Extract Reliable Relations from Wikipedia Texts for Practical Ontology Construction

Jin-Xia Huang[1,2], Kyung Soon Lee[2], Key-Sun Choi[3], Young-Kil Kim[1]

[1] Electronics and Telecommunications Research Institute, NLP Research Section, Automatic Speech Translation and Language Intelligence Research Department, Daejeon, Korea

[2] Chonbuk National University, Division of Computer Science and Engineering, CAIIT, Jeonju, Korea

[3] School of Computing, KAIST, Daejeon, Korea

{hgh, kimyk}@etri.re.kr, selfsolee@chonbuk.ac.kr, kschoi@kaist.ac.kr,

Abstract. A feature based relation classification approach is presented in this paper. We aimed to exact relation candidates from Wikipedia texts. A probabilistic and a semantic relatedness features are employed with other linguistic information for the purpose. The experiments show that, relation classification using the proposed relatedness features with surface information like word and part-of-speech tags is competitive with or even outperforms the one of using deep syntactic information. Meanwhile, an approach is proposed to distinguish reliable relation candidates from others, so that these reliable results can be accepted for knowledge building without human verification. The experiments show that, with the relation classification approach presented in this paper, more than 40% of the classification results are reliable, which means, at least 40% of the human and time costs can be saved in practice.

Keywords. Information classification, information extraction, feature-based, relatedness information, ontology building.

1 Introduction

Extracting relationships between entities from text is one of the most crucial issues to understand the semantic relations between entities and manage data in structural way [1]. The task of relation extraction is identifying relationships between two or more entities in given context. The arguments of the relationships can be named entities, noun phrases, domain specific terms, or events. The two related entities can be in the same sentence, in which case it is called intra-sentence relationship; or occur in different sentences but in same section or document, which is inter-sentence relationship. An intra-sentence relation can be explicit one or implicit one depends on the contexts of the two entities [2]. If there are constituents in a common syntactic structure with two entities explicitly convey a relation type, like "*consist of*" for *part-whole* relation, "*be a kind of*" for *isa* relation, it is an explicit relation. Otherwise it is implicit relation, like entities *car* and *window* in expression of "*car window*", for example [3]. Generally, relation extraction task can be separated to three steps – entity detection, relation detection and relation classification. Entity detection recognizes entities from contexts, relation detection extracts two related entities from texts and detects if they have relationship with each other, and relation classification classifies detected relations to certain relation types.

In this paper, aiming at building IT domain ontology from texts, we focus on the problem of relation classification on intra-sentence relation candidates. The arguments of the relations can be named entities like *Microsoft*; or general terms like *application*; or domain specific terms, like *Hopfield network*. The relation types include *isa*, *usedFor*, *produces*, and *provides*, which are predefined according to their frequencies in target IT domain. As a preprocessing, lexical patterns are used as
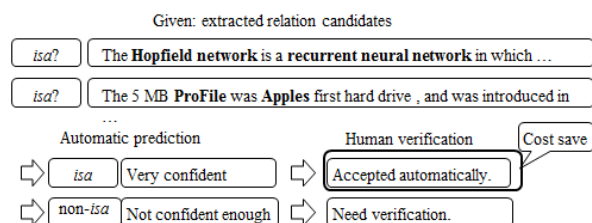
Fig. 1. Expecting working process in practice

filters to find explicit relation candidates for each relation type, so that the relation extraction problem can be transferred to a binary classification problem, with the precondition that the entities have been detected, and the extracted relation candidates can be either correctly or incorrectly.

The following examples show two relation candidates with their contexts, which are extracted with pattern "*be a*" and "*be*" for *isa* relation type, respectively.

