Parsing Natural Language Queries for Life Science Knowledge

Tadayoshi Hara
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, JAPAN
harasan@nii.ac.jp

Yuka Tateisi
Faculty of Informatics, Kogakuin University
1-24-2 Nishi-shinjuku, Shinjuku-ku,
Tokyo 163-8677, JAPAN
yucca@cc.kogakuin.ac.jp

Jin-Dong Kim
Database Center for Life Science
2-11-16 Yayoi, Bunkyo-ku,
Tokyo 113-0032, JAPAN
jdkim@dbcls.rois.ac.jp

Yusuke Miyao
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, JAPAN
yusuke@nii.ac.jp

Abstract

This paper presents our preliminary work on adaptation of parsing technology toward natural language query processing for biomedical domain. We built a small treebank of natural language queries, and tested a state-of-the-art parser, the results of which revealed that a parser trained on Wall-Street-Journal articles and Medline abstracts did not work well on query sentences. We then experimented an adaptive learning technique, to seek the chance to improve the parsing performance on query sentences. Despite the small scale of the experiments, the results are encouraging, enlightening the direction for effective improvement.

1 Introduction

Recent rapid progress of life science resulted in a greatly increased amount of life science knowledge, e.g. genomics, proteomics, pathology, therapeutics, diagnostics, etc. The knowledge is however scattered in pieces in diverse forms over a large number of databases (DBs), e.g. PubMed, Drugs.com, Therapy database, etc. As more and more knowledge is discovered and accumulated in DBs, the need for their integration is growing, and corresponding efforts are emerging (BioMoby[1], BioRDF[2], etc.).

Meanwhile, the need for a query language with high expressive power is also growing, to cope with the complexity of accumulated knowledge. For example, SPARQL[3] is becoming an important query language, as RDF[4] is recognized as a standard interoperable encoding of information in databases. SPARQL queries are however not easy for human users to compose, due to its complex vocabulary, syntax and semantics. We propose natural language (NL) query as a potential solution to the problem. Natural language, e.g. English, is the most straightforward language for human beings. Extra training is not required for it, yet the expressive power is very high. If NL queries can be automatically translated into SPARQL queries, human users can access their desired knowledge without learning the complex query language of SPARQL.

This paper presents our preliminary work for NL query processing, with focus on syntactic parsing. We first build a small treebank of natural language queries, which are from Genomics track (Hersh et al., 2004; Hersh et al., 2005; Hersh et al., 2006; Hersh et al., 2007) topics (Section 2 and 3). The small treebank is then used to test the performance of a state-of-the-art parser, Enju (Ninomiya et al., 2007; Hara et al., 2007) (Section 4). The results show that a parser trained on Wall-Street-Journal (WSJ) articles and Medline abstracts will not work well on query sentences. Next, we experiment an adaptive learning technique, to seek the chance to improve the parsing performance on query sentences. Despite the small scale of the experiments, the results enlighten directions for effective

[1] http://www.biomoby.org/
[2] http://esw.w3.org/HCLSIG_BioRDF_Subgroup
[3] http://www.w3.org/TR/rdf-sparql-query/
[4] http://www.w3.org/RDF/

Proceedings of the 2011 Workshop on Biomedical Natural Language Processing, ACL-HLT 2011, pages 164–173,
Portland, Oregon, USA, June 23-24, 2011.

	GTREC			
	04	05	06	07
Declarative	1	0	0	0
Imperative	22	60	0	0
Infinitive	1	0	0	0
Interrogative				
- WP/WRB/WDT	3 / 1 / 11	0 / 0 / 0	6 / 22 / 0	0 / 0 / 50
- Non-*wh*	5	0	0	0
NP	14	0	0	0
Total	58	60	28	50

Table 1: Distribution of sentence constructions

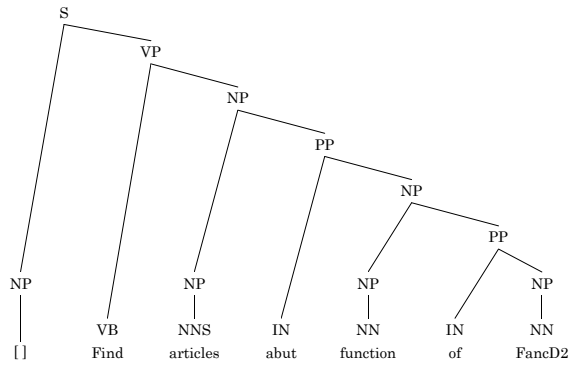

Figure 1: The tree structure for an imperative sentence

improvement (Section 5).

