

# A WordNet Based Rule Generalization Engine In Meaning Extraction System \*

Joyce Yue Chai and Alan W. Biermann

Department of Computer Science  
Box 90129, Duke University  
Durham, NC 27708-0129  
Internet: {chai,awb}@cs.duke.edu

**Abstract.** This paper presents a rule based methodology for efficiently creating meaning extraction systems. The methodology allows a user to scan sample texts in a domain to be processed and to create meaning extraction rules that specifically address his or her needs. Then it automatically generalizes the rules using the power of the WordNet system so that they can effectively extract a broad class of information even though they were based on extraction from a few very specific articles. Finally, the generalized rules can be applied to large databases of text to do the translation that will extract the particular information the user desires. A recently developed mechanism is presented that uses the strategy of over-generalizing to achieve high recall (with low precision) and then selectively specializing to bring the precision up to acceptable levels.

## 1 Introduction

The tremendous topics available on Internet give rise to the demand for an easily adaptable meaning extraction system for different domains. Adapting an extraction system to a new domain has proved to be a difficult and tedious process. Many research groups have taken steps towards customizing information extraction systems efficiently, such as BBN [10], NYU [6], SRI [2], SRA [7], MITRE [1], UMass [5], etc. In a rule based meaning extraction system, ideally one would like to have both unambiguous rules and generalized rules. In this way, the target information can be precisely activated by the unambiguous rules, and at the same time, the human effort involved in enumerating all the possible ways of expressing the target information can be eliminated by the generalized rules. However, practically, it's very hard to achieve both.

We have proposed a rule generalization approach and implemented it in our trainable meaning extraction system. The system allows the user to train on a small amount of data in the domain and creates the specific rules. The rule generalization routines will generalize the specific rules to make them general for the new information. In this way, rule generalization makes the customization

---

\* This work has been supported by a Fellowship from IBM Corporation.

for a new domain easier by eliminating the effort in creating all the possible rules.

This paper describes the automated rule generalization method and the usage of WordNet [8]. First, it gives a brief introduction to WordNet and investigates the possibility of using WordNet to achieve generalization; then it presents experimental results based on the idea of generalization; finally, it illustrates an augmented generalization method for controlling the degree of generalization based on the user's needs.

## 2 Overview of System

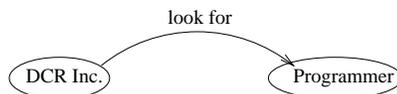
The system contains three major subsystems which, respectively, address training, rule generalization, and the scanning of new information. First, each article is partially parsed and segmented into Noun Phrases, Verb Phrases and Prepositional Phrases. An IBM LanguageWare English Dictionary and Computing Term Dictionary, a Partial Parser<sup>2</sup>, a Tokenizer and a Preprocessor are used in the parsing process. The Tokenizer and the Preprocessor are designed to identify some special categories such as e-mail address, phone number, state and city etc. In the training process, the user, with the help of a graphical user interface(GUI) scans a parsed sample article and indicates a series of semantic net nodes and transitions that he or she would like to create to represent the information of interest. Specifically, the user designates those noun phrases in the article that are of interest and uses the interface commands to translate them into semantic net nodes. Furthermore, the user designates verb phrases and prepositions that relate the noun phrases and uses commands to translate them into semantic net transitions between nodes. In the process, the user indicates the desired translation of the specific information of interest into semantic net form that can easily be processed by the machine. For each headword in a noun phrase, WordNet is used to provide sense information. For headwords with senses other than sense one, the user needs to identify the appropriate senses, and the Sense Classifier will keep the record of these headwords and their most frequently used senses. When the user takes the action to create the semantic transitions, a Rule Generator keeps track of the user's moves and creates the rules automatically. These rules are specific to the training articles and they need to be generalized in order to be applied on other articles in the domain. The rule generalization process will be explained in the later sections. During the scanning of new information, with the help of a rule matching routine, the system applies the generalized rules to a large number of unseen articles from the domain. The output of the system is a set of semantic transitions for each article that specifically extract information of interest to the user. Those transitions can then be used by a Postprocessor to fill templates, answer queries, or generate abstracts [3].

---

<sup>2</sup> We wish to thank Jerry Hobbs of SRI for providing us with the finite-state rules for the parser.

Original Training Sentence: DCR Inc. is looking for C programmers.