– The Hopfield network *is a* recurrent neural network in which…,
– The 5 MB **ProFile** *was* **Apples** first hard drive, and was introduced in September 1981 at a price of…

From the context, we can see the first relation candidate is correctly detected, while the second one is not. These relation triples should be verified by human developers even after relation classification, to assure only the correct relation triples added to ontology. The task of this paper is classifying the relation candidates extracted with simple pattern matching approach from text, to predict if the candidates really hold the relation types. Confidence score given by the classifier is employed, and the prediction results with high confidence can be added to ontology directly without human verification. The process is as Figure. 1 in our expectation:

The contributions of this paper are as following:

– A feature-based approach for relation classification is presented, in which probabilistic and semantic relatedness information between patterns and relation types is proposed, and employed with lexical

features. The performance is competitive or outperforms some well-known features including syntactic ones.
– An approach is proposed to distinguish reliable predictions by using confidence score, which is normally provided by relation classifier. A significant percentage of human and time costs can be saved as the result.

The rest of the paper is organized as follows: Section 2 describes previous work. Section 3 gives the problem definition and outlines the general design of our approach. Section 4 describes in detail the features employed, and Section 5 presents the experimental evaluation. Section 6 contains conclusions and directions for future work.

2 Related Works

Relation extraction has gained increasing interests in recent years. Most of these works focused on relation extraction between named entities [4-7], and achieved significant progress especially according to the programs like Automatic Content Extraction (ACE)[1], in which annotated corpus are shared for evaluation and competition. Meanwhile, there are also increasing needs toward relation extraction and classification on general or domain specific terms for the purpose of knowledge building [8-11]. The latter task is more challenging for several reasons: 1) the semantic categories of the terms are more various compare to the named entities, which means the sense ambiguities of the terms are relatively high; 2) the relation types between terms are much diverse than the ones between named entities like human names, institutes, dates or addresses.

Supervised approaches have been broadly employed for relation extraction and relation classification [2-5, 10, 12-13]. Supervised approaches include feature-based approaches and kernel-based approaches. Kernel-based approaches compute similarities between parse trees or strings using different kernel functions [12]. Feature-based approaches investigate various features including lexicon, part-of-speech (POS)

---

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace.

information, syntactic information and semantic information to represent relation candidates, and classify the relations with vector space machines like support vector machines (SVM) [5, 7, 13], maximum entropy model (MEM) based classifiers [4], and deep neural networks (DNN) [14-15].

The performance of these feature-based models is strongly depended on the quality of the extracted features [15]. In feature-based approaches, it is reported that chunk information contributes more than deep syntactic information [5, 13]. The semantic features are also broadly employed in existing researches. For example, the semantic categories of the entities like Person, Country, and Organization are employed for named entity related relation extraction and classification [4-5,13]. But it is also reported that, for other types of the entities like general or domain specific terms, this kind of semantic information does not help much and can be even harm to the performance [11]. The reason is, as we mentioned above, that the terms have higher sense ambiguities, thus there are various semantic categories used in the feature expressions, which might cause data sparseness problem especially when we lack of training data. Zeng et al. [15] adopted word embeddings to transformed lexicon features to enhance the performance of relation classification (semantic role labeling).

In this paper, we adopt probabilistic and semantic relatedness features to reflect the relatedness between patterns and the relation types in an explicit way [16]. The relatedness information is acquired from both WordNet [17] – which is semantic relatedness information; and training corpus – which is probabilistic relatedness information. Our experiments show that the proposed relatedness features contribute to the classification performance in a significant way. We also utilize the well know features including word, POS and syntactic information which proposed in existing researches [4-5, 13].

In practical relation extraction for ontology building, human verification is still required for all cases as well as the accuracy of relation extraction is not comparable with the one of the human developers, and this is a very time and cost consuming part in practice. To solve this problem, this paper proposes an approach which utilizes confidence score provided by the classifier to tell reliable predictions, which results in the cost saving in a significant way.

3 Problem Description

This paper aimed to classify the explicit relationships between entities. The entities can be domain specific terms, noun phrases, and named entities. It is assumed that the entities and the relation candidates are already detected by a simple pattern matching approach, through which two entities are extracted while they occur in a common syntactic structure with other constituents match one of the predefined patterns.

Given a relation candidate with entities e_1 and e_2, which context W matches pattern p. What we want to predict is its relation type r:

$$f:(e_1, e_2, p, W) \rightarrow r.$$

The relation candidates and their contexts, with the patterns they matched, are represented with features, which features will be described in coming section, in feature extraction phase. Then they are put into the relation classifier to predict its relation type r. The relation classifier is trained with labeled data, which are relations and their contexts already verified by human annotators.

