A Statistical Approach for Universal Networking Language-Based Relation Extraction

Dat P. T. Nguyen and Mitsuru Ishizuka

***Abstract*—In the effort to enrich available information with machine-processable semantics, Universal Networking Language (UNL) was defined as an artificial intelligent language that is able to represent information and knowledge described in natural languages. One of the main components of UNL is a set of binary relations that represents semantic relationships between concepts in sentences. To provide machine-processable semantics for computers, extraction of such semantic relationships from natural language text is a must. In this paper, we present a method to solve the problem of UNL semantic relation extraction in English sentences. With the assumption that the positions of phrases in a sentence between which there exists a relation have been identified, we focus on the problem of classifying the relation between the given phrases. The UNL relation classifier was developed by using statistical techniques applied on several lexical and syntactic features. In addition to the common used features, we also propose a new feature that reflects the actual semantic relation of two phrases independent on words in the between. Using our new feature in this problem gives the preliminary results that have shown the promising advantages of the feature in some other semantic relation recognition tasks. The evaluation on dataset supplied by UNDL organization shows that our system obtained the result at about 79% accuracy.**

***Index Terms* — Semantic Relation Classification, Semantic Relation Extraction, Universal Networking Language.**

I. Introduction

The Internet has emerged as the most powerful networking infrastructure for communication. By using the Internet, people all over the world can exchange information to each other at anytime and anywhere. However, there is still a language barrier which prevents people in different countries from communicating by their own language [5].

Again, the exponential growth of the Internet has made its content increasingly difficult to find, access, present and maintain for a wide variety of users [10]. In addition, the current Internet's content was designed for humans to read, not for computer to manipulate meaningfully [12]. It means, the available functions of computer are limited in just retrieving information and displaying to people without processing the semantics. Therefore, computers give little support for people in finding, accessing and maintaining the Internet's content.

To deal with the above problems, Universal Networking Language (UNL) was defined as an artificial language that is able to represent information and knowledge described in natural languages [13]. Thus, it enables computers to process information in form of knowledge across the language barrier [6].

Let us introduce some main components of UNL which includes:

- **Universal Words (UWs):** is UNL's vocabulary. However, UWs are different from words of natural languages in that each UW represents only one concept that may be simple or compound [13]. For example, the Universal Word *"state(icl>do(obj>thing))"* denotes the action of presenting something while the word *"state(icl>situation)"* denotes the meaning of a situation. As we can see, while a word in English or other natural languages may have several meanings, each UW indicates only one meaning. In other words, an entry of UWs corresponds to only one word sense or concept.
- **Relations:** defines relationships between pairs of UWs. UNL uses directed binary relations to describe objectivity information of sentences [13]. The types of relationships are differentiated by labels. For example, the relations in *Figure 1* indicate that the agent initiating the action "adopt" is "the conference", the time when the action "adopt" happened is "the year 1964", etc...Please refer to the Appendix section for the full list of 45 UNL relations.
- **Attributes:** describes the subjectivity of sentences including time with respect to the speaker *(past, prensent, future)*, speaker's view of aspect *(begin, continue, complete...)*, speaker's view of reference *(specific, non-specific...)*, speaker's focus *(emphasis, theme, title...)*, speaker's attitudes *(confirmation, exclamation...)* and speaker's view point *(ability, admire...)* [13].

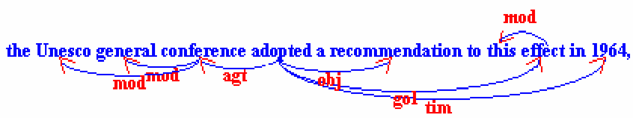

Fig. 1. UNL's relations in the fragment *"the Unesco general conference adopted a recommendation to this effect in 1964"*

Dat P. T. Nguyen is with the Faculty of Information Technology, University of Natural Sciences, HoChiMinh, Vietnam. He is currently a graduate student at Graduate School of Information Science and Technology, The University of Tokyo, 113-8656 Tokyo, Japan (phone: +81-80-5682-1505; e-mail: nptdat@miv.t.u-tokyo.ac.jp)

Mitsuru Ishizuka is with Graduate School of Information Science and Technology, The University of Tokyo, 113-8656 Tokyo, Japan (phone: +81-3-5841-6347; e-mail: ishizuka@miv.t.u-tokyo.ac.jp)

Using the above components, UNL can represent the information and knowledge for sentence by sentence [5]. The representations can be visualized by semantic networks in which nodes indicate UWs and arcs indicate relations as shown in *Figure 2*. Differently from natural languages, UNL representations are unambiguous [6].

