Hindawi Publishing Corporation
Journal of Biomedicine and Biotechnology
Volume 2008, Article ID 371710, 9 pages
doi:10.1155/2008/371710

*Research Article*

# Detection of Gene Interactions Based on Syntactic Relations

**Mi-Young Kim**

*School of Computer Science and Engineering, Sungshin Women's University, Seoul 136-742, Korea*

Correspondence should be addressed to Mi-Young Kim, miykim@sungshin.ac.kr

Received 29 August 2007; Accepted 19 December 2007

Recommended by Daniel Howard

Interactions between proteins and genes are considered essential in the description of biomolecular phenomena, and networks of interactions are applied in a system's biology approach. Recently, many studies have sought to extract information from biomolecular text using natural language processing technology. Previous studies have asserted that linguistic information is useful for improving the detection of gene interactions. In particular, syntactic relations among linguistic information are good for detecting gene interactions. However, previous systems give a reasonably good precision but poor recall. To improve recall without sacrificing precision, this paper proposes a three-phase method for detecting gene interactions based on syntactic relations. In the first phase, we retrieve syntactic encapsulation categories for each candidate agent and target. In the second phase, we construct a verb list that indicates the nature of the interaction between pairs of genes. In the last phase, we determine direction rules to detect which of two genes is the agent or target. Even without biomolecular knowledge, our method performs reasonably well using a small training dataset. While the first phase contributes to improve recall, the second and third phases contribute to improve precision. In the experimental results using ICML 05 Workshop on Learning Language in Logic (LLL05) data, our proposed method gave an F-measure of 67.2% for the test data, significantly outperforming previous methods. We also describe the contribution of each phase to the performance.



## 1. INTRODUCTION

Determining interactions between proteins and genes are essential in describing biomolecular phenomena [1]. Thus, many recent studies have sought to extract interaction information from biomolecular text using natural language processing technology. However, we have insufficient biomolecular data annotated with linguistic information. In 2005, the ICML05 Workshop on Learning Language in Logic (LLL05) task provided a small training dataset annotated with POS-tags and syntactic relations. This was an experimental challenge for gene interactions using linguistic information. Previous studies have insisted that linguistic information was useful for improving the detection of gene interactions. However, the experimental results for the LLL05 data gave a reasonable precision but poor recall. To improve recall without sacrificing precision, we propose a three-phase method to detect gene interactions using syntactic relation information, and apply it to a small training dataset lacking domain knowledge. Through experimentation, we show that our proposed method significantly outperforms existing methods, and describe the contribution of each phase to its performance.

This paper is organized as follows. Section 2 presents previous work on gene interactions. Section 3 explains our three-phase method in detail. Section 4 describes the training and test data used for our experiments and presents experimental results that demonstrate that our three-phase method is effective for detecting gene interactions. Finally, we provide our conclusions.

## 2. PREVIOUS WORK

The task of relation mining in the biomedical domain has been studied extensively in recent years. Current research includes protein-protein interactions [2, 3], subcellular locations [4], and disease-treatment relationships [5], and systems based on sequence modeling and pattern- or rule-based extraction best detect protein-protein interactions [2, 6, 7]. Using text mining technology for automatic protein(gene) interactions resulted in high precision, but low recall [8]. Many studies have used linguistic information to improve

performance in detecting gene interactions. To improve recall without sacrificing precision, Otasek et al. [8] expanded the diversity of sentence structures recognized by a syntactic parser through additional training, and Park et al. [9] presented a method using bidirectional incremental parsing. Experiments deduced 182 relations out of 492 sentences showing 48% recall and 80% precision. Many linguistic processes have been used to deduce gene interactions, including bidirectional incremental parsing, combinatory categorical grammar (CCG), coordination, apposition, compound noun processing, and positive/negative predicate learning. With these methods, linguistic information achieved reasonable precision, but still poor recall.

Blaschke et al. [10] assumed that sentences derived from sets of abstracts contained a significant number of protein names connected by verbs that indicate the type of relationship between them. They restricted the problem domain and imposed several strong assumptions that included prespecified protein names and a limited set of verbs to represent actions. Consequently, they constructed simple verb rules only for six proteins.

Several works examining gene interactions are based on LLL05 open data. Hakenberg et al. [11] used sentence alignment and finite-state automata optimized with a genetic algorithm. First, they applied a pattern-generating algorithm. Then, they learned patterns with finite-state automata based on a genetic algorithm. For example, "Agent1, Target3, Pattern2" implies that Agent1 interacts with Target3 via Pattern2. In biomolecular text, the agent or target can be encapsulated in another term based on some conditions, for example, apposition, modifying nouns, and so on. However, the method in [11] cannot deal with a situation in which genes are encapsulated in other terms via syntactic relations. They did not use linguistic information provided in the LLL05 data. Error analysis revealed that they wrongly detected an agent and its target in a pair of genes, although they correctly detected two genes that interact with each other. Linguistic information might correct this type of error.