2 Syntactic Features of Query Sentences

While it is reported that the state-of-art NLP technology shows reasonable performance for IR or IE applications (Ohta et al., 2006), NLP technology has long been developed mostly for declarative sentences. On the other hand, NL queries include wide variety of sentence constructions such as interrogative sentences, imperative sentences, and noun phrases. Table 1 shows the distribution of the constructions of the 196 query sentences from the topics of the ad hoc task of Genomics track 2004 (GTREC04) and 2005 (GTREC05) in their narrative forms, and the queries for the passage retrieval task of Genomics track 2006 (GTREC06) and 2007 (GTREC07).

GTREC04 set has a variety of sentence constructions, including noun phrases and infinitives, which are not usually considered as full sentences. In the 2004 track, the queries were derived from interviews eliciting information needs of real biologists, without any control on the sentence constructions.

GTREC05 consists only of imperative sentences. In the 2005 track, a set of templates were derived from an analysis of the 2004 track and other known biologist information needs. The derived templates were used as the commands to find articles describing biological interests such as methods or roles of genes. Although the templates were in the form "Find articles describing ...", actual obtained imperatives begin with "Describe the procedure or method for" (12 sentences), "Provide information about" (36 sentences) or "Provide information on" (12 sentences).

GTREC06 consists only of *wh*-questions where a *wh*-word constitutes a noun phrase by itself (i.e. its part-of-speech is the WP in Penn Treebank (Marcus et al., 1994) POS tag set) or is an adverb (WRB). In the 2006 track, the templates for the 2005 track were reformulated into the constructions of questions and were then utilized for deriving the questions. For example, the templates to find articles describing the role of a gene involved in a given disease is reformulated into the question "What is the role of gene in disease?"

GTREC07 consists only of *wh*-questions where a *wh*-word serves as a pre-nominal modifier (WDT). In the 2007 track, unlike in those of last two years, questions were not categorized by the templates, but were based on biologists' information needs where the answers were lists of named entities of a given type. The obtained questions begin with "what + *entity type*" (45 sentences), "which + *entity type*" (4 sentences), or "In what + *entity type*" (1 sentence).

In contrast, the GENIA Treebank Corpus (Tateisi et al., 2005)[5] is estimated to have no imperative sentences and only seven interrogative sentences (see Section 5.2.2). Thus, the sentence constructions in GTREC04–07 are very different from those in the GENIA treebank.

3 Treebanking GTREC query sentences

We built a treebank (with POS) on 196 query sentences following the guidelines of the GENIA Treebank (Tateisi and Tsujii, 2006). The queries were first parsed using the Stanford Parser (Klein and Manning, 2003), and manual correction was made

[5] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi?page=GENIA+Treebank

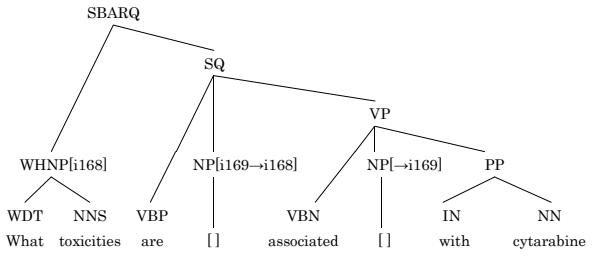

Figure 2: The tree structure for an interrogative sentence

by the second author. We tried to follow the guideline of the GENIA Treebank as closely as possible, but for the constructions that are rare in GENIA, we used the ATIS corpus in Penn Treebank (Bies et al., 1995), which is also a collection of query sentences, for reference.