Obviously, one of the important tasks to create UNL representations from natural language text is to extract relations between pairs of UWs. Relation extraction task can be divided into two subtasks: that of identifying pairs of UWs between which it is likely to have a relation and that of identifying the relation label for the pairs. In this paper, we focus on the second problem. That means, we develop a label classifier given pairs of UWs between which there exists a relation. Although UNL can represent information and knowledge described by any kind of natural languages, the classifier we present in this paper processes for English text. With the assumption that we have

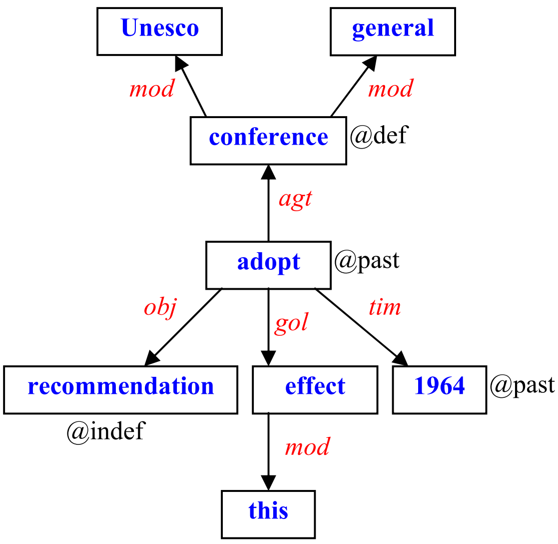

Fig. 2. UNL's semantic network for the fragment *"the Unesco general conference adopted a recommendation to this effect in 1964"*

already mapped the words in the English text with the corresponding UWs, the problem now can be stated: given an English text and a pair of phrases in the text between which there exists a relation, identify the type of relation for the pair.

In this work, we apply statistical techniques to train our classifier. First, a number of features of the two phrases will be extracted from the text to create some feature vectors. Along with some well-known features which were usually used in Semantic Role Labeling task such as Phrase Type, Head Word, Voice, Dependency Path and Normal Path, we propose a new feature that gives an improvement for the classifier. Then, we count the occurrences of the relations for each feature vector. Finally, we estimate the probability of each relation type given a feature vector. We report our classifier's performance trained on the dataset provided by Universal Networking Digital Language foundation (UNDL).

In the next section, we will review some related works. In section 3, we describe the methodology used to deal with the problem. We then delineate the experiment, report the performance of our system and provide some discussions in section 4. Finally, we conclude and present the future works.

II. Related Works

The problem of recognizing semantic relation has emerged in some tasks such as Semantic Role Labeling [1] [11] [8], and Relation Detection and Characterization (RDC) [9]. Let us examine the similarity between our problem and these tasks.

Semantic Role Labeling is a process of assigning semantic role for a specific target word in a sentence. For example, the target word "broke" has "Anglers" as its agent, "the ice" as its Whole_patient, and "to fish" as its Purpose in the following sentence:

[$_{Agent}$ Anglers] ***broke*** *[$_{Whole\_patient}$ the ice] [$_{Purpose}$ to fish] and diesel bubbled up to the surface.*

The semantic relations between a target word and its semantic roles are defined in its associated frame semantic. The above target word "broke" is a lexical unit associated with the frame semantic called Cause_to_fragment. FrameNet project [4] defines a set of frame semantics along with associated lexical units for each frame. It also provides a lexical resource which is fully hand-annotated with semantic role. Based on such resource, there are some works aiming to build automatic semantic role labeling systems using statistical methods or machine learning. The earliest state-of-the-art labeling system was proposed by D. Gildea and D. Jurafsky in [1]. In this work, the system was trained by deriving syntactic features from syntactic parse tree and applying statistical methods on FrameNet lexical resource. Some other works applying machine learning methods on a variety of syntactic features derived from different parsers as in [11] & [8].