The relation type r can be one of *isa*, *usedFor*, *produces*, *provides*, and no-relation. No-relation means it is possible that the relation candidate does not hold any relation type in above. Considering each relation type already has its own patterns predefined, the multi-classification task can be transferred to a binary classification task, in which the relation type r is either 1 or 0. For certain relation type: 1 means given relation candidate holds certain relation type, 0 means it doesn't hold that type of relation. For example, to the relation candidates in Figure. 1, if the candidate holds *isa* relation, it should be classified to 1; otherwise 0.

The four relation types in this paper are selected according to their frequencies in IT domain. The procedure of how these relation types are selected are as following: several human annotators are required to extract all relation candidates from Wikipedia texts in IT domain; a series of relation types in ConceptNet [8] are given to the human annotators as reference, meanwhile it is also allowed that extra relation types can be

proposed/employed in necessary. As the result, several relation types are newly employed with existing relation types in ConceptNet, among them the most frequently used ones are as following:

– isa: can be a subclass relationship between two classes, or an instanceOf relationship between an instance and a class (it means, this paper does not distinct either a term is a class or an instance).
– usedFor: in a relation of "A usedFor B", domain A can be used for, or used in B.
– produces: "A produces B" indicates B is generated, created, or manufactured from empty, by A.
– provides: "A provides B", means B is an existing one, but offered, provided, or supported by A.

Not only the relation types, but also the lexical patterns are discovered by the human annotators during the procedure of relation annotation. Table 1 shows some of the examples:

4 Feature Selection

Feature selection is an important issue for feature based classification, because select what kind of features has strong impact on the classification performance. Most of the feature selection researches in relation classification field are only performed on named entity related relation types [4, 7, 13, 18]. This paper assesses the impacts of different features in the relation classification on general or domain specific terms. The employed features in this paper include word feature, POS feature, and syntactic feature. In addition, a new feature which reflects the relatedness information between patterns and relation types is also proposed. The relatedness information includes semantic and probabilistic relatedness information, which can be acquired from WordNet and corpus, respectively.

The features computed in this paper are described below, with an example of parse tree given in Figure. 2 for a sentence "**Application streaming** *is a relatively new form of* **software distribute method** using application virtualization". The relation candidate (application

Table 1. Patterns are predefined for each relation type

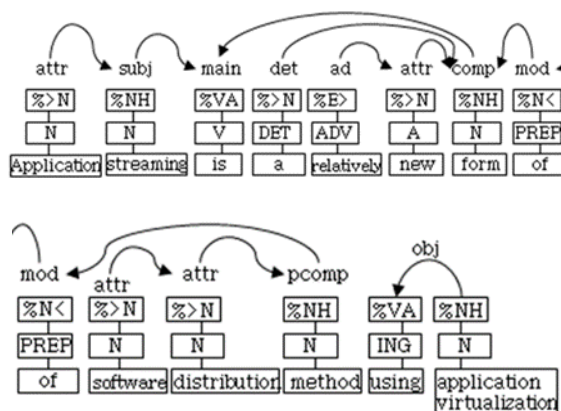| Relation	Pattern	Relation	Pattern
isa	be	provides	provide
	be a form of		offer
	such as		invest
produces	produce	usedFor	be use for
	invent		be use as
	establish		be available for

Fig. 2. An example of Connexor parsing result

streaming, software distribute method) is extracted with an *isa* pattern "be a * of". The parser adopted here is Connexor parser [19].