Greenwood et al. [12] extracted patterns based on paths in MINIPAR dependency trees [13]. The nodes in the dependency trees from which patterns were derived were either a lexical item or a semantic category, such as a gene, protein, agent, or target. Patterns were learned using a weakly supervised bootstrapping method. They extended the patterns based on eight seed patterns and trained the model using the basic dataset without coreference, as provided by the LLL05 challenge organizers. The F-measure for the test data in LLL05 was 14.8%. The failure of the system to extract meaningful relations can be traced back to the errors that MINIPAR introduced in the dependency trees.

Goadrich et al. [14] used Gleaner as an inductive logic programming approach and further applied Brill Tagger, a shallow parser based on conditional random fields, and Porter stemmer. They also used much linguistic information, including sentence-structure predicates, the frequencies of words, lexical properties, and semantic knowledge using Mesh. The F-measure for the test data was 25.1%. Gleaner suffered from not distinguishing between an agent and a target well because no syntactic structure was used.

Popelinsky and Blatak [15] used Brill Tagger and WordNet, and Katrenko et al. [16] created a simple ontology specifically for use in the LLL05 challenge. However, they did not show reasonable recall.

Riedel and Klein [17] obtained the best performance on the LLL05 challenge task using syntactic chains. They assumed that clauses had to connect both genes transitively. Therefore, they generated a set of clauses based on chains of syntactic relations between two genes. The method achieved an F-measure of 52.6% on the dataset without coreferences, demonstrating that using syntactic information from the annotated datasets significantly improved performance. A CCG parser handled both POS-tagging and parsing. However, recall was only 46.2%, and the system needs to improve recall.

For GENIA and ATCR data, Rinaldi et al. [18] also used linguistic approach. They find agents and targets from the syntactic patterns directly connected with interaction verbs with subject or object functions. So, they do not consider the case that agent or target is encapsulated in another term, and indirectly connected with interaction verbs. In addition, there is a limit that they find agents and targets only from the subject and object relations.

Combining syntactic dependency information with features based on word sequences could lead to further improvements in performance, as demonstrated by the more recent approaches to relation extraction [19–21].

We build on the conclusion of the previous work that linguistic information, especially syntactic information, is an important key for detecting gene interactions. However, we need a more robust method to improve recall without sacrificing precision. Based on syntactic relation information, we propose a three-phase-based method for detecting gene interactions.

Greenwood et al. [12] mentioned the failure of the system to extract meaningful relations can be traced back to the errors of the applied syntactic analyzer. If we use the annotated LLL05 syntactic relation information, we cannot testify the robustness of our system in real time. So, we also experiment the performance of our system based on a real-syntactic analyzer.

To objectively compare the performance of our system with that of previous systems, we use LLL05 data. In the next section, we explain our proposed three-phase method in detail.

## 3. THREE-PHASE DETECTION OF GENE INTERACTIONS

Let us explain LLL05 data formats. The LLL05 challenge focuses on extracting information on gene interactions in *Bacillus subtilis*. The training dataset is decomposed into two subsets of increasing difficulty. The first subset does not include coreferences or ellipsis, unlike the second subset. The training set without coreferences consists of 55 sentences, including 106 examples of genic interactions. It contains 70 examples of action, 30 examples of binding and promoter, and 6 examples of regulation.

A syntactic relation is important linguistic information for detecting the structure of text. Algorithm 1 shows one