The RDC task was defined in the Automatic Content Extraction program (ACE) whose goal is to develop technologies to support automatic processing of human language in text form. The ACE program defined a list of entity types along with possible relations between the entities. The RDC task requires certain relations between entities to be detected from natural language text. For example, a good RDC system should successfully identify "chairman" as *individual person*, "board" as *group person* and *management relation* between them in the fragment *"...been the chairman of its board"*. In [9], N. Kambhatla dealed with the RDC task by employing Maximum Entropy models to combine lexical, syntactic and semantic features derived from the text.

III. Method

We trained the classfier by using statistical techniques based on the features extracted from natural language text. Because UNL relations are directed binary direction, we can certainly call the phrases in a relation source phrase and destination phrase. For each pair of source and destination phrases in a relation of the training set, we extracted linguistic features and created a feature vector. The number of occurrances of each relation label for each feature vector will be counted in the whole training set. In testing time, the probability for each relation label will be estimated given the feature vector associated with a pair of phrases. Then, the relation label with highest probability will be chosen and assigned to the phrases.

A. Feature selection

The features used in the classifier are described in the followings:

1) Phrase Type

The Part-Of-Speech tags of the two phrases in the relation derived from the syntactic parse tree of the sentence produced by Charniak's parser [9]. Some example values of this feature can be: NN (noun), JJ (adjective), VBN (verb).

2) Head word

The head words of the two phrases in the relation. To reduce the variety of values, the actual value should be the root of the head word provided by the syntactic parse tree. We hope that head word feature will reflect the semantic of the words. For example, in the phrase *"considering that the Unesco general conference adopted a recommendation to this effect in 1964"*, the relation between "conference" and "adopt" is likely to be "agt" relation while the relation between "adopt" and "recommendation" is likely to be "obj".

3) Voice

The value that indicates a verb is active or passive in case a phrase is verb. The feature is important for "agt" and "obj" relations because the positions of the subject and objects with respect to the verb in an active sentence are reversed in its passive form. Note that the UNL's "agt", "obj" relations indicate the relationships between the subject, object and there verb respectively.

4) Dependency Path

The string that indicates a path from the source phrase through the dependency tree to the destination phrase. Dependency path is constructed by collecting the Part-Of-Speech tags in the nodes, the relations on the edges and some movement-indicator characters ('^' indicates upward movement and 'v' indicates downward movement). We expect that this feature will map the grammatical functions between words with UNL relations. For example, the relation "agt" is corresponding to the "subject" function in dependency tree. Both of the noun phrases that have "agt" relation and "subject" function with respect to the verb in an active sentence indicate the entities which activate the action represented by that verb. We extract dependency path from the dependency tree derived from Minipar parser [2]. Dependency tree indicates dependency relationships between a head and a modifier created from constituent parse. *Figure 5* gives an example of dependency tree.

5) Syntactic Cross Path

The string representing a path from the source position through the syntactic parse tree to the destination position. This feature can be extracted by executing the following steps:

- For source and destination phrases, specify the highest nodes which still receive the head words from the phrases. We call them H1, H2 respectively.
- Specify the lowest common node that covers both of the phrases. We call the node C.
- Assume that the source phrase and the destination phrase are seperated, we have three cases:
  - The node C is higher than the nodes H1 and H2: trace upwards from H1 to C, then trace downwards to H2
  - The node H1 is higher or equals to the node C: trace downwards from C to H2.
  - The node H2 is higher or equals to the node C: trace upwards from H1 to C.

Note that when we trace through the tree, besides Part-Of-Speech tags, syntactic cross path also includes their head words if they are close-class words. It also includes movement-indicator characters.

Compare to the normal path feature proposed in [1], syntactic cross path feature has an advantage that it reflects the syntactic structure of the two concepts represented by the two phrases that is independent on the words lying between the phrases. For examples, in the *Table 1*, syntactic cross path feature gives the same value for the fragments which have different set of words between the two phrases "adopted" and "effect" because the semantic relations are fix. The syntactic parse trees are illustrated in *Figure 3 & 4*.