– **Word features**: the most basic features the relation candidate has. It includes the string which match the pattern (PAT_be_a_relatively_form_of), the main word of the pattern (PAT_be), the domain and range entities of the relation candidate (DOM_application_streaming, RAN_software_distribution_method), the headwords of the entities (WH1_streaming, WH2_method), and the words of the two entities (WM1_application, WM1_streaming, WM2_software, WM2_distribution, WM2_method).
– **Context features** in word level: the words after the domain entity (WA#) and before the range
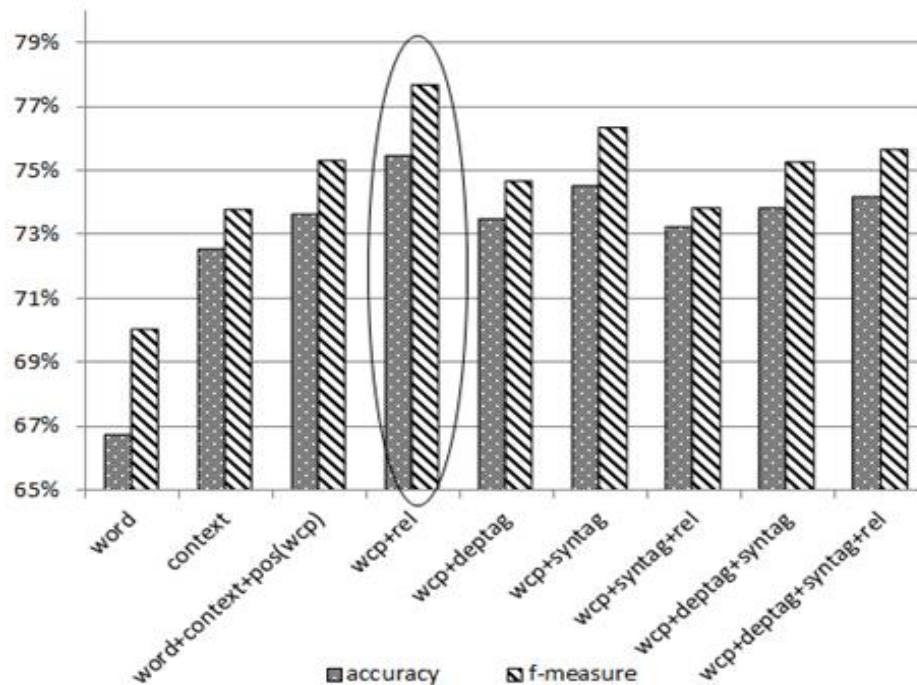
Fig. 3. Experiment results in feature selection

entity (WB#) in the parse tree. # can be 1 or 2, means the position of the words in the context: 1 is right before or after the entity, 2 is the other one (WB1_of, WB2_form, WA1_be, WA2_a). It is also a word level feature.

– **POS features**: POS tag of all above word level features (PM1_N, PM1_N; PM2_N, PM2_N, PM2_N; PB1_PREP, PB1_N; PA1_V, PA2_DET).

– **Syntactic features**: syntactic tags of all above word level features (TM1_>N, TM1_NH; TM2_>N, TM2_>N, TM2_NH; TB1_N<, TB2_NH; TA1_VA, TA2_>N).

– **Syntactic dependency features**: syntactic dependencies from Connexor parser show functional relations between words and phrases in sentences. (RM1_attr, RM1_subj; RM2_attr, RM2_attr, RM2_pcomp; RB1_mod, RB2_comp, RA1_main, RA2_det).

– **Relatedness features**: the probabilistic relatedness information between the pattern and the relation type (PATProb:0.7), the probabilistic and semantic relatedness information between the main word of the pattern and the relation type (PATMainProb:0.5, PATSim:1).

Probabilistic relatedness information is acquired from labeled data, by calculating the percentage of positive cases of the patterns (or main words of the patterns) in the relation type. Actually it is the accuracy of the patterns shown in pattern matching procedure. For example, the pattern "be a form of" has 71.87% of accuracy (PATProb:0.7), and the patterns which have "be" as their main words have accuracy 53.02% in average (PATMainProb:0.5).