| | |
|---|---|
| **ID** | 11064201-3 |
| **sentence** | In this mutant, expression of the spoIIG gene, whose transcription depends on both sigma(A) and the phosphorylated Spo0A protein, Spo0A∼P, a major transcription factor during early stages of sporulation, was greatly reduced at 43 degrees C. |
| **words** | word(0,"In,"0,1) word(1,"this,"3,6) word(2,"mutant,"8,13) word(3,"expression,"16,25) word(4,"of,"27,28) word(5,"the,"30,32) word(6,"spoIIG,"34,39) word(7,"gene,"41,44) word(8,"whose,"47,51) word(9,"transcription,"53,65) word(10,"depends,"67,73) word(11,"on,"75,76) word(12,"both,"78,81) word(13,"sigma(A),"83,90) word(14,"and,"92,94) word(15,"the,"96,98) word(16,"phosphorylated,"100,113) word(17,"Spo0A,"115,119) word(18,"protein,"121,127) word(19,"Spo0A∼P,"130,136) word(20,"a,"139,139) word(21,"major,"141,145) word(22,"transcription,"147,159) word(23,"factor,"161,166) word(24,"during,"168,173) word(25,"early,"175,179) word(26,"stages,"181,186) word(27,"of,"188,189) word(28,"sporulation,"191,201) word(29,"was,"204,206) word(30,"greatly,"208,214) word(31,"reduced,"216,222) word(32,"at,"224,225) word(33,"43,"227,228) word(34,"degrees,"230,236) word(35,"C,"238,238) |
| **lemmas** | lemma(0,"in") lemma(1,"this") lemma(2,"mutant") lemma(3,"expression") lemma(4,"of") lemma(5,"the") lemma(6,"spoIIG") lemma(7,"gene") lemma(8,"whose") lemma(9,"transcription") lemma(10,"depend") lemma(11,"on") lemma(12,"both") lemma(13,"sigA") lemma(14,"and") lemma(15,"the") lemma(16,"phosphorylated") lemma(17,"spo0A") lemma(18,"protein") lemma(19,"Spo0A-P") lemma(20,"a") lemma(21,"major") lemma(22,"transcription") lemma(23,"factor") lemma(24,"during") lemma(25,"early") lemma(26,"stage") lemma(27,"of") lemma(28,"sporulation") lemma(29,"be") lemma(30,"greatly") lemma(31,"reduce") lemma(32,"at") lemma(33,"43") lemma(34,"degree") lemma(35,"C") |
| **syntactic_relations** | relation("subj:V_PASS-N,"31,3) relation("mod_att:N-N,"7,6) relation("mod_att:N-ADJ,"34,33) relation("comp_during:N-N,"23,26) relation("comp_of:N-N,"26,28) relation("comp_on:V-N,"10,13) relation("mod_att:N-N,"23,22) relation("mod_att:N-ADJ,"18,16) relation("mod:V_PASS-ADV,"31,30) relation("mod_att:N-ADJ,"26,25) relation("mod_att:N-ADJ,"23,21) relation("mod_att:N-N,"18,17) relation("comp_on:V-N,"10,18) relation("appos,"19,23) relation("subj:V-N,"10,9) relation("appos,"18,19) relation("comp_of:N-N,"3,7) relation("comp_in:V-N,"31,2) relation("comp_of:N-N,"9,7) relation("comp_at:V_PASS-N," 31,34) relation("mod_att:N-N,"34,35) |
| **agents** | agent(13) agent(17) |
| targets | target(6) |
| **genic_relations** | genic_interaction(13,6) genic_interaction(17,6) |

Algorithm 1: Example of LLL05 training data.

example of syntactic relations between two genes in the LLL05 data. The syntactic relations provided in LLL05 were of the form $relation(rel_i, w_i, w_j)$, where $rel_i$ is one of a fixed set of syntactic relations between $w_i$ and $w_j$ assigned by the LLL parser. The detailed contents about LLL05 training data are described in Algorithm 2.

Figure 1 shows an example of a syntactic path. In Figure 1, **Spo0A**(agent) goes through four terms to reach **spoIIG**(target). The chain of terms is ⟨**Spo0A**(agent) → protein(N) → depend(V) → transcription(N) → gene(N) → **spoIIG**(target)⟩. In the chain, node depend(V) is the verb that indicates the interaction between **Spo0A**(agent) and **spoIIG**(target). However, depend(V) has direct syntactic relations with protein(N) and transcription(N), not with **Spo0A**(agent) or **spoIIG**(target). In other words, **Spo0A**(agent) was encapsulated in protein(N) with the relation ("mod_att"), and **spoIIG**(target) was encapsulated in transcription(N) with the relation ("mod_att") and ("comp_of").

Without any domain knowledge of biomolecular text, we automatically detect gene interactions using syntactic relations annotated in the LLL05 data. In the first phase, to improve recall, we detect the relations that encapsulate an agent or target. In the second phase, we automatically extract "interaction verbs" that indicate interactions between two genes. Next, to improve precision, we must determine which of the two genes is the agent and which is the target. To determine the agent and target for two genes, we learn direction rules on the relations from agent to target in the third phase. The three phases are explained in detail from the next subsection.