Let us give some examples of the above features. Assume we identify the label for a relation from "adopted" to "effect" in the following fragment:

Adopted *a recommendation to this* ***effect*** *in 1964*

Phrase type	:*VBD* (POS tag of "adopted"), *NN* (POS tag of "effect")
Head word	:*adopt, effect*
Voice	:*Active, Unspecified*
Dependency path	:*(V)obj v (N)mod v (Prep)pcomp-n v(N)*
Syntactic cross path	:*VP v NP v PP(to) v NP*

B. Probability prediction

As mentioned above, we will count number of relations for each feature vector in the training data and estimate the probability for each relation given a feature vector in testing. The conditional probability can be estimated as the following:

$$P(R|F) = \frac{\#(R,F)}{\#(F)}$$

where:

- *R:* relation label.
- *F:* feature vector.
- *#(R, F):* number of relations with label R counted for feature vector F in the training.
- *#(F):* number of relations which receive F as a feature vector counted in the training.

Sentence	Syntactic cross path feature	Normal path feature
"**adopted** a recommendation to this **effect** in 1964"	VP v NP v PP(to) v NP	VBD ^ VP v NP v PP v NP v NN
"**adopted** a recommendation to a highly beneficial **effect** in 1964"	VP v NP v PP(to) v NP	VBD ^ VP v NP v PP v NP v NP v NN

Table 1. Syntactic cross path feature gives same value for different fragment while normal path feature does not.

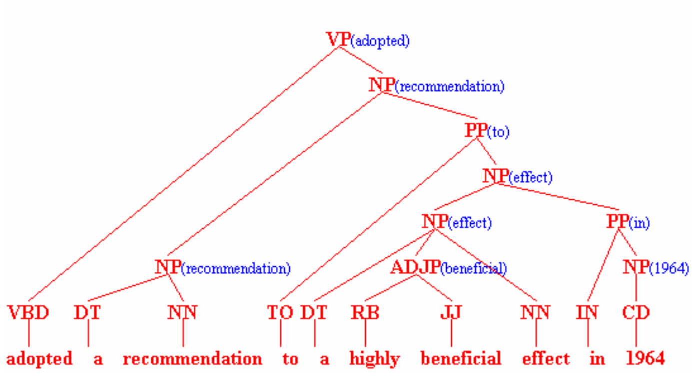

Fig. 3. Syntactic parse tree of the fragment *"adopted a recommendation to a highly beneficial effect in 1964"*

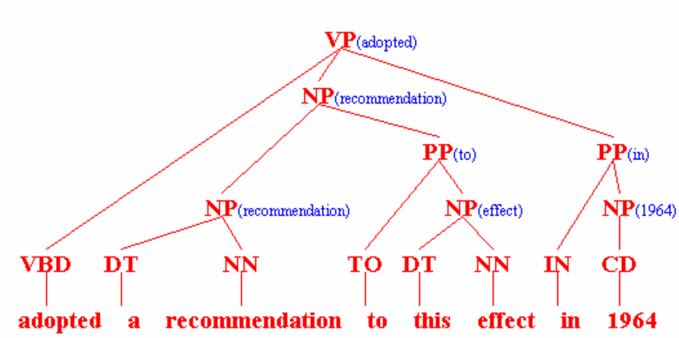

Fig. 4. Syntactic parse tree of the fragment *"adopted a recommendation to this effect in 1964"*

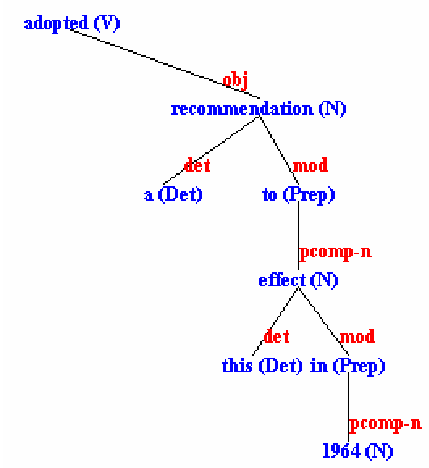

Fig. 5. Dependency tree of the fragment *"adopted a recommendation to this effect in 1964"*

Generally speaking, a feature vector F can be created from all the features mentioned above. However, the annotation data we have contains only 434 fragments in which there are 4764 relations distributing on 25 relation types. The average number of relations per relation type is 190 which is relatively small compare with the number of possible values of a feature vector. This leads to many cases that a value of feature vector in testing data is absent in training data and the system cannot predict the relation labels for such cases. Therefore, the training data is sparse to do statistics. *Table 3* shows the poor result of such system.