The semantic relatedness between the main word "be" and the relation type *isa* is 1 (PATSim:1), which is acquired from WordNet. For certain relation type, collect the main words of its patterns $\{w_1,\ldots w_i,\ldots w_n\}$, for example, {use, employ, available} for relation type *usedFor*, the semantic

relatedness between the main word w_i and the relation type $sim(w_i)$ is related to how many semantically similar words employed for the relation type. The more similar words of w_i employed in the patterns for the relation type, the higher relatedness score w_i gains.

$$score(w_i) = \sum_{j=1}^{n} 1/dis(w_i, w_j) \qquad (1)$$

$$sim(w_i) = \{\max_{j=1}^{n} score(w_j)\}/score(w_i) \qquad (2)$$

In Eq. 1, $dis(w_i, w_j)$ indicates the distance of w_i and w_j in WordNet: the distance of the words in the same synset is 1, the one of direct hyponym and hypernym is 2, and it is infinity if there is no path between two words in WordNet. To a given example {use, employ, available} for relation type *usedFor*, *score*(use) and *score*(employ) are both 2, while *score*(available) is 1, because *dis*(use, employ)=1 (these two words are in the same synset in WordNet), and *dis*(available, available)=1 too. According to Eq. 2, the final semantic relatedness *sim*(use) is 1, while *sim*(available) is 0.5.

5 Experimental Results

5.1 Evaluation on Feature Selection and Performances

Wikipedia pages in IT domain are downloaded for the experiments. The relation candidates are extracted from the first sections of the pages, which normally are definitions and core descriptions, by matching predefined patterns on parsed texts. Connexor parser [19] is used for parsing.

We tried to evaluate the features with *isa* relation classification first. 89 patterns which defined by human annotators are adopted, and 217,383 *isa* relation triples (relation candidates) are extracted from 63,225 pages. For relation classification evaluation, 36,527 triples from 11,128 pages among above data are randomly

selected as *isa* relation type data set, all of them are manually annotated. Among them, again, 1,158 triples from 370 pages are used for test set, and the left 35,389 triples from 10,758 pages are used as training data (First row in Table 2). The percentages of positive cases show how many of the candidates are really hold the relation type - it is the accuracy of pattern matching module indeed, and can be considered as baseline of the relation classification system.

Table 2. Data set

Relation type	Pattern number	Training set (positive cases)	Test set (positive cases)
isa	89	35,389 (54.7%)	1,158 (50.2%)
usedFor	22	720 (43.2%)	126 (42.9%)
produces	46	1,038 (51.4%)	155 (38.1%)
provides	17	1,803 (48.2%)	317 (47.3%)

An existing MEM toolkit MEXENT[2] is adopted for MEM based classifier. Figure. 3 shows the evaluation results of feature selection experiments for relation type *isa*. Both accuracy and f-measure are evaluated, in which f-measure is calculated based on the precision and recall. From the figure, we can see that the contribution of the relatedness feature is comparable with and even outperforms the one of dependency (deptag) and syntactic (syntag) features. The best performance is reported with the feature set "word, context, POS, relatedness" features (wcp+rel). The result using both relatedness and syntactic features (wcp+syntag+rel) is lower than using only one of them (wcp+syntag, wcp+rel), the case is also the same when we compare the result of using both dependency and syntactic features (wcp+deptag+syntag) with using only syntactic feature (wcp+syntag). The reason is seems that, over using of features cause redundancy of the feature, and low down the performance as the result.

Experiments on other three relation types (2nd ~ 4th rows in Table 2) are also performed. The feature sets "word, context, POS, relatedness (wcp+rel) ", which produced the best performance in Figure. 3,

---

[2] https://github.com/lzhang10/maxent

were employed for the experiments of other three relation types (Table 3).

The experiments in Table 3 show that, the performance of MEM model with F-measure was comparable (for "*provides*" relation type) or outperform (for other three relation types) the one of Bayesian classifier which is provided by WEKA[3] [16]. So we decided to adopt MEXENT in the following usefulness evaluation, because it produces confidence score for each prediction, which is required to detect if the prediction is reliable.

5.2 Usefulness Evaluation - Detect Reliable Relations

As we mentioned in the introduction of this paper, the relation classification in this paper aims at building IT domain ontology. The problem is that, either with or without automatic relation classification, the verification of human annotators is still required for all relation candidates before adding them to ontology, as well as the accuracy of the automatic classification is not perfect. To solve this problem, we assume that, if the classification accuracy is comparable with the consistency between two human annotators, then the results can be accepted by default without human verification. We can also assume that, even the accuracy of whole data is lower than the human consistency, there might be still part of the results have comparable or even better accuracy than human consistency.

