A Hybrid Approach for Semantic Structure Annotating of Text

Abstract

Facing the challenges of annotating naturally occurring text into semantic structured form for automatically information extracting, current Semantic Role Labeling (SRL) systems have been focusing on semantic predicate-argument structure. Based on the Concept Description Language for Natural Language (CDL.nl) which aims to describe the concept structure of text by a set of pre-defined semantic relations, we develop a parser to add a new layer of semantic annotation of natural language sentences. The parsing task is a relation extraction process with two steps: relation detection and relation classification. We put forward a hybrid approach with different methods for two steps: firstly, based on dependency analysis, a rule-based method is presented to detect all entity pairs between each pair there exits a relationship; secondly, we use a feature-based method to assign CDL.nl relation to each detected entity pair with Support Vector Machine. We report our preliminary results on our manual dataset annotated with CDL.nl relations.

1. Introduction

With the dramatic increase in the amount of textual information available in digital archives and the WWW, there has been growing interest in techniques for automatically extracting information from text. It is expected to identify information from sentences and put them in a structured format to be inquired and utilized in semantic computing applications such as web searching and information extraction[4]. Recently, a lot of attention has been devoted to Semantic Role Labeling (SRL) of natural language text with a layer of semantic annotation of predicate-argument structure, as called shallow semantic parsing, which is becoming an important component in many kinds of NLP applications[14, 7]. SRL is currently a well-defined task with a substantial body of work and comparative evaluation[10, 3]. Within the task of semantic role labeling, high-performance systems have been developed using FrameNet[1] and PropBank[15] corpora as training and testing material.

While Semantic Role Labeling focuses on predicate-argument structure, towards the goal of putting the whole sentence into a semantic structure form, Yokoi et al. (2005)[17] presented a descriptive language named CDL.nl (Concept Description Language for Natural Language) which is part of the realization of spirits of the work "semantic information processing"[11]. To form the semantic structure of natural language sentences in a graph representation, CDL.nl defines a set of semantic relations. They record semantic relationships showing how each meaningful entity (can be nominal, verbal, adjectival, adverbial) semantically relates to another entity. It connects all meaningful entities into a united graph representation, not only predicate-argument related entities.

So, with CDL.nl relation set, the task of structure annotation turns to be a relation extraction process which can be divided into two steps: relation detection–detecting entity pairs with each there exists a meaningful relationship; relation classification–labeling each detected entity pair with a specific relation. For CDL.nl relation extraction, the challenge we are facing is that not only the relation detection step is more difficult than a classification problem as in semantic role labeling, but also classification on a such wide variation of CDL.nl relation types is harder than on only predicate-argument roles. In this paper, we put forward a hybrid approach with two different methods for each step: firstly, based on dependency analysis, a rule-based method is presented for relation detection; secondly, a feature-based method is presented to assign CDL.nl relation to each detected entity pair based on different levels of syntactic analysis.

Our contributions can be summarized as follows:

- We develop a parser to add a new layer of semantic annotation of natural language sentences. By annotating text with deeper and wider semantic structure, it can expand the extent to which shallow semantic information can be used in real semantic computing applications such as Information Extraction and Text Summarization.
- Our study shows an intermediate phrase towards semantic parsing of natural language processing from syntactic processing. It will be useful to various NLP

applications such as machine translation and language understanding.

The rest of this paper is organized as follows. Section 2 shows the background in semantic role labeling domain about semantic roles in FrameNet, PropBank and semantic role labeling tasks. Section 3 introduces CDL.nl relation set and specifies its importance and challenges. Section 4 proposes our hybrid method for relation extraction. Section 5 reports our preliminary experimental results and our observations. We conclude our work in Section 6.

2 Background

During the last few years, corpora with semantic role annotation and automatic annotation systems have received much attention. Three corpora are available for developing and testing predicate-argument annotation–FrameNet[1], PropBank[15] and NomBank[12]. Semantic role labeling is the process of assigning a simple WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW, etc. structure to sentences in text. In this section, we focus on semantic role labeling systems which are based on FrameNet and PropBank.