To estimate the probability for such the cases, we reduce the restriction of the condition in the probability formula by dividing the general feature vector into smaller vectors. In other words, we reduce the dimension of feature vector. We estimate the conditional probability by using linear interpolation method as proposed in [1]:

$$P(R|\text{the two phrases}) = \lambda_1 * P_1(R\,|\,pt_1) + \lambda_2 * P_2(R\,|\,pt_2) + \lambda_3 * P_3(R\,|\,hw_1) + \lambda_4 * P_4(R\,|\,hw_2) + \lambda_5 * P_5(R\,|\,voice_1, voice_2) + \lambda_6 * P_6(R\,|\,depPath) + \lambda_7 * P_7(R\,|\,synCrossPath) + \lambda_8 * P_8(R\,|\,pt_1, pt_2, hw_1, hw_2, voice_1, voice_2, depPath, synCrossPath)$$

where:

- $\sum_{i=1}^{i=8} \lambda_i = 1$
- *pt_1, pt_2, hw_1, hw_2, $voice_1$, $voice_2$* : **phrase type**, **head word** and **voice** feature of phrase1, phrase2 respectively.
- *depPath, synCrossPath* : **dependency path** and **syntactic cross path** feature between phrase1 and phrase2

The algorithms for training and testing can be summarized as:

Training:

- *Input* : a training set including full information of relations
- *Output* : estimated conditional probabilities P_1,..., P_8

<u>Step1:</u> For each training fragment S; source phrase and destination phrase; relation label R, do:

Step1a: Receive syntactic parse tree T from Charniak parser and dependency tree D from Minipar parser for the fragment S.

Step1b: Based on the trees T and D, extract the features for the two phrases: pt_1, pt_2, hw_1, hw_2, $voice_1$, $voice_2$, depPath, synCrossPath

Step1c: Increase the following counters by 1:

- *$c_1(R, pt_1)$, $c_2(R, pt_2)$,..., $c_8(R, pt_1, pt_2, hw_1, hw_2, voice_1, voice_2, depPath, synCrossPath)$*
- *$t_1(pt_1)$, $t_2(pt_2)$,..., $t_8(pt_1, pt_2, hw_1, hw_2, voice_1, voice_2, depPath, synCrossPath)$*

where $c_i(R, F_i) = \#(R, F_i)$ and $t_i(F_i) = \#(F_i)$.

Step2: Calculate all the conditional probabilities:

$$P_1(R\,|\,pt_1) = \frac{c_1(R, pt_1)}{t_1(pt_1)}, ...,$$

$$P_8(R\,|\,pt_1, pt_2, hw_1, hw_2, voice_1, voice_2, depPath, synCrossPath) = \frac{c_8(R, pt_1, pt_2, hw_1, hw_2, voice_1, voice_2, depPath, synCrossPath)}{t_8(pt_1, pt_2, hw_1, hw_2, voice_1, voice_2, depPath, synCrossPath)}$$

Testing:

- *Input* : a fragment S; source phrase and destination phrase; all the conditional probabilities P_1, ..., P_8
- *Output* : relation label R

Step1: Receive syntactic parse tree T from Charniak parser and dependency tree D from Minipar parser for the fragment S.
Step2: Based on the trees T and D, extract the features for the two phrases: pt_1, pt_2, hw_1, hw_2, $voice_1$, $voice_2$, depPath, synCrossPath.
Step3: Estimate probabilities for all relation labels by using linear interpolation as listed above.
Step4: Choose the relation label R for the two phrases:

$$R = \arg\max_r P(r\,|\,the\,two\;phrases)$$

IV. Experiment

The dataset used to develop the system was supplied by UNDL Foundation. It contains 434 text fragments from UNESCO's documents annotated with 4764 UNL relations including 25 relation types. A relation includes the boundaries of source and destination phrases along with the relation name. For experiment, we randomly divided the dataset into training set and testing set. *Table 2* shows more details about training set and testing set.

	Text fragments	Relations	Average of number of Relation / Fragment
Training set	384	4189	10.91
Testing set	50	575	11.50
Total	434	4764	10.98

Table 2. Training set and testing set used for the experiment

Because we solve the problem of classifying the relation type given correct two phrases, the number of relation returned by the classifier equals to the truth number of relations existing in testing data. The accuracy of the system may be calculated by dividing the number of relation that correcly classified by the total number of relations.