To verify the assumptions and find a way to save human and time costs, the confidence score provided by the relation classifier is adopted. An evaluation on human consistency is performed first to compare the automatic prediction performance with the human ones (Table 4). Human consistency here means the agreement between two human annotators A and B, or A and C in the verification of relation candidates, while machine accuracy means the agreement between the classifier and human annotator A.

The test set for isa relation type in Table 2 is provided for both human annotators and automatic relation classifier. In the evaluation on human consistency, each annotator verifies examples

Table 3. Performance with MEM

Relation type	Accuracy	Precision	Recall	F-measure
isa	75.5%	71.53%	85.05%	77.71%
usedFor	61.1%	63.86%	73.62%	68.39%
produces	79.4%	86.36%	79.17%	82.61%
provides	62.2%	64.97%	61.08%	62.96%

Table 4. Consistancy between human annotators

Test Set ID	Triple #	Involved annotators	Common Results	Consistency
Test Set 1	458	A, B	380	82.97%
Test Set 2	1000	A, C	788	78.80%

Table 5. Data set for practical environment

Data set name	Triple number (positive cases)
Training set	35,389 (54.7%)
Network Standard	4,659 (52.2%)
Network Architecture	2,612 (68.45%)

independently without knowing other's verification results on the same examples. Given 1,158 *isa* relation candidates, two skilled annotators A and B verified the same 458 examples of them, and showed 82.97% agreement with each other. Then annotator C annotator C who joined this project for less than one month verified the same 1,000 examples of given candidates, and showed 78.80% of agreement (Table 4). The Cohen's Kappa score [20] in average was 0.58, which is in the "moderate" agreement range.

We take the same 1,158 *isa* relation candidates in Table 3 as the evaluation test set for usefulness evaluation, but sort the classification results with the prediction confidence scores provided by MAXENT classifier, and then evaluate the accuracy in different confidence range, to compare with the simple average consistency between human annotators, which is 80.89%. From Figure. 4, we can see that, the results which have higher prediction confidence scores tend to have higher
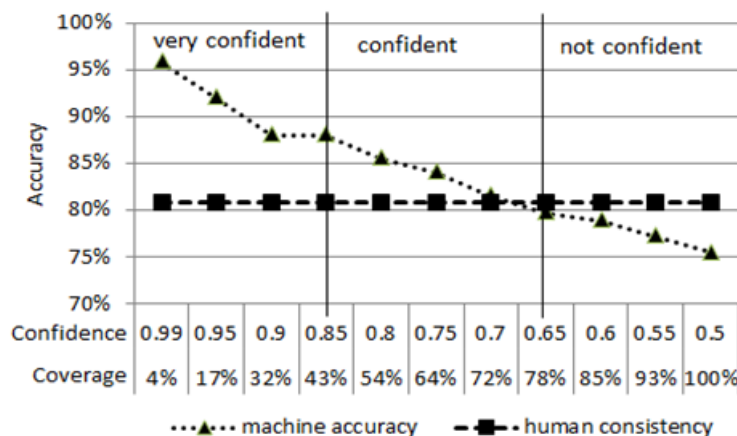

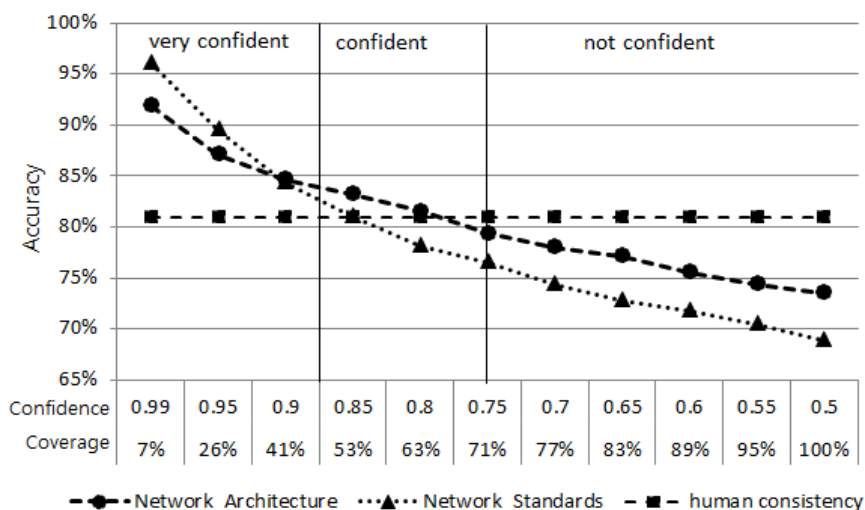**Fig. 4.** The accuracy and coverage according to different confidence score