2.1 FrameNet Semantic Roles

The Berkeley FrameNet project starting from 1998 is a primarily corpus-based lexicon-building project that documents the links between lexical items and the semantic frame(s) they evoke. Its starting point is the observation that words can be grouped into semantic classes, the so-called "frames", a schematic representation of situations involving various participants, props, and other conceptual roles. Each frame has a set of predicates (nouns, verbs or adjectives), which introduce the frame. For each semantic frame, it defines a set of semantic roles called **frame elements** which are shared by all predicates of the frame. The term lexical unit is used for a word in combination with one of its senses.

For example, the frame **Intentionally_create**, shown in Figure 1, is invoked by a set of semantically related predicates such as **verbs** *make* and *found*, **nouns** *creation* and *generation*, and is defined as: *The Creator creates a new entity, the Created_entity, possibly out of Components.* The roles defined for this frame, and shared by all its lexical entries, include **core roles** *Created_entity* and *Creator*, **non-core roles** *Co_participant*, *Components*, and so on.

FrameNet contains example sentences that illustrate all possible syntactic and semantic contexts of the lexical items taken into consideration. Besides the corpus, two other components distinguished in FrameNet are a set of lexical entries and a frame ontology.

Semantic role labeling processing

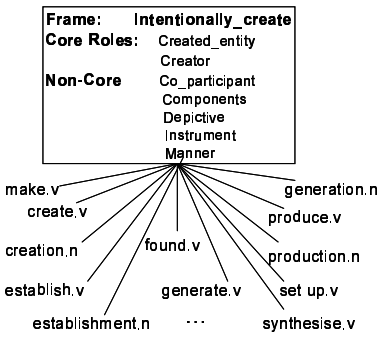

Figure 1. Sample frame from the FrameNet lexicon

Based on FrameNet annotation system, given a crude sentence, the role labeling process goes through (1) identifies all predicates; (2) disambiguates the frame for each predicate; and (3) labels the roles of arguments that relate to the predicate based on the frame definition.

Bill Gates is an American entrepreneur and the [$_{\text{Role}}$ *chairman] of [*$_{\text{Jurisdiction}}$ *Microsoft], [*$_{\text{Created\_entity}}$ *the software company] [*$_{\text{Creator}}$ *he] founded [*$_{\text{Co\_participant}}$ *with Paul Allen] [*$_{\text{Place}}$ *in Albuquerque, New Mexico] [*$_{\text{time}}$ *on April 4, 1975].*

Above is an example showing how to annotate a sentence using FrameNet roles. We can see that it annotates only predicate-argument roles and only for predicates "chairman" and "found", not for "entrepreneur" which is not encoded in any frame.

2.2 PropBank Semantic Roles

The FrameNet labels are rather rich in information, however, they might not always be transparent for users and annotators. The Proposition Bank (PropBank) lexicon was put forward first in 2000 to facilitate annotation, and later evolved into a resource on its own with the aim of adding a layer of semantic annotation to the Penn English TreeBank with verb-argument structure. So, the advantage of the PropBank approach is that by employing neutral labels, less effort is required from annotators to assign them. Furthermore, it creates the basis for the development of semi-automatic annotation of role labels, which is a necessary requirement if we want to annotate large corpora.

PropBank is constructed following a bottom-up strategy: starting from the various senses of a word, a frame-file is created for every verb. Such a frame-file contains thus all possible senses of the verb plus a set of example sentences that illustrate the context in which the verb can occur. For each sense of the verb, a role set and example sentences are available. The semantic roles covered by PropBank are the following:

Numbered arguments (*A0-A5, AA*): Because of the difficulty of defining a universal set of semantic or thematic

roles covering all types of predicates, PropBank defines semantic roles on a verb by verb basis. Semantic arguments of an individual verb are numbered, beginning with 0. For a particular verb, *Arg0* is generally the argument exhibiting features of a prototypical Agent while *Arg1* is a prototypical Patient or Theme. The meaning of each argument label depends on the usage of the verb in question.

Adjuncts (*AM*-): General arguments that any verb may take optionally. There are 13 types of adjuncts such as *AM-ADV* (general-purpose), *AM-TMP* (temporal).