**1> ID** : unique identifier

**2> sentence** : the original sentence

**3> words** : list of the sentence words
*transcription*-word (id_word, "string_word," start_word postion, end_word position)
ex> word(0,"Both,"0,3)

**4> lemmas** : normalized form of a word
*transcription*-lemma(id_word, "string_lemma")
ex> lemma(0,"Both")

**5> syntactic_relations** : syntactic relation between two words
*transcription*-relation("string_relation," id of the head, id of the dependent)
(a) string_relation is expressed as
"syntactic category:POStag of the head-POStag of the dependent."
ex> relation("mod_att:N-N,"8,7)
(b) POS-tags: V, V_PASS, N, ADJ, ADV
(c) syntactic categories: APPOS, COMP_prep, MOD, MOD_ATT, MOD_POST, MOD_PRED, NEG, OBJ, SUBJ

**6> agents** : agent of the genic interaction
*transcription*-agent(id of the word)

**7> targets** : target of the genic interaction
*transcription*-target (id of the word)

**8> genic_interactions** : an interaction between an agent and a target
*transcription*-genic_interaction(id of the agent, id of the target)

Please see http://genome.jouy.inra.fr/texte/LLLchallenge

Algorithm 2: Detailed contents of LLL05 training data.

### 3.1. Phase 1: constructing syntactic encapsulation categories for agents and targets

An agent or target gene is usually encapsulated in another term, and the verb that indicates the interaction between two genes has syntactic relations with two terms that encapsulate the genes. To improve recall for gene interactions, we must detect the encapsulation categories for candidate agents and targets. First, we find the syntactic chain from an agent to its target. In Figure 1, depend(V) is the verb that indicates an interaction between **Spo0A**(agent) and **spoIIG**(target). In this paper, we call the verb that indicates the interaction between an agent and its target an "interaction verb." As mentioned above, depend(V) has syntactic relations with protein(N) and transcription(N), but not with **Spo0A**(agent) or **spoIIG**(target). In a syntactic chain from an agent to its target, we call the node preceding an interaction verb a "metaagent," and the node following an interaction verb a "metatarget." In Figure 1, protein(N) is a metaagent, and transcription(N) is a metatarget.

We define the syntactic categories connecting an agent(target) and a metaagent(metatarget) "syntactic encapsulation categories." In Figure 1, mod_att and comp_of are examples of the syntactic encapsulation categories. To detect a metaagent and a metatarget, we should first identify an interaction verb in a syntactic chain. However, in the automatically obtained syntactic chains, we do not know which verb is an interaction verb. To overcome the problem, we extract the syntactic encapsulation categories from the syntactic chains that include only one verb in the training dataset.

### 3.2. Phase 2: extracting interaction verbs that indicate an interaction between two genes

To detect gene interactions, we must recognize the interaction verbs. In the second phase, we retrieve the interaction verbs that indicate an interaction between two genes. The verbs can be extracted while the first phase is performed. If we consider only the syntactic chains that contain only one verb, the size of the interaction verbs becomes very small. Since the LLL05 training dataset is small, we collect all the verbs in the syntactic chains from an agent to its target.

### 3.3. Phase 3: learning direction rules for detecting the agent and target in a pair of genes

According to the first and second phases, we can detect two genes that interact with each other.

Previous studies made many errors in attempts to recognize which of two genes was the agent or target. The incorrect detection of an agent and a target results in low precision. Therefore, a new method is required to recognize an agent and its target correctly in a pair of genes. In the third phase, we propose learning the directions of the syntactic relations in the syntactic path from an agent to its target. If we do not

Example sentence:
⟨In this mutant, expression of the spoIIG gene, whose transcription depends on both sigma(A) and the phosphorylated Spo0A protein, Spo0A∼P, a major transcription factor during early stages of sporulation, was greatly reduced at 43 degrees C.⟩

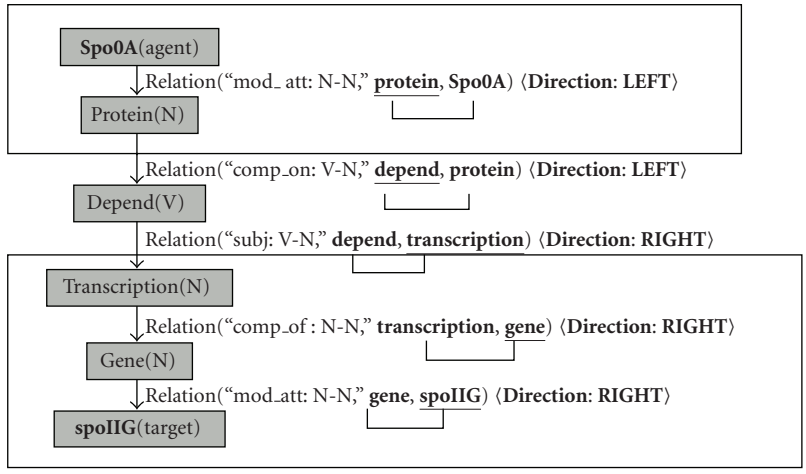


Figure 1: Example of a syntactic path based on LLL05 annotations.