Semantic Transition Built by the User through GUI:



Specific Rule Automatically Created by the Rule Generator:

[DCR Inc., NG, 1, company], [look\_for, VG, 1, other\_type], [programmer, NG, 1, other\_type]  $\longrightarrow$   
ADD\_NODE(DCR Inc.), ADD\_NODE(programmer), ADD\_RELATION(look\_for, DCR Inc., programmer)

Fig. 1. Semantic Transition and Specific Rule

### 3 Rule Generalization

#### 3.1 Rules

In a typical information extraction task, the most interesting part is the events and relationships holding among the events [2]. These relationships are usually specified by verbs and prepositions. Based on this observation, the left hand side (LHS) of our meaning extraction rules is made up of three entities. The first and the third entities are the target objects in the form of noun phrases, the second entity is the verb or prepositional phrase indicating the relationship between the two objects. The right hand side (RHS) of the rule consists of the operations required to create a semantic transition—ADD\_NODE, ADD\_RELATION. ADD\_NODE is to add an object in the transitions. ADD\_RELATION is to add a relationship between two objects. A semantic transition and its corresponding specific rule are shown in Fig. 1.

The specific rule in Fig. 1 can only be activated by a sentence with the same pattern as “DCR Inc. is looking for C programmers . . .”. It will not be activated by other sentences such as “IBM Corporation seeks job candidates in Louisville, KY with HTML experience”. Semantically speaking, these two sentences are very much alike. Both are expressing a fact that a company seeks professional people. However, without generalization, the second sentence will not be processed. So the use of the specific rule is very limited.

#### 3.2 WordNet and Generalization

**Introduction to WordNet** WordNet is a large-scale on-line dictionary developed by George Miller and colleagues at Princeton University [8]. The most useful feature of WordNet to the Natural Language Processing community is its attempt to organize lexical information in terms of word meanings, rather than word forms. Each entry in WordNet is a concept represented by a list of synonyms—the synset. The information is encoded in the form of semantic networks. For instance, in the network for nouns, there are “part of”, “is\_a”, “member of” . . . relationships between concepts. The hierarchical organization of

An Abstract Specific Rule:

$$(w_1, c_1, s_1, t_1), (w_2, c_2, s_2, t_2), (w_3, c_3, s_3, t_3)$$

$$\longrightarrow \text{ADD\_NODE}(w_1), \text{ADD\_NODE}(w_3), \text{ADD\_RELATION}(w_2, w_1, w_3)$$

A Generalized Rule:

$$(W_1, C_1, S_1, T_1) \in \text{Generalize}(sp_1, h_1), (W_2, C_2, S_2, T_2) \in \text{Generalize}(sp_2, h_2),$$

$$(W_3, C_3, S_3, T_3) \in \text{Generalize}(sp_3, h_3)$$

$$\longrightarrow \text{ADD\_NODE}(W_1), \text{ADD\_NODE}(W_3), \text{ADD\_RELATION}(W_2, W_1, W_3)$$

**Fig. 2.** Sample Rules

WordNet by word meanings [8] [9] provides the opportunity for automated generalization. With the large amount of information in semantic classification and taxonomy provided in WordNet, many ways of incorporating WordNet semantic features with generalization are foreseeable. At this stage, we only concentrate on the Hypernym/Hyponym feature.

A hyponym is defined in [8] as follows: “A noun X is said to be a hyponym of a noun Y if we can say that *X is a kind of Y*. This relation generates a hierarchical tree structure, i.e., a taxonomy. A hyponym anywhere in the hierarchy can be said to be “a kind of” all of its superordinates. ...” If X is a hyponym of Y, then Y is a hypernym of X.

**Generalization** From the training process, the specific rules contain three entities on the LHS as shown in in Fig. 2. Each entity ( $sp$ ) is a quadruple, in the form of  $(w, c, s, t)$ , where  $w$  is the headword of the trained phrase;  $c$  is the part of the speech of the word;  $s$  is the sense number representing the meaning of  $w$ ;  $t$  is the semantic type identified by the preprocessor for  $w$ .