At first, we tried to build the system by using only one common feature vector created from all features. As mentioned above, testing on sparse data gives a poor result shown in *Table 3*. It is interesting to note some details about testing process of this system. In case of using our syntactic cross feature, there are 475 cases of common feature vectors in testing data are absent in training data. Since we have 575 as the total number of relations in testing data, the system can predict for only 100 relations in which it gives 93 successful cases. This means, the system will give a promising result of about 93% accuracy if we have enough annotation data. However, the system that replace syntactic cross feature by normal path feature gives the estimated result at 73.43% of accuracy in ideal annotation.

	(1)	(2)
Number of accurately predicted relations	93	47
Number of predicted relations	100	64
Number of absent feature values	475	511
Number of relations in testing data	575	575
Accuracy	16.17%	8.17%
Estimated accuracy if enough training data is used	93%	73.43%

Table 3: The accuracy of the systems that estimate probabilities based on only one feature vector containing all the extracted features
(1) Extracted feature: pt1, pt2, hw1, hw2, voice1, voice2, depPath, synCrossPath
(2) Extracted feature: pt1, pt2, hw1, hw2, voice1, voice2, depPath, normalPath

We developed some systems using different set of features to compare the effectiveness of dependency path, syntactic cross path and normal path features. We set the equal values for λ_1,.., λ_8 to assure that all features give the same affect on the final result. *Table 4* reports the result of such systems. Note that all the systems use all the basic features such as phrase type, head word and voice along with some optional features such as dependency path, syntactic cross path and normal path. The system that uses dependency path, along with syntactic cross path yields 73.20% while replacing the syntactic cross path feature by the normal path feature yields only 70.26%. Dismissing the dependency path feature, the accuracy of the third system which uses syntactic cross path features gives higher accuracy than that of the fourth system, which uses normal path feature. We can easily realize that replacing the normal path feature by syntactic cross feature improves the accuracy by about 3%. Comparing the systems which use only one optional feature, we can see that the system useing syntactic cross path feature gives the highest result. Therefore, we can conclude that the syntactic cross path feature is better than the normal path feature and dependency path feature in this problem.

Our strategy to choose λ values is to give the higher value to the more important feature. The best result we could find is 78.96% accuracy given by the system using all the basic features, along with dependency path, syntactic cross path features and the set (0.025, 0.025, 0.075, 0.075, 0.30, 0.30, 0.075, 0.125) as values for λ. However, we cannot assure this set of λ values give the best in the whole. Furthermore, the combination of features is dependent on the linear interpolation fomula we are using. All of these problems may keep the system from gaining the actual best result based on the above features. Thus, we intend to deal with these problems by applying machine learning mehods.

Table 5 shows the more details about precision, recall and F-value for each type of relation in testing data. This result was produced by the best system mentioned above. Note that the classifier gives better results for the relation types which appear more frequently in training data. As a result, we hope the performance will be better if we have more annotations for training data.

No	Features	Accuracy (%)
1	pt1, pt2, hw1, hw2, voice1, voice2, depPath, synCrossPath	73.20
2	pt1, pt2, hw1, hw2, voice1, voice2, depPath, normal path	70.26
3	pt1, pt2, hw1, hw2, voice1, voice2, synCrossPath	68.52
4	pt1, pt2, hw1, hw2, voice1, voice2, depPath	67.65
5	pt1, pt2, hw1, hw2, voice1, voice2, normalPath	65.56
6	pt1, pt2, hw1, hw2, voice1, voice2	63.13

Table 4. The accuracy of the systems using different set of features. Equal values of λ are set to make sure that all features give the same affect on the final result

Rel	All data	Testing data					
		(1)	(2)	(3)	P	R	F
agt	**171**	29	24	19	0.65	0.79	0.71
and	**560**	44	42	36	0.81	0.85	0.83
aoj	**907**	102	102	89	0.87	0.87	0.87
bas	**2**						
ben	**4**						
cnt	**11**	4					
con	**17**	3					
dur	**12**	2					
fmt	**5**						
frm	**7**	2					
gol	**101**	15	9	7	0.46	0.77	0.58
ins	**4**						
man	**331**	48	30	29	0.60	0.96	0.74
met	**15**	1	1				
mod	**1253**	138	166	135	0.97	0.81	0.88
obj	**961**	141	176	119	0.84	0.67	0.75
plc	**130**	11	5	3	0.27	0.60	0.37
plf	**1**						
pof	**8**	1					
pur	**168**	25	16	13	0.52	0.81	0.63
qua	**4**	1					
rsn	**8**						
scn	**5**						
src	**17**	2	2	2	1	1	1
tim	**62**	6	2	2	0.33	1	0.50
Total	**4764**	575	575	454	0.7896	0.7896	0.7896