Semantic role labeling processing

Based on the PropBank annotation system, given a sentence, the role labeling process goes through (1) identifies each verbal predicate and (2) labels its arguments.

Bill Gates is an American entrepreneur and the chairman of Microsoft, [$_{\text{ARG1}}$ *the software company] [*$_{\text{ARG0}}$ *he] [*$_{\text{rel}}$ *founded] [*$_{\text{AM\_MAN}}$ *with Paul Allen] [*$_{\text{AM-LOC}}$ *in Albuquerque, New Mexico] [*$_{\text{AM-TMP}}$ *on April 4, 1975.]*

Above is an example showing how to annotate a sentence with PropBank roles. We can see that PropBank focuses on verb predicate-argument roles.

2.3 Semantic Role Labeling Tasks

Gildea and Jurafsky[6] (2002) presented the first semantic role labeling system to apply a statistical learning technique based on the FrameNet data. They describe a discriminative model for determining the most probable role for a constituent given the predicate, the frame. This task has been the subject of a previous Senseval task (Automatic Semantic Role Labeling)[10] and two shared tasks on semantic role labeling in the Conference on Natural Language Learning (2004&2005)[3].

Systems contributed to Senseval shared task were evaluated to meet the same objectives as the Gildea and Jurafsky study using the FrameNet data. In Senseval-3 two different cases of automatic labeling of semantic roles were considered. The Unrestricted Case requires systems to assign semantic roles to the test sentences for which the boundaries of each role were given and the predicates identified. The Restricted Case requires systems to (i) recognize the boundaries of semantic roles for each evaluated frame as well as to (ii) assign a label to it. Eight teams participated in the task, with a total of 20 runs for two cases. The average precision over all Unrestricted Case runs is 0.803 and the average recall is 0.757. And the average precision over all Restricted Case runs is 0.595 and the average recall is 0.481 which is noticeably lower than the first case, indicating the additional difficulty of identifying the frame element boundaries.

CoNLL-2004, 2005 shared task evaluated SRL systems based on the PropBank corpus. Given a sentence, with a number of target verbs marked, a semantic role labeling system is to develop a machine learning system to recognize and label the arguments of each verb predicate. Nineteen systems participated in the CoNLL-2005 shared task. They approached the task in several ways, using different learning components and labeling strategies with different types of linguistic features, providing a comparative description and results. Evaluation is performed on a collection of unseen test sentences, that are marked with target verbs and contain only predicted input annotations, the best results in the shared task almost reached an F1 at 80 in the WSJ test set and almost 78 in the combined test.

3 CDL.nl Semantic Relation Extraction Task

Yokoi et al. (2005)[17] presented CDL.nl (Concept Description Language for Natural Language) which is used to describe the semantic/concept structure of text as a core member of W3C Common Web Language[1]. Different from existed dependency parsers which represent grammatical dependency structure of text, it is used to describe semantic dependency structure of plain text in graph form. The two basic elements for describing the structure are Entity and Relation, where the element Entity is used to represent a constituent of sentences with a head word. A set of relations[2] is defined to represent the meaning of the relationships between a pair of entities. The entity which heads the relation is called head entity and the other one is called tail entity. A lexicon named UNLKB is used to organize entities for CDL.nl according to their semantic behaviors which is based on their participated relations, more details about the lexicon are shown in Section 4.2.3.

3.1 CDL.nl Semantic Relation Set

With similar objectives as PropBank to add a layer of semantic annotation on natural language sentences, but different from roles in PropBank, where role semantics depends on the verb and verb usage, or verb sense in a sentence, CDL.nl predefines a set of semantic relations. And additional information for distinguishing from similar relations is also described. For example, the definition of *aoj*(nominal entity with attribute) contains two parts:

Definition: *aoj* indicates a nominal thing that is in a state or has an attribute.

Differences between related relations: A thing with an attribute is different from *mod* in that *mod* gives some restriction of the concept in focus, while *aoj* indicates a thing of a state or characteristic.

Example: for the short sentence " Leaves are green", there is a relation typed as *aoj* between green and leaves, so machine can understand that "leaves" here have the attribute "green".