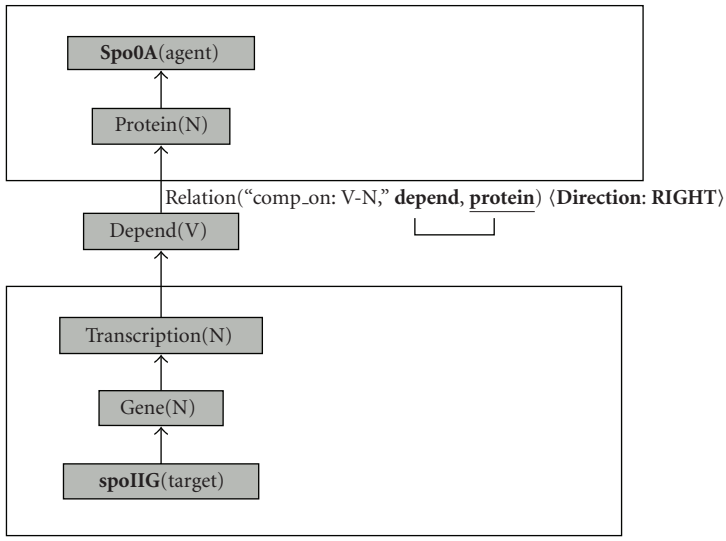


Figure 2: Reverse syntactic path of Figure 1 for a negative rule.

permit the reverse direction, the agent and target will not be detected wrongly and thus improve the precision.

We learn the direction of a syntactic relation related with an interaction verb. For a syntactic relation, direction is defined as follows. If a syntactic relation is *relation(syntactic category, current node, next node),* the direction is "RIGHT," since the next node is written to the right of the current node. If a syntactic relation is *relation(syntactic category, next node, current node),* the direction is "LEFT" because the next node is written to the left of the current node. Figure 1 also shows an example of direction information of a syntactic path. Among the directions, we retrieve only the direction information of an interaction verb.

The direction information is dependent on the syntactic category of the relation and the lexical word of the current node. In learning, we retrieve a syntactic category (a lexical word) and direction information for an interaction verb, and we make a template ⟨lexical word, syntactic category, direction⟩.

We construct direction information for all relations concerning interaction verbs in the training data. Based on the direction information, we learn direction rules. Let us explain the direction rule-learning algorithm, which is shown in Algorithm 3.

We obtain two types of rule set. One is a positive rule set obtained by learning the direction from an agent to its target.

**1> Alignment of a positive rule set**
(1.1) Collect ⟨lexical word A, relation B, direction C⟩ in the paths **from all Agents to their Targets**.
(1.2) For any lexical word A, and relation B, if both of ⟨A, B, RIGHT⟩ and ⟨A, B, LEFT⟩ exist in the positive rule set, we remove both rules, and add a modified rule ⟨A, B, ANY⟩.

**2> Alignment of a negative rule set**
(2.1) Collect ⟨lexical word A, relation B, direction C⟩ in the paths **from all Targets to their Agents**.
(2.2) For any lexical word A, and relation B, if both of ⟨A, B, RIGHT⟩ and ⟨A, B, LEFT⟩ exist in the negative rule set, we remove both rules, and add a modified rule ⟨A, B, ANY⟩.

**3> Construction of Final direction rules**
For every rule ⟨A, B, C⟩ in the positive rule set, for any lexical word A, relation B, and direction C.

(3.1) If ⟨A, B, C⟩ also exists in the negative rule set, we obtain a direction rule ⟨A, B, ANY⟩.
(3.2) Otherwise, if ⟨A, B, OPPOSITE C⟩ exists in the negative rule set, we obtain a direction rule ⟨A, B, C⟩.
(3.3) Otherwise, we obtain a direction rule ⟨A, B, C⟩.

Algorithm 3: Direction rule-learning algorithm.

Table 1: Positive and negative rule sets for the sentence in Figure 1.

| | |
|---|---|
| Positive rule | ⟨depend, subj, RIGHT⟩ |
| Negative rule | ⟨depend, comp_on, RIGHT⟩ |

The other is a negative rule set obtained by learning the direction from a target to its agent in reverse order. Figure 2 shows the reverse syntactic path from a target to its agent of the sentence in Figure 1. The positive and negative rules for the sentence in Figure 1 are shown in Table 1. From the positive and negative rule sets, we construct direction rules according to the following subsections.

#### 3.3.1. Alignment of positive/negative rule sets

First, we align the positive and negative rule sets. Here, "align" means the modification of any conflict in a rule set. For any lexical word A and relation B, if a conflict of two direction rules exists in a rule set, then we remove both rules, and add a modified rule ⟨A, B, ANY⟩. Because the direction information is not trustworthy, we set direction "ANY." "ANY" means any direction is okay. The process for aligning a rule set is shown in 1> and 2> of Algorithm 3.