Fig. 5. The accuracy and coverage according to confidence score
on "Network Architecture" and "Network Standards" test set

accuracy, and about 40% of them which have the highest confidence scores (confidence score>=0.85) show higher accuracy than human consistency, while the accuracy of top 70% of the results (confidence score>=0.65) is comparable with the human ones. It means that, at least 40% of the results which gain higher confidence scores in prediction can be accepted by default without the verification of human annotators, which means, 40% of human and time costs can be saved.

The test set and the training set in Figure. 4 are from the same evaluation set, it means not only

their categories are the same, but also the distribution of the relation triples over the categories is similar in training and test set.

To simulate the real practical environment to verify our assumption again, we suppose the human annotators firstly labeled some *isa* relations from different categories to build training set (which is the same with above experiment in Figure. 4), and the machine classifier needs to classify the extracted relation candidates from two categories "Network Standards" and "Network Architecture", where the relation candidates are not contained in

the training set (Table 5). In this experiment, the training set still covers all categories of the test set; however, the distribution of the relation triples over the categories would be very different between the test set and the training set.

Assume the human consistency on the test set is the same with the one in training set. The experiment results in Figure. 5 show the similar trend with the one in Figure. 4, which still supports our assumption, which is that the results with higher confidence scores tend to have higher accuracy. Again, about 40% of the classification results with confidence scores higher than 0.87 shows higher accuracy than human consistency. However, the threshold is different from the one in Figure. 4, which was 0.85.

Our experiments (Figure.4 and Figure.5) indicate that, prediction confidence score can be used to detect reliable relations from automatic classification results in practical ontology construction. However, the threshold of the confidence score has to be decided through evaluation.

6 Conclusion

In this paper, a feature-based approach for relation classification is presented. Both probabilistic and semantic relatedness information between patterns and relation types is employed as features, and the experiments showed that the relatedness feature is comparable and even outperforms syntactic and dependency features. The probabilistic relatedness information can be acquired from training data, while the semantic relatedness can be calculated using WordNet or other similar taxonomies.

An approach is proposed to distinguish reliable results from others, so that the reliable relations can be added to ontology without human verification, and so time and human costs can be saved in practice. Confidence score provided by relation classifier is employed in this approach. The evaluation results show that with the relation classification approach proposed in this paper, there are about top 40% of the results with higher confidence scores have high accuracy, and the results are comparable with or out-perform the consistencies between human developers. It

indicates that using automatic relation classifier in this paper with the confidence score it provides, at least 40% of the human and time costs can be saved in practice without losing too much of reliability.

As the future work, we are focusing on how to use unlabeled data in an efficient way for a large scale task – extract relations from web scale texts. In the meanwhile, we are also exploring more relatedness information between the entity terms and the relation types, while this paper only focus on the pattern related relatedness.

References

1. **Bach, N. & Sameer, B. (2007).** *A Review of Relation Extraction, Literature review for Language and Statistics.*

2. **Girju, R., Badulescu, A., & Moldovan, D. (2003).** Learning semantic constraints for the automatic discovery of part-whole relations. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology,* pp.1–8. DOI: 10.3115/1073445.1073456.

3. **Girju, R., Badulescu, A., & Moldovan, D. (2006).** Automatic Discovery of Part-Whole Relations. *Computational Linguistics,* Vol. 32, No. 1. DOI: 10.1162/coli.2006.32.1.83.

4. **Kambhatla, N. (2004).** Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations. *Proceedings of 42nd Annual Meeting of the Association for. Computational Linguistics.* DOI: 10.3115/1219044.1219066.