[1] http://www.w3.org/2005/Incubator/cwl/

[2] http://www.miv.t.u-tokyo.ac.jp/mem/yyan/CDLnl/

Facing the challenge of defining a universal set of semantic or thematic relation covering all types of semantic relationships between entities, CDL.nl defines a set of semantic relations containing 44 relation types which are organized into three groups:

- **Intra-event relation**: Relations defining case roles, which are divided into 6 abstract relations, *QuasiAgent*, *QuasiObject*, *QuasiInstrument*, *QuasiPlace*, *QuasiState* and *QuasiTime*. And each abstract relation contains several concrete relations which express concrete semantic information. Such as *QuasiAgent* contains five semantic relations, *agt*(agent), *aoj*(thing with attribute), *cag*(co-agent), *cao*(co-thing with attribute), *ptn* (partner). To show the advantage of this sub-set relations, we take the *cag*(co-agent) as a example, in the sentence "John walks with Mary", "Mary" is the co-agent of event "walks", so we know the facts both "John walks" and "Mary walks".

- **Inter-entity relations**: In addition to event-specific numbered roles, CDL.nl defines 13 more general relation types that can apply to different types of head entity. As the definition of relation type *pur*(purpose) shows, besides action entity, NominalEntity can also activate *pur* relation. Other inter-entity relations are *pur*(purpose), *seq*(sequence), *equ*(equivalent), etc.

- **Qualification relations**: relations representing qualification relationships between modified entity and modifier entity. There are 9 qualification relations, containing *mod*(modification), *pos*(possessor), *qua*(quantity). This sub-set of relations is important to describe entity with more different properties.

Comparing to FrameNet, PropBank, CDL.nl relation set can be used to annotate not only facts in sentences about WHO did WHAT to WHOM or with WHOM, WHEN, WHERE, WHY, HOW, but also What has WHICH properties, and so on. A directed graph where Entity is regarded as node and Relation is regarded as arc, can be used to represent the semantic structure. Entity can be classified into elemental entity and composite entity. Composite Entity is a hyper node which contains structure of Entity and Relation within it. Unlike a hyper node in graph theory, however, nodes inside and outside Composite Entity may be linked with each other by a direct arc.

Figure 2 is an example showing the graph structure annotated with CDL.nl relations. Comparing to annotation with FrameNet and PropBank, it supports our idea that with CDL.nl relation set, plain sentences can be annotated with not only predicate-argument relations, but also those between each pair of entities there exists a meaningful relationship, such as the *equ*(equivalent) relation between entities "Microsoft" and "the software company"

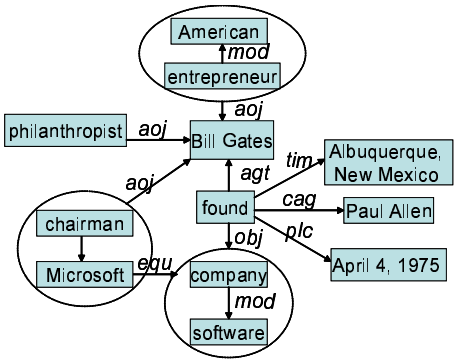

Figure 2. The graph structure of sentence "*Bill Gates is an American entrepreneur, philanthropist and chairman of Microsoft, the software company he founded with Paul Allen in Albuquerque, New Mexico on April 4, 1975.***"**

shows that both refer to the same object, and *aoj*(thing with attribute) relation between "American entrepreneur" and "Gates" showing that "Gates" has an attribute as "American entrepreneur".

3.2 Challenges of Automatic CDL.nl Relation Extraction

Task of structure annotation with CDL.nl relation set can be turn to be a relation extraction process which can be divided into two steps: relation detection–detecting entity pairs between each pair there exists a meaningful relationship; relation classification–labeling each detected entity pair with a specific relation.