#### 3.3.2. Construction of direction rules from positive and negative rule sets

After alignment of positive and negative rule sets, we construct direction rules from the two rule sets. The algorithm used to obtain direction rules is shown in 3> of Algorithm 3.

Consider every rule ⟨A, B, C⟩ in the positive rule set, for any lexical word A and relation B, and direction C.

In Algorithm 3, (3.1) case indicates that direction information C is changed to ANY. Since the same direction exists in both the positive and negative rule sets, the direction information is not trustworthy. Therefore, we change the direction information into ANY.

In (3.2) case, the direction information C in the positive rule is still used in the obtained direction rule. The case indicates that the negative rule set has "OPPOSITE C" direction. If C is "RIGHT," then "OPPOSITE C" means "LEFT." Otherwise, if C is "LEFT," then "OPPOSITE C" means "RIGHT." Since the direction in the negative rule set is opposite with that in the positive rule set, the direction information in the template is trustworthy.

(3.3) case indicates that the negative rule set does not have any rule concerning A and B. The obtained direction rule is same with the original template in the positive rule set. The examples of learned direction rules are shown in Table 2. For an interaction verb A, the relations not learned in the training data can appear in the test data. So, we add a default rule ⟨A, otherwise, ANY⟩ as described in Table 2. The default rule permits any direction is okay for other relations not appearing in the training data. Because the training data is so small, the default rule can resolve data sparseness problem.

### 3.4. Applying our proposed method to test data

The procedure to detect gene interactions in the test data is as follows. We detect agent candidates from the test set using the gene dictionary provided by LLL05. Starting from an agent candidate node, we extend all possible syntactic paths. The obtained syntactic encapsulation categories, interaction verbs, and direction rules through three phases are applied to test data according to the following procedure.

For each syntactic chain, we repeat the following procedure.

(1) If a current node is a gene and syntactic chain contains any interaction verb, then we determine that the current node is a target, and stop the extension of the syntactic chain.
(2) Otherwise, if the category of the syntactic relation of the next node candidate is a syntactic encapsulation

Table 2: Examples of direction rules learned through the third phase (based on MINIPAR).

| Lexical word | Relation | Direction | Lexical word | Relation | Direction |
|---|---|---|---|---|---|
| activate | aux | RIGHT | affect | aux | LEFT |
| activate | otherwise | ANY | affect | *i* | RIGHT |
| bind | conj | LEFT | affect | obj | ANY |
| bind | *i* | RIGHT | affect | otherwise | ANY |
| ... | ... | ... | drive | obj | LEFT |
| bind | otherwise | ANY | drive | *s* | RIGHT |

Table 3: Performances of our system and other previous systems using LLL05 syntactic tags.

| Performance on test data(%) | | Hakenberg et al. [11] | Goadrich et al. [14] | Riedel and Klein [17] | Popelinsky and Blatak [15] | Katrenko et al. [16] | Our system (Using LLL05 tags) |
|---|---|---|---|---|---|---|---|
| Using LLL05 syntactic tags | Precision | 28.1 | 28.3 | 60.9 | 46.5 | 39.2 | 67.9 |
| | Recall | 31.4 | 79.6 | 46.2 | 50.0 | 26.5 | 66.6 |
| | F-measure | 29.6 | 41.7 | 52.6 | 48.2 | 31.6 | 67.2 |

Table 4: Performances of our system and other previous systems using MINI-PAR.

| Performance on test data(%) | | Greenwood et al. [12] | Our system (Using MINIPAR) |
|---|---|---|---|
| Using MINIPAR | Precision | 22.2 | 32.4 |
| | Recall | 11.1 | 68.5 |
| | F-measure | 14.8 | 44.0 |

Table 5: Change in performance when one phase is removed.

| | Performance on the test data based on LLL05 syntactic relations(%) | |
|---|---|---|
| Using all phases | Precision | 67.9 |
| | Recall | 66.6 |
| | F-measure | 67.2 |
| Without the first phase (there is no "syntactic encapsulation categories") | Precision | 0 |
| | Recall | 0 |
| | F-measure | 0 |
| Without the second phase (all verbs are considered "interaction verbs") | Precision | 24.6 |
| | Recall | 88.8 |
| | F-measure | 38.5 |
| Without the third phase (there is no syntactic direction information) | Precision | 39.7 |
| | Recall | 72.2 |
| | F-measure | 51.3 |

category, we extend the syntactic chain by adding the next node candidate.