Table 5. Precision, Recall and F-Value for each type of relations in testing data. (1: Actual, 2: System Answer, 3: Correct)

V. Conclusion

We have introduced UNL as a language for expressing knowledge and information that can be described in natural language text. We also presented a statistical approach for classifying UNL semantic relations which have been shown to be successfully applied in Semantic Role Labeling task. In addition, we proposed a new feature called syntactic cross path that was proved to be better than normal path and dependency path features used in [1] and [8]. Most of the path features proposed by previous works reflect all the phrases between the two underlying constituents. Independent of what exists between the two constituents, our syntactic cross path reflects the actual semantic relation between the constituents. The analysis on empirical evaluation shows that our new feature gives a promising result in semantic relation extraction tasks.

We intend to continue this work by applying machine learning methods to find a better way of feature combination in estimating probability for a relation. This work should also be extended to deal with the problem of identifying the boundaries of source and destination phrases of a relation and to map natural language words with corresponding Uws. That means the problem of identifying UNL relations from natural language text will be completely solved.

Appendix

The following table lists all of UNL relations defined in UNL 2005 [13]. In the examples, we underline pairs of phrases between which there exists a relation. However, we do not indicate the directions.

No	Rel	Definition	Example
1	agt	(agent) indicates a thing in focus that initiates an action	<u>John</u> <u>breaks</u>
2	and	(conjunction) indicates a partner to have conjunctive relation to	…<u>singing</u> and <u>dancing</u>
3	aoj	(thing with attribute) indicates a thing that is in s state or has an attribute	<u>Skiiing</u> is <u>nice</u>. <u>I</u> <u>have</u> a pen.
4	bas	(basis) indicates a thing used as the basis (standard) of comparison	John is <u>more</u> quiet than <u>shy</u>.
5	ben	(beneficiary) indicates an indirectly related beneficiary or victim of an event or state	It is <u>good</u> for <u>John</u> to …
6	cag	(co-agent) indicates a thing not in focus that initiates an implicit event that is done in parallel	To <u>walk</u> with <u>John</u>
7	cao	(co-thing with attribute) indicates a thing not in focus that is in a parallel state	<u>be</u> with <u>you</u>
8	cnt	(content) indicates the content of a concept	a <u>language</u> generator “<u>deconverter</u>”...
9	cob	(affected co-thing) indicates a thing that is directly affected by an implicit event done in parallel or an implicit state in parallel	John was <u>injured</u> in the accident with his <u>friends</u>
10	con	(condition) indicates a	If you are <u>tired</u>,