Considering the first step, semantic role detection in SRL systems involve only classifying each syntactic element in a sentence into either a semantic argument or a non-argument by giving a predicate, so it is a binary-classification problem. But the task of detecting of CDL.nl relation is strictly not a classification problem, and conceptually, the system has to consider all possible subsequence (i.e., consecutive words) pairs in a sentence. To this respect, the detection of dependency relations is similar to our relation detection task. As evident from the CoNLL-X shared task on dependency parsing [2], there are currently two dominant models for data-driven dependency parsing. The first is "all-pairs" approach, where every possible arc is considered in the construction of the optimal parse. The second is the "stepwise" approach, where the optimal parse is built stepwise and where the subset of possible arcs considered depend on previous decisions. Clearly, "all-pairs" approach requires exponential time in its worst case. And while "stepwise" approach builds parse depending on previous decisions, our task of CDL.nl relations annotated in sentences are relatively independent from each other. So, the challenges of

our first step of relation extraction is that we need a efficient method which is adequate for independent relation detection considering all possible subsequences.

For the second step, while semantic role classification involves classifying each semantic argument identified into a specific semantic role, our relation classification task involves assigning a specific CDL.nl relation to each detected entity pair to form the graph structure of the sentence. The challenges are: 1, we have to consider all 44 relation types at the same time; 2, one major problem faced by semantic annotation of text is the fact that similar syntactic patterns may introduce different semantic interpretations and similar meanings can be syntactically realized in many different ways.

4 A Hybrid Approach for Automatic Relation Extraction

Facing the above challenges of extracting CDL.nl relations, in this Section, we present a hybrid approach with different methods for two steps: firstly, based on dependency analysis, a rule-based method is put forward for relation detection; secondly, we use a feature-based method to assign CDL.nl relation to each detected entity pair.

4.1 Rule-based Entity Pair Identification

The language processing has been going through syntactic processing, dependency analysis, shallow semantic parsing. To find relationship between entities in the level of semantic processing, as dependency analysis shows the head-modifier relations between words in the level of surface-syntactic processing in a word-to-word way, we use it as the base to perform our relation detection task.

In dependency parsing[16], the task is to create links between words and names the links according to their syntactic function. By identifying the syntactic head of each word in the sentence, the analysis result is represented in a dependency graph, where the nodes are the words of the input sentence and the arcs are the binary relations from head to dependent. Often, but not always, it is assumed that all words except one have a syntactic head, which means that the graph will be a tree with the single independent word as the root. In labeled dependency parsing, a specific type (or label) is assigned to each dependency relation holding between a head word and a dependent word.

Different from "all-pairs" and "stepwise" approaches, based on dependency tree structure generated from Connexor dependency parser[3], we present a rule-based method for relation detection done with a simple algorithm and it is illustrated with Figure 3:

[3]www.connexor.com

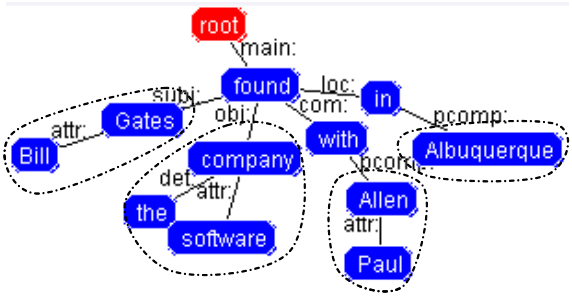

Figure 3. Syntactic analysis example from Connexor

- Step 1: generate a dependency tree for each input sentence, it specifies the syntactic head of each word in the sentence.

- Step 2: find a headNode set from the dependency tree, each of it can be a headword of a head entity to govern a relation. We select nodes with subtrees and also omit those which cannot be headNodes by creating a head stoplist.

- Step 3: for each headNode, check each of its subtrees to find those which can be tail entities related to the headNode. We create a tail stoplist containing those cannot be root nodes of subtrees of tail entities. If the root node of a subtree is in the tail stoplist, we continue to check the immediate grandchildren until reaching the leave nodes.

- Setp 4: a simple post-processing is applied to correct the boundaries where the dependency tree does not show right the relationships.

As shown in Figure 3, for the sentence "Bill Gates found the software company with Paul Allen in Albuquerque", from the dependency tree, the follow entity boundaries are generated from the dependency tree: found, (Bill Gates), found, (the software company), found, (Paul Allen), found, Albuquerque and company, software.