(3) Otherwise, if the current lexical word is an interaction verb and the direction of the next node candidate is consistent with the direction rules, then we extend the syntactic chain.

In the finally obtained syntactic chains, we determine that the first node is an agent and the last node is its target.

## 4. EXPERIMENTAL EVALUATION

### 4.1. Performance of our three-phase method versus those of other methods

With more and more biomedical datasets becoming publicly available, there has been some research effort on corpus design issues and usage in biomedical natural language processing [22, 23]. For a reasonable comparison with previous methods, we applied the training and test data from the LLL05 challenge task. As mentioned before, the LLL05 training dataset without coreference consists of 55 sentences, including 106 genic interactions, and the test data consist of 144 sentences.

Our experiment focused on the following three points.

(1) Based on the LLL05 syntactic tags, the performance of our three-phase method versus that of previous methods.

(2) Based on a real-syntactic analyzer, the performance of our three-phase method versus that of previous methods.

(3) The change in performance when each phase is removed.

In the experiments, we obtained the following five results.

(1) Our three-phase detection method for gene interactions achieved an F-measure of 67.2% using LLL05-annotated syntactic relations, and 44.0% using a real-syntactic analyzer (see Tables 3 and 4).

(2) Using LLL05 syntactic tags, our three-phase method achieved an improvement of 14.6% to 37.6% over previous methods (see Table 3).

(3) Our method significantly outperformed Greenwood et al. [12], which also used MINIPAR (see Table 4).

(4) When the second or third phase was removed, the precision became significantly worse (see Table 5).
(5) When the first phase was removed, there were no interaction results. It means the first phase is important for the improvement of recall (see Table 5).

As shown in Table 3, of the systems evaluated, our system performed the best with a precision of 67.9%, recall of 66.6%, and an F-measure of 67.2 percent.

### 4.2. Discussion of results

We will summarize the significance of each phase introduced in Section 3. As shown in Table 5, every phase is important for its performance. Without the first phase, if no syntactic relations are considered encapsulation categories, then no pairs of genes are generated. Only this result shows the decrease of recall among three results in Table 5. It demonstrates that the syntactic encapsulation categories contribute to the improvement of recall.

Without the second phase, if all the verbs are considered interaction verbs, the precision is very low, which results from the generation of too many wrong syntactic paths. Without the third phase, if we do not consider direction information, then the recall increases and the precision significantly decreases, which also result from the construction of many wrong syntactic paths.

The experiments prove that the second and third phases contribute to the improvement of precision, and the first phase to the improvement of recall. We conclude that all three phases are important for detecting gene interactions.

To experiment the robustness of our method in real time, we have used MINIPAR, an existing syntactic analyzer. The system based on annotated syntactic relations in LLL05 significantly outperforms that using MINIPAR. This is because of the errors in syntactic relations and POS-tags that MINIPAR produced.

## 5. CONCLUSION

To improve recall without sacrificing precision, this paper proposes a three-phase method for the automatic detection of gene interactions using syntactic relations. The proposed method does not require domain knowledge. To improve recall, in the first phase, we construct syntactic encapsulation categories of agent and target. In the second phase, we construct interaction verbs that connect pairs of genes that interact with each other. To improve precision, in the third phase, we learn direction information to detect which of the two genes is the agent or target. The experimental results show that our three-phase method performs significantly better than previous methods. Our method achieved a precision of 67.9%, a recall of 66.6%, an F-measure of 67.2% using LLL05 syntactic relations. We conclude that our proposed three-phase method is effective for detecting gene interactions. Furthermore, we demonstrated that every phase is important for performance.

In the future, we need to expand the size of the training dataset and experiment with a large dataset.

## ACKNOWLEDGMENT

This work was supported by the Sungshin Women's University research grant of 2007.

