test is about 71. This means 29% of relation instances extracted by the system may be wrong. Error data from NER affects the result of relation extraction following NER. F1 score of competeTechnology relation is significantly low. This is because the number of extracted relation instances for the competeTechnology relation from texts is very small. Many technology names may be recognized as product names and consequently cause this problem. Among the relation types, the two entity types that need to be distinguished from each other are competeTechnology and productConsistTechnology. If these two relations are not distinguished correctly, overall test results are affected.

Table 8. Performance of business RE

Type	Precision	Recall	F1 score
partOfProduct	59.91	66.27	62.93
similarProduct	68.45	79.25	73.45
competeProduct	88.79	82.03	85.27
elementOfTechnology	63.01	93.43	75.26
similarTechnology	62.02	85.78	71.99
competeTechnology	100.00	4.17	8.00
productConsistTechnology	53.65	50.49	52.02
Total	66.59	75.85	**70.92**

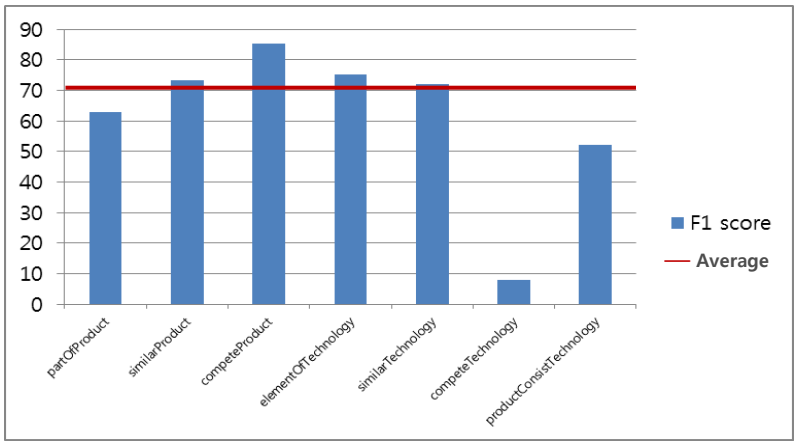

Fig. 4. Comparison of each type in RE task

Regardless of the score, errors in information extraction should be fixed because the extracted instances will be used for real analytics services. We know it is not possible to get rid of all types of errors in information extraction systems. We cannot control the result of prediction because it is performed automatically by the statistical model. To fix these errors, we can make rules that can be applied when the system performs these predictions for entities and relations. We prepared two types of rules: rules for NER and rules for RE. First, it is required to improve the accuracy of NER. Technology name and product name that include special characters such as '#', '$', '&', and, '?' have higher probability of being recognized incorrectly. We can make rules that filter out such names from the extracted NER instances. In addition, casting rules

with a casting dictionary that fix the type of entity instances are defined and applied as well. For example, Google glass is a product name. No matter what type the statistical model predicts Google glass as, the system classify it into product name by using a casting rule. Relation rules are built from a relation dictionary. Each instance in the relation dictionary is composed of subject, object, and relation. Each relation instance in the dictionary needs to have as many variations as possible. The relation dictionary can be used for defining casting rules for relations. The casting rules for relations are executed before the system stores the results.

With the supervised learning and rule-based method, our system extracted many instances of relations from texts. The number of relation instances extracted from all the resources we collected is presented in **Table 9**.

Table 9. Number of relation instances

Relation name	# instances
competeProduct	1,168,747
competeTechnology	14,997
elementOfTechnology	581,711
partOfProduct	1,837,494
productConsistTechnology	529,433
similarProduct	1,855,153
similarTechnology	1,211,775
Total	**7,199,310**

The most extracted relation types are partOfProduct and simlarProduct (Fig. 5). These relations are both related to *product name*. This means that there are much more mentions about products and their relation, especially their components or their competitors, in documents. The ratio of extracted relation instances does not follow the recalls of relation types. This is because the ratio of relation types in evaluation is only from the limited training corpus, while the extracted relation instances are from the real documents that we target. The most unextracted relation is *competeTechnology*, which does not means that there are only few *competeTechnology* relations in document. We do not know how many mentions about the competition between technologies are in documents. We can find the reason why this result happens from the recall. The recall of *competeTechnology* is just 4.17%. Therefore, we just assume that our machine learning model is weak in extracting *competeTechnology* relation from documents.

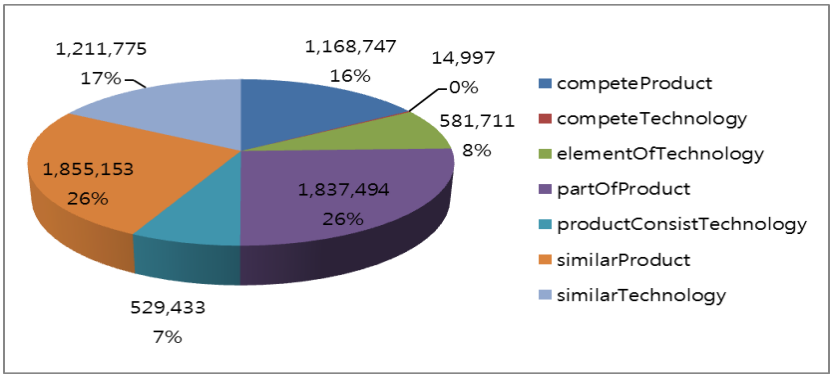

Fig. 5. The ratio of relations

Our method automatically builds a business knowledge base for business purposes. It is almost impossible to extract useful information from massive text data and make knowledge base for business purposes manually. This can be automated using our system. Even though there are some errors, the proposed system is quite useful and has several applications. First, it helps make a hierarchy for technologies and products. This hierarchy is useful for the process of information extraction. Second, it is possible to see competitive technologies (or products) against certain technologies (or products). Third, a company can realize its competitors who have similar or competitive technologies or products in the industry. Last, the system provides companies with useful information while developing new technologies or products. The knowledge base built by the system describes current competitive technologies and products and also the technologies to be focused on in the future.