4.2 Machine Learning Method for Relation Classification

With all entity pairs have been detected, facing the challenges of labeling each pair with a specific CDL.nl relation, we describe a feature-based relation classification method which uses features to represent diverse knowledge of three levels of language processing: syntactic analysis, dependency parsing and lexical construction.

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	Bill	bill	attr:>2	@A> %>N N NOM SG
2	Gates	gates	subj:>3	@SUBJ %NH N NOM SG
3	found	find	main:>0	@+FMAINV %VA V PAST
4	the	the	det:>6	@DN> %>N DET
5	software	software	attr:>6	@A> %>N N NOM SG
6	company	company	obj:>3	@OBJ %NH N NOM
7	with	with	com:>3	@ADVL %EH PREP
8	Paul	paul	attr:>9	@A> %>N N NOM SG
9	Allen	allen	pcomp:>7	@<P %NH N NOM SG
10	in	in	loc:>3	@ADVL %EH PREP
11	Albuquerque	albuquerque	pcomp:>10	@<P %NH N NOM SG
12	<s>	<s>		

Figure 4. Syntactic analysis example from Connexor

4.2.1 Syntactic Features

Benefit from the Connexor Parser, richful linguistic tags can be extracted as features to classify relations between entities. For each pair of entities of relation instances, we extract a syntactic feature set F_S containing the following features:

Morphology Features: Morphological information tells the details of word forms used in text. For example, for **Noun** words, there are five tags: *N*(noun), *SG*(sigular), *PL*(plural), *NOM*(nominative) and *GEN*(genitive). Connexor Parser defines 70 morphology tags.

Syntax Features: Whereas morphology gives information on forms of words, syntax describes both surface syntactic and syntactic function information of words. For example, *%NH* (nominal head) and *%>N* (determiner or premodifier of a nominal) are surface syntactic tags, *@SUB* (Subject) and *@F-SUBJ* (Formal subject) are syntactic function tags. Connexor Parser defines 40 Syntax tags.

4.2.2 Dependency Features

For each pair of entities of relation instances, to extract a dependency feature set F_D, we define a dependency token $DT = (dep, path)$, where dep contains two labels, one is the first depend label in the dependency path which is governed directly by the headword of head entity; the other is the final label in the dependency path accepted by the headword of participant entity, since both of them are closest to represent direct dependency functions of the entity pair. $path$ is the path in the parse tree from the head entity to the other entity.

Figure 4 and show some examples of syntactic and dependency information of the sentence "*Bill Gates found the software company with Paul Allen in Albuquerque*". The 4th Column named syntactic relation of Figure 4 shows dependency relations.

4.2.3 Lexical Features

To face the problem that similar syntactic patterns may introduce different semantic interpretations. In this section, we use lexical meaning knowledge to deal with it. Lexical meaning knowledge contains two kinds of information: word sense and semantic behavior[9].

Two lexical resources built with extensive human effort over years of work–WordNet and UNLKB are used to capture lexical meaning knowledge. Each resource encodes a different kind of knowledge and has its own advantages. To explicitly capture these knowledge, a set of lexical feature F_L is extracted containing word-sense and word-behavior features for head words of entities:

Word-Sense Features

WordNet [5] is an on-line lexical system whose smallest unit is "synset", i.e. an equivalence class of word senses under the synonym relation. Synsets are organized by semantic relations such as Synonymy, Antonymy and Hyponymy. In WordNet 3.0, the total of all unique noun, verb, adjective, and adverb strings is actually 155287 along with 206941 word-sense pairs, containing 11529 verbs with 25047 verb-sense pairs. We use hypernymy and synonymy to represent word sense feature and also use synonymy to extend the later resource. Since each word may have many hypernym senses, we select the top four senses.

Take the word "chairman" for example, it has only one sense {*'chairman' in noun: president, chairman, chairwoman, chair, chairperson,*} which has eight levels of hierarchy senses, the top four are {{noun: living thing, animate thing}, {noun: object, physical object}, {noun: entity}, {noun: causal agent, cause, causal agency}}.