Figure 1 shows the tree for an imperative sentence. A leaf node with [ ] corresponds to a null constituent. Figure 2 shows the tree for an interrogative sentence. Coindexing is represented by assigning an ID to a node and a reference to the ID to the node which is coindexed. In Figure 2, WHNP[i168] means that the WHNP node is indexed as i168, NP[i169→i168] means that the NP node is indexed as i169 and coindexed to the i168 node, and NP[→i169] means that the node is coindexed to the i169 node. In this sentence, which is a passive *wh*-question, it is assumed that the logical object (*what toxicities*) of the verb (*associate*) is moved to the subject position (the place of i169) and then moved to the sentence-initial position (the place of i168).

As most of the query sentences are either imperative or interrogative, there are more null constituents compared to the GENIA Corpus. In the GTREC query treebank, 184 / 196 (93.9%) sentences contained one or more null constituents, whereas in GENIA, 12,222 / 18,541 (65.9%) sentences did. We expected there are more sentences with multiple null constituents in GTREC compared to GENIA, due to the frequency of passive interrogative sentences, but on the contrary the number of sentences containing more than one null constituents are 65 (33.1%) in GTREC, and 6,367 (34.5%) in GENIA. This may be due to the frequency of relative clauses in GENIA.

4 Parsing system and extraction of imperative and question sentences

We introduce the parser and the POS tagger whose performances are examined, and the extraction of imperative or question sentences from GTREC treebank on which the performances are measured.

4.1 HPSG parser

The Enju parser (Ninomiya et al., 2007)[6] is a deep parser based on the HPSG formalism. It produces an analysis of a sentence that includes the syntactic structure (i.e., parse tree) and the semantic structure represented as a set of predicate-argument dependencies. The grammar is based on the standard HPSG analysis of English (Pollard and Sag, 1994). The parser finds a best parse tree scored by a maxent disambiguation model using a Cocke-Kasami-Younger (CKY) style algorithm.

We used a toolkit distributed with the Enju parser for training the parser with a Penn Treebank style (PTB-style) treebank. The toolkit initially converts the PTB-style treebank into an HPSG treebank and then trains the parser on it. We used a toolkit distributed with the Enju parser for extracting a HPSG lexicon from a PTB-style treebank. The toolkit initially converts the PTB-style treebank into an HPSG treebank and then extracts the lexicon from it.

The HPSG treebank converted from the test section was used as the gold-standard in the evaluation. As the evaluation metrics of the Enju parser, we used labeled and unlabeled precision/recall/F-score of the predicate-argument dependencies produced by the parser. A predicate-argument dependency is represented as a tuple of $\langle w_p, w_a, r \rangle$, where w_p is the predicate word, w_a is the argument word, and r is the label of the predicate-argument relation, such as `verb-ARG1` (semantic subject of a verb) and `prep-ARG1` (modifiee of a prepositional phrase).

4.2 POS tagger

The Enju parser assumes that the input is already POS-tagged. We use a tagger in (Tsuruoka et al., 2005). It has been shown to give a state-of-the-art accuracy on the standard Penn WSJ data set and also on a different text genre (biomedical literature) when trained on the combined data set of the WSJ data and

[6] http://www-tsujii.is.s.u-tokyo.ac.jp/enju

the target genre (Tsuruoka et al., 2005). Since our target is biomedical domain, we utilize the tagger adapted to the domain as a baseline, which we call "the GENIA tagger".

4.3 Extracting imperative and question sentences from GTREC treebank

In GTREC sentences, two major constructions of sentences can be observed: imperative and question sentences. These two types of sentences have different sentence constructions and we will observe the impact of each or both of these constructions on the performances of parsing or POS-tagging. In order to do so, we collected imperative and question sentences from our GTREC treebank as follows:

- **GTREC imperatives** - Most of the imperative sentences in GTREC treebank begin with empty subjects "(NP-SBJ */-NONE-)". We extracted such 82 imperative sentences.
- **GTREC questions** - Interrogative sentences are annotated with the phrase label "SBARQ" or "SQ", where "SBARQ" and "SQ" respectively denote a *wh*-question and an yes/no question. We extracted 98 interrogative sentences whose top phrase labels were either of them.