For each  $sp = (w, c, s, t)$ , if  $w$  exists in WordNet, then there is a corresponding synset in WordNet. The hyponym/hypernym hierarchical structure provides a way of locating the superordinate concepts of  $sp$ . By following additional Hypernymy, we will get more and more generalized concepts and eventually reach the most general concept, such as  $\{entity\}$ . Based on this scenario, for each concept, different degrees of generalization can be achieved by adjusting the distance between this concept and the most general concept in the WordNet hierarchy. The function to accomplish this task is  $Generalize(sp, h)$ , which returns a synset list  $h$  levels above the concept  $sp$  in the hierarchy.

The process of generalizing rules consists of replacing each  $sp = (w, c, s, t)$  in the specific rules by a more general superordinate synset from its hypernym hierarchy in WordNet by performing the  $Generalize(sp, h)$  function. The degree of generalization for rules varies with the variation of  $h$  in  $Generalize(sp, h)$ .

A generalized rule is shown in Fig. 2. The  $\in$  symbol signifies the subsumption relationship. Therefore,  $a \in b$  signifies that  $a$  is subsumed by  $b$ , or, in WordNet terms, concept  $b$  is a superordinate concept of concept  $a$ . The generalized rule states that the RHS of the rule gets executed if *all* of the following conditions hold:

- A sentence contains three phrases (not necessarily contiguous) with head-

- words  $W_1$ ,  $W_2$ , and  $W_3$ .
- The quadruples corresponding to these headwords are  $(W_1, C_1, S_1, T_1)$ ,  $(W_2, C_2, S_2, T_2)$ , and  $(W_3, C_3, S_3, T_3)$ .
  - The synsets, in WordNet, corresponding to the quadruples, are subsumed by  $Generalize(sp_1, h_1)$ ,  $Generalize(sp_2, h_2)$ , and  $Generalize(sp_3, h_3)$  respectively.

During the scanning process, the generalized rules are used to create semantic transitions for new information.

### 3.3 Experiments and Discussion

We have conducted a set of experiments based on seven levels of generalization. We set the `MAX_DEPTH` to 6. At degree 0, if entity one and/or entity three in the rule occurred lower than depth 6 in the WordNet hierarchy, we generalized them to their hypernym at depth 6. At degree 1, two object entities that appear lower than depth 5 in the hierarchy were generalized to their hypernym at depth 5. At degree  $i$  ( $0 \leq i \leq 6$ ), the object entities in the rules with depths greater than  $(MAX\_DEPTH - i)$  were generalized to their Hypernymy at depth  $(MAX\_DEPTH - i)$ .

The system was trained on three sets of articles from the *triangle.jobs* USENET newsgroup, with emphasis on the following seven facts:

- Company Name. Examples: IBM, Metro Information Services, DCR Inc.
- Position/Title. Examples: programmer, financial analyst, software engineer.
- Experience/Skill. Example: 5 years experience in Oracle.
- Location. Examples: Winston-Salem, North Carolina.
- Benefit. Examples: company matching funds, comprehensive health plan.
- Salary. Examples: \$32/hr, 60K.
- Contact Info. Examples: Fax is 919-660-6519, e-mail address.

The first training set contained 8 articles; the second set contained 16 articles including the first set; and the third set contained 24 articles including those in the first two sets. For rules from each training set, seven levels of generalization were performed. Based on the generalized rules at each level, the system was run on 80 unseen articles from the same newsgroup to test its performance on the extraction of the seven facts.

The evaluation process consisted of the following step: first, each unseen article was studied to see how many facts of interest were present in the article; second, the semantic transitions produced by the system were examined to see if they correctly caught any facts of interest. Precision is the number of transitions correctly conveying certain semantic information out of the total number of transitions produced by the system; recall is the number of facts correctly embodied in the transitions out of the total number of facts present in the articles.

Precision decreases from 96.1% to 68.4% for the first training set as the degree of generalization increases from 0 to 6. The first set of eight training articles has better performance on precision than the other two sets. For the third

training set of 24 articles, recall increases from 48.2% to 76.1% as generalization degree increases. As expected, the third training set out-performed the other two training sets on recall. The overall performance of recall and precision is defined by F-measurement [4], which is

$$\frac{(\beta^2 + 1.0) * P * R}{\beta^2 * P + R}$$

where  $P$  is precision,  $R$  is recall,  $\beta = 1$  if precision and recall are equally important. The F-measurement with respect to the degree of generalization on three different training sets is shown in Fig. 3. The F-measurement for the second and the third training sets reaches its peak when generalization degree is 5, which suggests that more generalization doesn't necessarily provide better performance.