## REFERENCES

[1] P. Uetz and R. L. Finley Jr., "From protein networks to biological systems," *FEBS Letters*, vol. 579, no. 8, pp. 1821–1827, 2005.
[2] M. Huang, X. Zhu, Y. Hao, D. G. Payan, K. Qu, and M. Li, "Discovering patterns to extract protein-protein interactions from full texts," *Bioinformatics*, vol. 20, no. 18, pp. 3604–3612, 2004.
[3] N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo, "Extracting human protein interactions from MEDLINE using a full-sentence parser," *Bioinformatics*, vol. 20, no. 5, pp. 604–611, 2004.
[4] B. J. Stapley, L. A. Kelley, and M. J. Sternberg, "Predicting the sub-cellular location of proteins from text using support vector machines," in *Proceedings of the 7th Pacific Symposium on Biocomputing*, pp. 374–385, Lihue, Hawaii, USA, January 2002.
[5] B. Rosario and M. Hearst, "Classifying semantic relations in bioscience texts," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04)*, pp. 430–437, Barcelona, Spain, July 2004.
[6] J. Xiao, J. Su, G. Zhou, and C. Tan, "Protein-protein interaction extraction: a supervised learning approach," in *Proceedings of the 1st Symposium on Semantic Mining in Biomedicine (SMBM '05)*, pp. 51–59, Hinxton, Cambridgeshire, UK, April 2005.
[7] J. Saric, L. Jensen, R. Ouzounova, I. Rojas, and P. Bork, "Large-scale extraction of protein/gene relations for model organisms," in *Proceedings of the Symposiumon SemanticMining in Biomedicine*, p. 50, Hinxton, Cambridgeshire, UK, April 2005.
[8] D. Otasek, K. Brown, and I. Jurisica, "Confirming protein-protein interactions by text mining," in *Proceedings of the 6th SIAM Conference on Text Mining*, Bethesda, Md, USA, April 2006.
[9] J. C. Park, H. S. Kim, and J. J. Kim, "Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar," in *Proceedings of the 6th Pacific Symposium on Biocomputing (PSB '01)*, pp. 396–407, Mauna Lani, Hawaii, USA, January 2001.
[10] C. Blaschke, M. A. Andrade, C. Ouzounis, and A. Valencia, "Automatic extraction of biological information from scientific text: protein-protein interactions," in *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB '99)*, pp. 60–67, Heidelberg, Germany, August 1999.
[11] J. Hakenberg, C. Plake, U. Leser, H. Kirsch, and D. R. Schuhmann, "LLL05 challenge: genic interaction extraction-identification of language patterns based on alignment and finite state automata," in *Proceedings of the ICML05 Workshop on Learning Language in Logic (LLL '05)*, pp. 38–45, Bonn, Germany, August 2005.
[12] M. A. Greenwood, M. Stevenson, Y. Guo, H. Harkema, and A. Roberts, "Automatically acquiring a linguistically motivated genic interaction extraction system," in *Proceedings of the ICML05 Workshop on Learning Language in Logic (LLL '05)*, Bonn, Germany, August 2005.

[13] D. Lin, "Dependency-based evaluation of MINIPAR," in *Proceedings of the Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May 1998.

[14] M. Goadrich, L. Oliphant, and J. Shavlik, "Learning to extract genic interactions using Gleaner," in *Proceedings of the ICML05 Workshop on Learning Language in Logic (LLL05)*, Bonn, Germany, August 2005.

[15] L. Popelinsky and J. Blatak, "Learning genic interactions without expert domain knowledge: comparison of different ILP algorithms," in *Proceedings of the ICML05 Workshop on Learning Language in Logic (LLL '05)*, Bonn, Germany, August 2005.

[16] S. Katrenko, M. S. Marshall, M. Roos, and P. Adriaans, "Learning biological interactions from Medline abstracts," in *Proceedings of ICML05 Workshop on Learning Language in Logic (LLL '05)*, Bonn, Germany, August 2005.

[17] S. Riedel and E. Klein, "Genic interaction extraction with semantic and syntactic chains," in *Proceedings of the ICML05 Workshop on Learning Language in Logic (LLL '05)*, Bonn, Germany, August 2005.

[18] F. Rinaldi, G. Schneider, K. Kaljurand, et al., "Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach," *Artificial Intelligence in Medicine*, vol. 39, no. 2, pp. 127–136, 2007.

[19] S. Zhao and R. Grishman, "Extracting relations with integrated information using kernel methods," in *Proceedings of the Association for Computational Linguistics*, pp. 419–426, Ann Arbor, Mich, USA, June 2005.

[20] M. Zhang, J. Zhang, J. Su, and G. Zhou, "A composite kernel to extract relations between entities with both flat and structured features," in *Proceedings of the Computational Linguistics and Association for Computational Linguistics (COLING-ACL '06)*, Sydney, Australia, July 2006.

[21] J. Jiang and C. Zhai, "A systematic exploration of the feature space for relation extraction," in *Proceedings of Human Language Technologies: The North American Chapter of the Association for Computational Linguistics (NAACLHLT '07)*, pp. 113–120, Rochester, NY, USA, April 2007.

[22] K. B. Cohen, L. Fox, P. V. Ogren, and L. Hunter, "Corpus design for biomedical natural language processing," in *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases*, pp. 38–45, Detroit, Mich, USA, June 2005.

[23] K. B. Cohen, L. Fox, P. V. Ogren, and L. Hunter, "Empirical data on corpus design and usage in biomedical natural language processing," in *Proceedings of the American Medical Informatics Association (AMIA '05)*, pp. 156–160, Washington, DC, USA, November 2005.