5 Experiments

We examine the POS-tagger and the parser for the sentences in the GTREC corpus. They are adapted to each of GTREC overall, imperatives, and questions. We then observe how the parsing or POS-tagging accuracies are improved and analyze what is critical for parsing query sentences.

5.1 Experimental settings

5.1.1 Dividing corpora

We prepared experimental datasets for the following four domains:

- **GENIA Corpus (GENIA) (18,541 sentences)**
  Divided into three parts for training (14,849 sentences), development test (1,850 sentences), and final test (1,842 sentences).
- **GTREC overall (196 sentences)**
  Divided into two parts: one for ten-folds cross validation test (17-18 $\times$ 10 sentences) and the other for error analysis (17 sentences)

Target	GENIA tagger	Adapted tagger
GENIA	99.04%	-
GTREC (overall)	89.98%	96.54%
GTREC (imperatives)	90.32%	97.30%
GRREC (questions)	89.25%	94.77%

Table 2: Accuracy of the POS tagger for each domain

- **GTREC imperatives (82 sentences)**
  Divided into two parts: one for ten-folds cross validation test (7-8 $\times$ 10 sentences) and the other for error analysis (7 sentences)
- **GTREC questions (98 sentences)**
  Divided into two parts: one for ten-folds cross validation test (9 $\times$ 10 sentences) and the other for error analysis (8 sentences)

5.1.2 Adaptation of POS tagger and parser

In order to adapt the POS tagger and the parser to a target domain, we took the following methods.

- **POS tagger** - For the GTREC overall / imperatives / questions, we replicated the training data for 100,000 times and utilized the concatenated replicas and GENIA training data in (Tsuruoka et al., 2005) for training. For POS tagger, the number of replicas of training data was determined among $10^n (n = 0, \ldots, 5)$ by testing these numbers on development test sets in three of ten datasets of cross validation.
- **Enju parser** - We used a toolkit in the Enju parser (Hara et al., 2007). As a baseline model, we utilized the model adapted to the GENIA Corpus. We then attempted to further adapt the model to each domain. In this paper, the baseline model is called "the GENIA parser".

5.2 POS tagger and parser performances

Table 2 and 3 respectively show the POS tagging and the parsing accuracies for the target domains, and Figure 3 and 4 respectively show the POS tagging and the parsing accuracies for the target domains given by changing the size of the target training data.

The POS tagger could output for each word either of one-best POS or POS candidates with probabilities, and the Enju parser could take either of the two output types. The bracketed numbers in Table 3 and

Parser	GENIA			Adapted		
POS	Gold	GENIA tagger	Adapted tagger	Gold	GENIA tagger	Adapted tagger
For GENIA	88.54	88.07 (88.00)	-	-	-	-
For GTREC overall	84.37	76.81 (72.43)	83.46 (81.96)	89.00	76.98 (74.44)	86.98 (85.42)
For GTREC imperatives	85.19	78.54 (77.75)	85.71 (85.48)	89.42	74.40 (74.84)	88.97 (88.67)
For GTREC questions	85.45	76.25 (67.27)	83.55 (80.46)	87.33	81.41 (71.90)	84.87 (82.70)

[using POS candidates with probabilities (using only one best POS)]