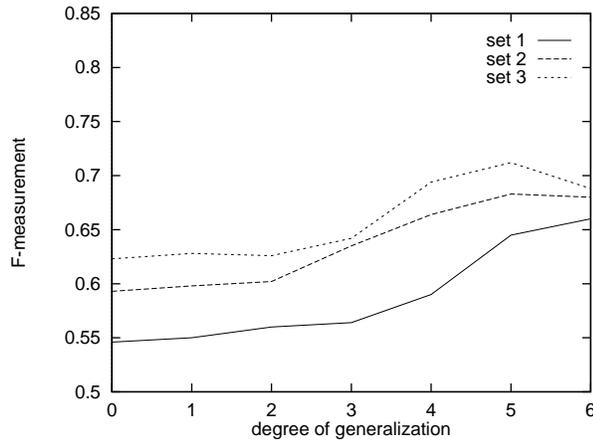


Fig. 3. F Measurement vs. generalization degree

The amount of training affects the performance too. Fig. 4 shows the F-measurement with respect to the amount of training. The outermost curve is for generalization degree 6, and the innermost curve is for degree 0. It shows that, for a specific domain, by applying the generalization approach, an enormous amount of training is not absolutely necessary. There will be a certain threshold for the F-measurement.

The effect of generalization degree on individual facts is shown in Fig. 5. For different fact extractions, generalized rules performed differently. The degree of generalization had the biggest impact in the extraction of *position/title*. The recall jumped from 31.6% to 82.5% when degree increased from 0 to 6. Some other facts such as *salary* were not changed much by the generalization. The recall did increase, but not greatly, only from 20% to 26.7%. This indicates the effect of generalization varies among different facts. It is more effective in extracting the

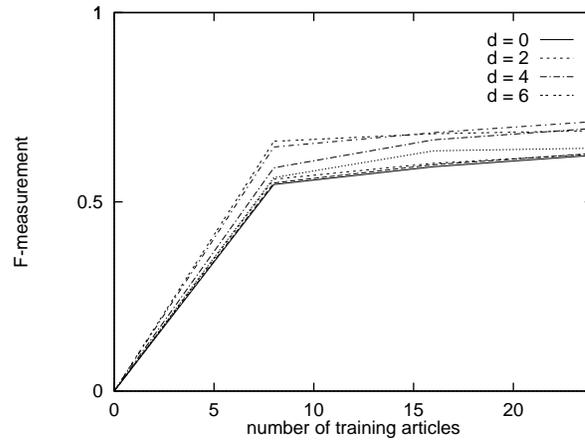


Fig. 4. recall vs. training set at different degree

fact, such as *position/title*, that is expressed in a learnable, comparably small set of pattern structures, with variations on the contents of the structures.

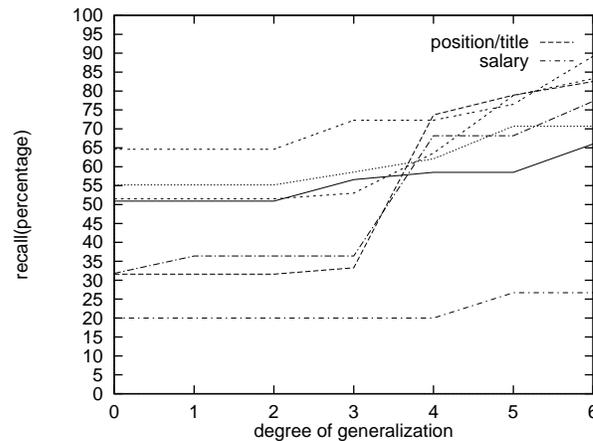


Fig. 5. recall of extracting individual fact vs. degree of generalization

Moreover, with the increase in the degree of generalization, precision tends to fall while recall tends to increase. The question that arises here is: What degree of generalization gives us the best compromise between precision and recall? If the user prefers high recall and doesn't care too much about the precision, or vice-versa, is there any way to control the generalization level in order to meet the user's needs?

## 4 Augmented Rule Generalization

An augmented generalization approach is introduced to find the optimal level of generalization based on user's special needs.