5. **Zhou, G., Su, J., Zhang, J., & Zhang, M. (2005).** Exploring Various Knowledge in Relation Extraction. *Proceedings of the. 43rd Annual*

Meeting of the Association for. Computational Linguistics, pp. 427–434. DOI: 10.3115/ 1219840.1219893.

6. **Bunescu, R.C. & Mooney, R.J. (2007).** Learning to Extract Relations from the Web using Minimal Supervision. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL),* pp. 576–583.

7. **Zhang, Z. (2008).** Mining relational data from text: From strictly supervised to weakly supervised learning. *Inf. Syst.,* Vol. 33, No. 3, pp. 300–314. DOI: 10.1016/j.is.2007.10.002.

8. **Liu, H. & Singh, P. (2004).** ConceptNet: A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal,* Vol. 22, No. 4, pp. 211–226. DOI: 10.1023/B:BTTJ.0000047600.45421.6d.

9. **Rosario, B. & Hearst, M.A. (2005).** Multi-way Relation Classification: Application to Protein-Protein Interactions. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP),* pp. 732–739. DOI: 10.3115/1220575.1220667.

10. **Chang, D.S. & Choi, K.S. (2006).** Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities. *Information Processing & Management,* Vol. 42, No. 3, pp. 662–678. DOI: 10.1016/j.ipm.2005.04.004.

11. **Huang, J.X., Shin, J.A., & Choi, K.S. (2007).** Enriching Core Ontology with Domain Thesaurus through Concept and Relation Classification. *Proceedings of the OntoLex 2007 Workshop associated with ISWC2007.*

12. **Zhang, M., Zhang, J., & Su, J. (2006).** Exploring Syntactic Features for Relation Extraction using a Convolution Tree Kernel. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL,* pp. 288–295. DOI: 10.3115/1220835.1220872.

13. **Zhou, G. & Zhang, M. (2007).** Extracting relation information from text documents by exploring various types of knowledge. *Inf. Process. Manage.* Vol. 43, No. 4, pp. 969–982. DOI: 10.1016/j.ipm. 2006.09.012

14. **Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011).** Natural language processing (almost) from scratch. *The Journal of Machine Learning Research,* Vol. 12, pp. 2493–2537.

15. **Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014).** Relation classification via convolutional deep neural network. *Proceedings of COLING,* pp. 2335–2344.

16. **Huang, J.X., Choi, K.S., Kim, C.H., & Kim, Y.K. (2010).** Feature-based Relation Classification Using Quantified Relatedness Information. *ETRI Journal,* Vol. 32, No. 3. DOI: 10.4218/etrij.10.0209. 0353.

17. **Miller, G.A. (1990).** WordNet: An online lexical database. *International Journal of Lexicography,* Vol. 3, No. 4, pp. 235–312.

18. **Chen, J., Ji, D., Tan, C.L., & Niu, Z. (2006).** Relation Extraction Using Label Propagation Based Semi-Supervised Learning. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics,* pp. 129–136. DOI: 10.3115/1220175.1220192.

19. **Connexor (2008).** *The Connexor Language Parsers and Taggers for English Website.* http://www.connexor.eu/

20. **Cohen, J. (1960).** A coefficient of agreement for nominal scales. *Educational and Psychological Measurement,* Vol. 20, pp. 37–46.

Jin-Xia Huang received her M.S. degree in Computer Science from KAIST in 2000. She is now a senior researcher of Natural Language Processing Research Section, ETRI.

Kyung Soon Lee received her Master and PhD degrees in Computer Science from KAIST in 1997 and 2001, respectively. She is now a full professor of Division of Computer Science and Engineering, Chonbuk National University.

Key-Sun Choi received his Master and PhD degrees in Computer Science from KAIST in 1980 and 1986, respectively. He is now a full professor of Computer Science Department, KAIST since joining in 1988.

Young-Kil Kim received his Master and PhD degrees in Computer Science from Hanyang University in 1995 and 1997, respectively. He is now a Principal Researcher and Director of Natural Language Processing Research Section, ETRI.