Word-Behavior Features

Based on CDL.nl semantic relation set, for each usage of the word, we define semantic behavior as a series of CDL.nl semantic relations in which the word participates. Since many words have different senses and usages, they may have several semantic behaviors. The UNLKB[4] is a lexicon which organizes words in a hierarchy structure form by their semantic behaviors. It covers nouns, verbs, adjectives and adverbs and also associates semantic relations in behavior representation with word type restrictions. The total of all word-behavior pairs is about 65000, containing 15000 verb-behavior pairs. It explicitly implements the close relationship between syntax and semantics for nouns, verbs, adjectives and adverbs. Here are some word-behavior pairs of word *give* in UNLKB:

give(agt>thing,obj>thing)
give(agt>thing,gol>person,obj>thing)
give(agt>thing,gol>thing,obj>thing)
give(agt>volitional thing,obj>action)

[4] www.undl.org/unlsys/uw/unlkb.htm

It shows that word *give* has at least these four kinds of semantic behaviors. And for the second behavior, it has *agent* relation with a thing-type word, *goal* relation with a person-type word and *object* relation with a thing-type word. Here type of a word is a hypernym word of the word.

Since UNLKB suffers from the coverage problem. We use the synonymy set from WordNet to extend them based on the assumption: words with same senses tend to share the same behaviors.

5 Experiments

5.1 Experimental Setting

Since this is the first work to extract CDL.nl relations from plain form text, currently there is no existed dataset for us to use for training and testing. After 46 person-days of discussion and manual annotation effort, we create a dataset[5] which contains about 1700 sentences. It was annotated with 13487 CDL.nl relations including 44 relation types. We evaluate the systems by using 10-fold cross validation with this dataset.

For the relation detection evaluation, a test file of 170 sentences is used. To evaluate the performance of the relation classification, we use one-vs-all scheme in which each binary classifier will be trained for each relation label. The classifier evaluation is carried out using the SVM-light software[8] with our syntactic, dependency and lexical features.

5.2 Preliminary Experimental Results

The aim of our experiments is twofold: on the one hand, we study the performance of rule-based relation detection method. On the other hand, we evaluate our feature set for relation classification with SVM. For both of the purpose, three widely used evaluation measures (precision, recall and F-value) are computed.

- **Evaluation on rule-based relation detection**

For the first purpose of evaluation, the following quantities are considered to compute precision, recall and F-value:

- p = the number of detected entity pairs.
- p+ = the number of detected entity pairs which are actual entity pairs.
- n = the number of actual entity pairs.

[5] http://www.miv.t.u-tokyo.ac.jp/mem/yyan/CDLnl/

Table 1. Evaluation on rule-based relation detection

Task	Precision	Recall	F-value
RelationDetection	62.65	68.33	65.37

Table 2. Preliminary performance of using different features

Kernel	Precision	Recall	F-value
K_S	79.33	85.78	82.43
K_D	**83.62**	**83.56**	**83.59**
K_L	73.49	81.63	77.35
K_{S+D}	**85.63**	**85.91**	**85.77**
K_{S+D+L}	**86.35**	**87.43**	**86.89**

Precision(P) = p+/p Recall(R) = p+/n
F-value(F) = 2*P*R/(P + R)

The results of evaluating the test file are shown in Table 1. The performance is a not high. Through error analysis of the detection results, we conclude the reasons may be: 1, some special phrases should be treated as elemental entities, while our algorithm still generates entity pairs inside of these phrases. 2, at the level of semantic information processing, we are trying to find deeper relationships than surface function relations. In some cases, when surface analysis is not able to reflect deep semantic information directly, we need to improve our detection method. 3, some of the detection errors resulted from failures by the dependency parser.

- **Evaluation on feature-based relation classification**

For the second purpose of evaluating the performance of features for relation classification, first assuming that relations have been detected correctly, we test three feature set separately and the following two simple combination set:

$$F_{SD} = F_S \bigcup F_D$$

Combination of syntactic and dependency features.

$$F_{SDL} = F_S \bigcup F_D \bigcup F_L$$

Combination of syntactic, dependency and lexical features.