### 4.1 Tunable Generalization Engine

Rules with different degrees of generalization on their different constituents will have a different behavior when processing new information. Within a particular rule, the user might expect one entity to be relatively specific and the other entities to be more general. For example, if a user is interested in finding all DCR Inc. related jobs, the first entity should stay as specific as that in Fig. 1, and the third entity should be generalized. We have designed a Tunable Rule Generalization Engine to control the generalization degree. The engine consists of the following parts:

- Complete Rule Generalization Routine.
- Interface for Relevant Transitions.
- Statistical Classifier
- Rule Tuner

**Complete Rule Generalization Routine** For each specific rule, Complete Rule Generalization Routine locates the most general concepts for both the first and the third entities, and makes the specific rule the most general rule. A specific rule and its most general rule are shown in Fig. 6.

**Interface for Relevant Transitions** The most general rules are applied to the training corpus and a set of semantic transitions are created. Some transitions are relevant while the others are not. Users are expected to select the relevant transitions through a user interface. The system will keep a database of transitions and user selections. A sample portion of the database is shown in Fig. 6. When the most general rules are applied to extract useful information, the system achieves the highest recall, and the lowest precision.

**Statistical Classifier** The statistical classifier starts with the database of transitions and relevant information. For each most general rule  $R_i$ , the statistical classifier will calculate the following probabilities:

$$\begin{aligned} \text{Relevancy\_Rate}(R_i) &= \frac{\text{number of relevant transitions created}}{\text{total number of transitions created}} \\ \text{Object\_1\_Relevancy\_Rate}(R_i) &= \frac{\text{number of relevant object\_1 created}}{\text{total number of object\_1 created}} \\ \text{Object\_2\_Relevancy\_Rate}(R_i) &= \frac{\text{number of relevant object\_2 given relevant object\_1}}{\text{total number of relevant object\_1 created}} \end{aligned}$$

$\text{Relevancy\_Rate}(R_i)$  is the measure of how well the most general rule  $R_i$  performs on extracting the relevant information. Very high  $\text{Relevancy\_Rate}(R_i)$

Specific Rule:

[degree, NG, 3, other\_type], [in, PG, 0, other\_type], [field, NG, 3, other\_type]  
 ADD\_NODE(degree), ADD\_NODE(field), ADD\_RELATION(in, degree, field)

Most General Rule:

$(W_1, C_1, S_1, T_1) \in \{abstraction\}, (W_2, C_2, S_2, T_2) \in \{in\}, (W_3, C_3, S_3, T_3) \in \{psychological\_feature\}$   
 ADD\_NODE( $W_1$ ), ADD\_NODE( $W_3$ ), ADD\_RELATION( $W_2, W_1, W_3$ )

Database of Transitions Created by the Most General Rule:

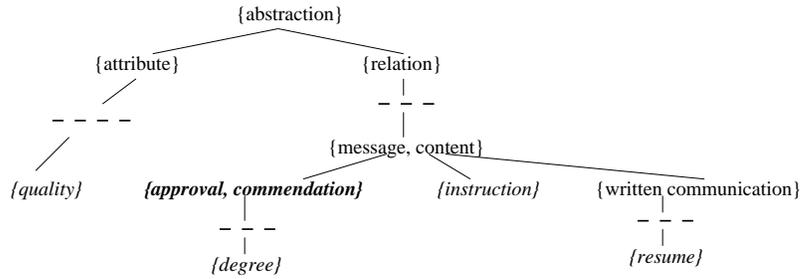
index	obj1	obj1 relevant	relation	obj2	obj2 relevant	count
1	quality	no	in	technical issues	no	1
2	instruction	no	in	instruction program	no	1
3	one degree	yes	in	health related field	yes	1
4	BS	yes	in	technical discipline	yes	2
5	your resume	no	in	graphical preference	no	1
6	BS	yes	in	science	yes	1

**Fig. 6.** Database of Transitions

such as 97% suggests that the most general rule  $R_i$  does not produce much over-generation in this domain. It implies that  $R_i$  can bring very high recall, but is not responsible for the low precision. This rule can be kept in the rule base for future use without further tuning.

If the *Relevancy\_Rate* is low and beyond the user’s tolerance, some actions should be taken to make the rule less general. If *Object\_1\_Relevancy\_Rate*( $R_i$ ) is lower than the user’s tolerance, the first entity in the rule needs to be tuned. If *Object\_2\_Relevancy\_Rate*( $R_i$ ) is lower than the user’s tolerance, the third entity in the rule needs to be tuned. The tuning will be done by Rule Tuner.