		non-focused event or state that conditions a focused event or state	we will go straight home
11	coo	(effected co-thing) indicates a co-occurrent event or state for a focused event or state	… was crying while running
12	dur	(duration) indicates a period of time during which an event occurs or a state exists	… work nine hours (a day)
13	equ	(effected co-thing) indicates an equivalent concept	the deconverter (a language generator)
14	fmt	(range/from-to) indicates a range between two things	the alphabets from a to z
15	frm	(origin) indicates an initial state of a thing or a thing initially associated with the focused thing	a visitor from Japan
16	gol	(goal/final state) indicates a final state of object or a thing finally associated with the object of an event	the lights changed from green to red
17	icl	(included/a kind of) indicates an upper concept or a more general concept	a bird is a (kind of) animal
18	ins	(instrument) indicates an instrument to carry out an event	look at stars through a telescope
19	int	(intersection) indicates all common instances to have with a partner concept	an intersection of tableware and cookware
20	iof	(an instance of) indicates a class concept that an instance belongs to	Tokyo is a city in Japan
21	man	(manner) indicates a way to carry out an event or the characteristics of a state	move quickly I often visit him.
22	met	(method/means) indicates a means to carry out an event	… solve … with dynamics
23	mod	(modification) indicates a thing that restricts a focused thing	the whole story a master plan
24	nam	(name) indicates a name of a thing	his son "Hikari"
25	obj	(affected thing) indicates a thing in focus that is directly affected by an event or state	the sugar melts into … I have a pen.
26	opl	(affected place) indicates a place in focus affected by an event	… pat … on shoulder
27	or	(disjunction) indicates a partner to have disjunctive relation to	Will you stay or leave?
28	per	(proportion/rate/distribution) indicates a basis or unit of proportion, rate or distribution	eitgh hours a day
29	plc	(place) indicates a place where an event occurs, or a state that is true, or a thing that exists	… cook … in the kitchen
30	plf	(initial place) indicates a place where an event begins or a state that becomes true	traveling from Tokyo
31	plt	(final place) indicates a place where an event ends or a state that becomes false	to travel to Boston
32	pof	(part of) indicate a concept of which a focused thing is a part	the preamble of a document
33	pos	(possessor) indicates the possessor of a thing	John's dog
34	ptn	(partner) indicates an indispensable non-focused initiator of an action	… compete with John
35	pur	(purpose) indicates the purpose or objective of an agent of an event or the purpose of a thing that exists	… come to see you
36	qua	(quantity) indicates the quantity of a thing or unit	Two cups of coffee
37	rsn	(reason) indicates a reason why an event or a state happens	... didn't go because of the rain
38	scn	(scene) indicates a scene where an event occurs, or state is true, or a thing exists	… win a prize in a contest
39	seq	(sequence) indicates a prior event or state of a focused event or state	It was green and then red.
40	src	(source/initial state) indicates the initial state of an object or thing initially associated with the object of an event	The lights changed from green to red.
41	tim	(time) indicates the time an event occurs or a state is true	… leave on Tuesday
42	tmf	(initial time) indicates the time an event starts or a state becomes true	… work from morning to [till] night
43	tmt	(final time) indicates a time an event ends or a state becomes false	… be full till tomorrow
44	to	(destination) indicates a final state of a thing or a final thing (destination) associated with the focused thing	a train for London
45	via	(an intermediate place or state) indicates an intermediate place or state of an event	… bike … through the Alps

Acknowledgment

We are grateful to Dr. T. Yokoi, Dr. H. Uchida, Ms. M. Zhu and other members of ISec for their collaboration; Dr. Nigel Collier for his helpful comments; Dr. Y. Matsuo, Mr. D. Bollegala and other members of Ishizuka laboratory for their warm supports.

References

[1] D. Gildea, D. Jurafsky, *Automatic Labeling of Semantic Roles*, In proceedings of the 38th Annual Meeting of the ACL, pages 512-520, Hong Kong, October 2000.

[2] D. Lin, *Dependency-based evaluation of MINIPAR*, In Workshop on the Evaluation of Parsing Systems, Gradana, Spain, 1998

[3] E. Charniak, *A maximum-entropy-inspired parser*, In proceedings of NAACL, pages 132-139, Seattle, Washington, 2000.

[4] FrameNet project: http://framenet.icsi.berkeley.edu

[5] H. Uchida, *The Universal Networking Language Beyond Machine Translation*, International Symposium on Language in Cyberspace, Seoul, Korea, September 2001.

[6] H. Uchida, M. Zhu, *UNL2005 from Language Infrastructure*

toward Knowledge Infrastructure, special speech at PACLING, August 2005.

[7] Institute of Semantic Computing, *CDL (Concept Description Language): A common Language for Semantic Computing*, iSec Technical Report, May 2005.

[8] K. Hacioglu, *Semantic role labeling using dependency trees*, In proceedings of the 20th International Conference on Computational Linguistic (COLING), Geneva, Switzerland, August 2004

[9] N. Kambhatla, *Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations*, In proceedings of the 42th Annual Meeting of the ACL, Barcelona, Spain, July 2004.

[10] O. Lassila et. al, The Semantic Web and its languages, IEEE Intelligent Systems, pages 67-73, November-December 2000.

[11] S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, D. Jurafsky, *Semantic Role Labeling Using Different Syntactic Views*, In proceedings of the 43rd Annual Meeting of the ACL, page 581-588, Ann Arbor, Michigan, June 2005.

[12] T. Berners-Lee, J. Hendler, O. Lassila, *The Semantic Web*, Scientific American, May 2001.

[13] UNL Center of UNDL Foundation, *Universal Networking Language (UNL) Specifications Version 2005*, Available at http://www.undl.org/unlsys/unl/unl2005/