From Table 2 we can get two observations, one is that using different feature set, the performance is different. This shows that each set contributes differently to our task. Another observation is that adding features continuously can improve the performance, which indicates they provide additional clues to the previous setup. While syntax features treat two entities as independent entities; the dependency features introduce dependency connection with grammatical function information between entities. The lexical features introduce the meanings of entities, it helps in distinguishing semantic relations in case of same syntactic and

Table 3. Overall performance of relation extraction

TASK	Precision	Recall	F-value
Relation Detection(RD)	62.65	68.33	65.37
Relation Classification(RC)	86.35	87.43	86.89
RD + RC	51.62	57.94	54.60

dependency features using word sense and usage information, and so by adding then into the feature vector, the performance is boosted.

- **Overall performance of relation extraction**

Table 3 shows the preliminary result of combining two steps. Though the performance of relation classification step is quite adequate, the performance of relation detection is relatively low. We can see that although facing so many challenges, CDL.nl relations can be extracted by our approach with performance that Precision, Recall, F-value are 51.62, 57.94, 54.60 respectively. Data analysis reveals that beside dependency analysis, our method of relation detection can be improved by integrating diverse information from different levels of natural language processing.

6 Conclusions

In this paper, facing challenges of semantic annotation of Web text, we have described a new parser of which (1) we used a new set of semantic relations of CDL.nl which are more competent than that of SRL to represent the semantic structure of text in a graph representation and (2) we proposed a hybrid relation extraction approach with two different methods: firstly, based on dependency analysis, a rule-based method is presented to detect all entity pairs between each of pair there exits a relationship; secondly, we use a feature-based method to assign CDL.nl relation to each detected entity pair from different levels of natural language processing. Experiments on our manual dataset showed that the our approach works better on relation classification than on relation detection which can be improved by integrating diverse levels of information from natural language processing.

References

[1] Baker, C.F.; Fillmore, C.J.; and Lowe, J.B. The Berkeley FrameNet Project. *In Proc. COLING-ACL-98.*

[2] Buchholz, S.; C.J.; and Marsi, E. CoNLL-X shared task on Multilingual Dependency Parsing *In Proc. COLING-X-06.*

[3] Carreras, X.; and Marquez, L. Introduction to the CoNLL-2005 shared task: Semantic role labeling. *In Proc. CoNLL-05.*

[4] Cimiano, P.; Erdmann, M.; and Ladwig, G. Corpus-based Pattern Induction for a Knowledge-based Question Answering Approach. *In Proc. ICSC-07.*

[5] Fellbaum, C. WordNet: An electronic lexical database. *Cambridge, MA: MIT Press, 1998.*

[6] Gildea, D.; and Jurafsky, D. Automatic Labeling of Semantic Roles. *Computational Linguistics, 28:3, pages 245-288, 2002*

[7] Harabagiu, S.; Bejan, C.A.; and Morarescu, P. Shallow semantics for relation extraction. *In Proc. IJCAI-05.*

[8] Joachims, T. Text Categorization with Support Vecor Machine: learning with many relevant features. *In Proc. ECML-98.*

[9] Levin, B. English Verb Classes and Alternation: A Preliminary Investigation. *The University of Chicago Press, 1993.*

[10] Litkowski, K. Senseval-3 task automatic labeling of semantic roles. *In Senseval-3.*

[11] Marvin, M. Semantic Information Processing. *MIT Press, Cambridge, MA.*

[12] Meyers, A.; Reeves, R.; Macleod, C.; et al. Annotating Noun Argument Structure for NomBank. *In Proc. LREC-04.*

[13] Miller, S.; Fox, H.; Ramshaw, L.; and Weischedel, R. A novel use of statistical parsing to extract information from text. *In 6th Applied Natural Language Processing Conference.*

[14] Narayanan, S.; Harabagiu, S. Question answering based on semantic structures. *In Proc. COLING-04.*

[15] Palmer, M.; Gildea, D.; and Kingsbury, P. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics, 31(1).*

[16] Tapanainen, P.; and Jarvinen, T. A non-projective dependency parser. *In Proc. ANLP-97, Washington, D.C.*

[17] Yokoi, T.; Yasuhara, H.; Uchida, H.; et al. CDL (Concept Description Language): A Common Language for Semantic Computing. *In WWW2005 Workshop on the Semantic Computing Initiative (SeC2005).*