**Rule Tuner** For each entity in the most general rule that has been identified for tuning by Statistical Classifier, Rule Tuner will make it more specific to the user’s interests. For example, in Fig. 6, the Statistical Classifier decides that {abstraction} in the most general rule is too general, then the Rule Tuner will start to put constraints on this entity by decreasing the generalization degree of the original specific concept {degree}. Since {instruction}, {quality}, {resume} are irrelevant concepts, we need to find a most general hypernym of {degree}, which is not the hypernym of {instruction}, {quality} and {resume}. From the hypernym hierarchy as shown in Fig. 7, {approval, commendation} is the desired hypernym. The concept {approval,commendation} will replace concept {abstraction} in the most general rule to form the optimally generalized rule. The concept such as {approval, commendation} in the example is more general than the original specific concept, and at the same time, is not responsible for the over-generation. We call this concept *Uppermost Relevant Concept*. For each entity in the rule which needs to be tuned, Rule Tuner will go through all the corresponding objects and find the *Uppermost Relevant Concept* for that entity and replace the original most general concept with the *Uppermost Rele-*



**Fig. 7.** Hypernym Hierarchy

*vant Concept*. After Rule Tuner examines every entity in every rule, a set of optimally generalized rules are created. The generalization level is different on each entity based on the user's interests.

## 4.2 Experiments and Results

We applied the optimally generalized rules created by the Tunable Generalization Engine on extracting *position/title* information from *triangle.job* newsgroup. The system was trained on 32 articles from the domain and 19 specific rules were created. Then we passed the rules to the Tunable Rule Generalization Engine and created a set of optimally generalized rules. We applied this set of optimal rules to 130 unseen articles.

Three more experiments were conducted to compare the results. One was to apply the specific rules to the unseen articles, another one was to apply the most generalized rules to the unseen articles, the third one was to apply the rules we manually generalized without the use of the Tunable Generalization Engine. The result is shown in Table 1, where precision is the number of relevant transitions out of the total number of transitions; recall is the number of *position/title* correctly fetched out of the total number of *position/title* that should be fetched.

When the specific rules were applied, the system reached the highest precision at 100%, but the lowest recall at 39%. When the most general rules were applied, the system achieved the highest recall at 70%, but the lowest precision at 27%. By using Tunable Generalization Engine, the optimized rules pushed up precision by 50% and only sacrificed 1% recall. Automatically optimized rules performed better than the manually optimized rules.

## 5 Conclusion and Future Work

This paper describes a generalization approach based on WordNet. The rule generalization makes it easier for the meaning extraction system to be customized to a new domain. Tunable Generalization Engine makes the system adaptable to the user's needs. The idea of first achieving the highest recall with low precision,

**Table 1.** Performance Comparison

	Specific Rules	Most General Rules	Manually Optimized Rules	Automatically Optimized Rules
Recall	39%	70%	65%	69%
Precision	100%	27%	75%	77%
F-Measure	56%	39%	70%	73%

then pushing up the precision while keeping the recall comparably steady has been successful. We are currently studying how to enhance the system performance by further refining the generalization approach.

## References

1. Aberdeen, John, et al.: Description of the *ALEMBIC* System Used for MUC-6, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 141-155, November 1995.
2. Appelt, Douglas E., et al.: SRI International: Description of the FASTUS System Used for MUC-6, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 237-248, November 1995.
3. Amit Bagga, Joyce Chai: A Trainable Message Understanding System, *to appear at ACL Workshop on Computational Natural Language Learning*, 1997.
4. Chinchor, Nancy: MUC-4 Evaluation Metrics, *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, June 1992, San Mateo: Morgan Kaufmann.
5. Fisher, David, et al.: Description of the UMass System as Used for MUC-6, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 127-140, November 1995.
6. Grishman, Ralph.: The NYU System for MUC-6 or Where's the Syntax? *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 167-175, November 1995.
7. Krupka, George R.: Description of the SRA System as Used for MUC-6, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 221-235, November 1995.
8. Miller, G.A., et al.: *Five Papers on WordNet*, Cognitive Science Laboratory, Princeton University, No. 43, July 1990.
9. Resnik, Philip: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *Proceedings of IJCAI-95*
10. Weischedel, Ralph.: BBN: Description of the PLUM System as Used for MUC-6, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 55-69, November 1995.