6. Conclusion

In this paper, we present a supervised learning and rule-based method to automatically make a business knowledge base. This method is fundamentally based on information extraction, but different with existing ones. We set up a machine learning environment specialized for the business knowledge base and applied casting rules to improve the performance of NER and RE. The evaluation is F1 score 74.61 and 70.92 for RE, while the error data can be fixed by rules for business purposes. We expect that other researchers and engineers will benefit from the proposed method when they try to build their business knowledge base.

References

[1] A. McCallum and W. Li, "Early Results for Named Entity Recognition with Conditional Random Fields, features Induction and Web-Enhanced Lexicons," in *Proc. of Conference on Computational Natural Language Learning*, May 31-June 1, 2003.

[2] M. Collins, "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms," in *Proc. of Empirical Methods in Natural Language Processing*, July 6-7, 2002.

[3] D. Nadeau and S. Sekine, "A Survey of Named Entity Recognition and Classification," *Linguisticae Investigations*, vol. 30, pp. 3-26. 2007. Article (CrossRef Link)

[4] N. Bach and S. Badaskar, "A survey on relation extraction," *Language Technologies Institute, Carnegie Mellon University*, 2007.

[5] M. Mintz, S. Bills, R. Snow and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. of the Association for Computational Linguistics*, August 2-7, 2009.

[6] C. Lee, P. M. Ryu and H. K. Kim, "Named Entity Recognition using a Modified Pegasos Algorithm," in *Proc. of the 20th ACM International Conference on Information and Knowledge Management*, October 24-28, 2011.

[7] S. Shin, C. H. Jeong, D. Seo, S. P. Choi and H. Jung, "Improvement of the Performance in Rule-Based Knowledge Extraction by Modifying Rules," in *Proc. of the 2nd International Workshop on Semantic Web-based Computer Intelligence with Big-data*, November 9-11, 2013.

[8] C. N. Seon, J. H. Yoo, H. Kim, J. H. Kim and J. Seo, "Lightweight Named Entity Extraction for Korean Short Message Service Text," *KSII Transactions on Internet & Information Systems*, vol. 5, no. 3, pp. 560-574, 2011. Article (CrossRef Link)

[9] M. Zhang, J. Zhang, J. Su and G. Zhou, "A composite kernel to extract relations between entities with both flat and structured features," in *Proc. Of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 825-832, July 17-21, 2006.

[10] Z. Guodong, S. Jian, Z. Jie and Z. Min, "Exploring various knowledge in relation extraction," in *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 427-434, June 25-30, 2005.
[11] M. Mintz, S. Bills, R. Snow, D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, August 2-7, 2009.
[12] S. Shin, Y. S. Choi, S. K. Song, S. P. Choi and H. Jung, "Construction of Test Collection for Automatically Extracting Technological Knowledge," *Journal of Korea Content Society*, vol.12, no.7, 2012 (in Korean).

Sungho Shin is a senior researcher at Korea Institute of Science and Technology Information (KISTI) since 2002, and is with his Ph.D. degree at Korea Advanced Institute of Science and Technology (KAIST) from 2012. He received his B.S. and M.S. degree in Business Administration (Management Information Systems in detail) from Kyungpook National University (KNU), Korea in 2000 and 2002. Recently he has researched and developed information and event extraction system for intelligent systems. His current research interest includes information and event extraction, text mining and Natural Language Processing (NLP).

Hanmin Jung works as the head of the Dept. of Computer Intelligence Research and chief researcher at Korea Institute of Science and Technology Information (KISTI), Korea since 2004. He received his B.S., M.S., and Ph.D. degrees in Computer Science and Engineering from POSTECH, Korea in 1992, 1994, and 2003. Previously, he was senior researcher at Electronics and Telecommunications Research Institute (ETRI), Korea, and worked as CTO at DiQuest Inc, Korea. Now, he is also adjunct professor at University of Science & Technology (UST), Korea, visiting professor at Central Officials Training Institute (COTI), Korea, standing director at Korea Contents Association, director at Korean Society for Internet Information, director at Computer Intelligence Society, director at Korea Information Technology Convergence Society, and committee member of ISO/IEC JTC1/SC32. His current research interests include decision making support mainly based in the Semantic Web and text mining technologies, Big Data, information retrieval, human-computer interaction (HCI), data analytics, and natural language processing (NLP). For the above research areas, over 520 papers and patents have been published and created (confirmed by Google Scholar).

Mun Yong Yi is Professor and Chair of the Department of Knowledge Service Engineering at Korea Advanced Institute of Science and Technology (KAIST). Before joining KAIST, he taught at University of South Carolina as Assistant Professor (1998-2004) and (tenured) Associate Professor (2005-2009). He earned his Ph.D. in Information Systems from University of Maryland, College Park. His current research interests include technology adoption and diffusion, IT training and computer skill acquisition, user experience, knowledge engineering, and semantic Web. His work has been published in a number of journals including Information Systems Research, Decision Sciences, Decision Support Systems, Information & Management, International Journal of Human-Computer Studies, IEEE Transactions on Consumer Electronics, and Journal of Applied Psychology. He is a former associate editor of MIS Quarterly and a current associate editor of International Journal of Human-Computer Studies and a senior editor of AIS Transactions on Human-Computer Interaction.