Table 3: Accuracy of the Enju parser for GTREC

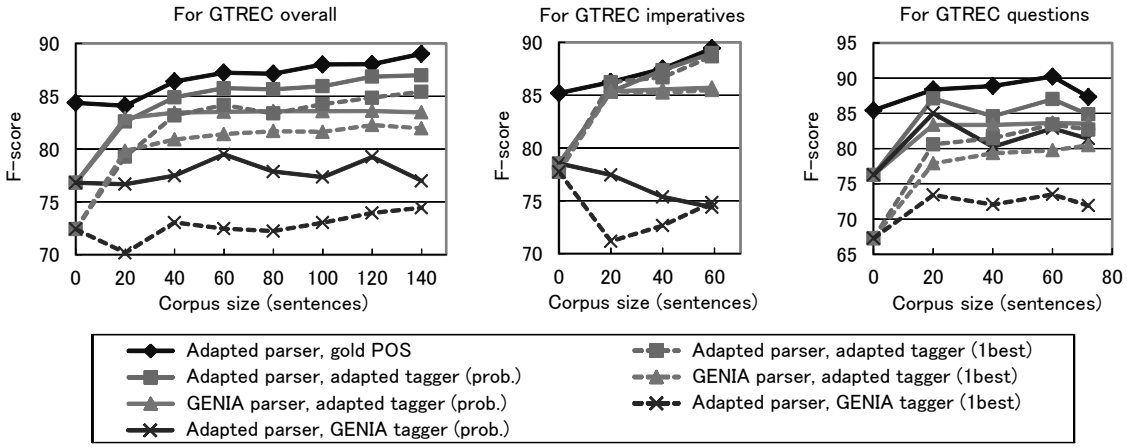

Figure 4: Parsing accuracy vs. corpus size

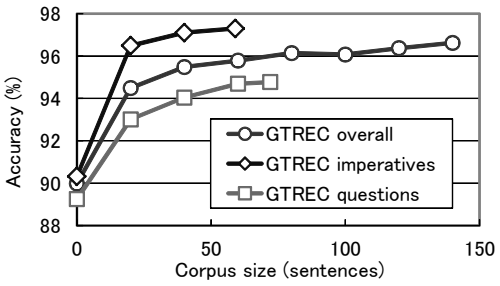

Figure 3: POS tagging accuracy vs. corpus size

Correct → Error	GENIA tagger	Adapted tagger
For GTREC overall (17 sentences)		
NN → NNP	4	0.6
VB → NN	4	0
WDT → WP	4	0
NN → JJ	1	1.9
For GTREC imperative (seven sentences)		
FW → NNP / NN / JJ	7	4
VB → NN	4	0
NN → NNP	2	0
For GTREC question (eight sentences)		
WDT → WP	3	0
VB → VBP	2	1
NNS → VBZ	2	0

(The table shows only error types observed more than once for either of the taggers)

Table 4: Tagging errors for each of the GTREC corpora

the dashed lines in Figure 4 show the parsing accuracies when we utilized one-best POS given by the POS tagger, and the other numbers and lines show the accuracies given by POS candidates with probabilities. In the rest of this section, when we just say "POS tagger", the tagger's output is POS candidates with probabilities.

Table 4 and 5 respectively compare the types of POS tagging and parsing errors for each domain between before and after adapting the POS tagger, and Table 6 compares the types of parsing errors for each domain between before and after adapting the parser. The numbers of errors for the rightmost column in each of the tables were given by the average of the ten-folds cross validation results.

In the following sections, we examine the impact of the performances of the POS taggers or the parsers on parsing the GTREC documents.

Error types	GENIA parser	
	GENIA tagger	Adapted tagger
For GTREC overall (17 sentences)		
Failure in detecting verb	12	0.2
Root selection	6	0
Range of NP	5	5
PP-attachment	4	3
Determiner / pronoun	4	1
Range of verb subject	4	4
Range of verb object	3	3
Adjective / modifier noun	2	3
For GTREC imperatives (seven sentences)		
Failure in detecting verb	8	0
Root selection	4	0
Range of NP	3	4
PP-attachment	3	1.8
Range of PP	2	2
For GTREC questions (eight sentences)		
Range of coordination	5	3
Determiner / pronoun	3	0
PP-attachment	3	1
Range of PP	2	2
Subject for verb	2	1

(The table shows only the types of parsing errors observed more than once for either of the parsers)

Table 5: Impact of adapting POS tagger on parsing errors

5.2.1 Impact of POS tagger on parsing

In Table 2, for each of the GTREC corpora, the GENIA tagger dropped its tagging accuracy by around nine points, and then recovered five to seven points by the adaptation. According to this behavior of the tagger, Table 3 shows that the GENIA and the adapted parsers with the GENIA tagger dropped their parsing accuracies by 6–15 points in F-score from the accuracies with the gold POS, and then recovered the accuracies within two points below the accuracies with the gold POS. The performance of the POS tagger would thus critically affect the parsing accuracies.

In Figure 3, we can observe that the POS tagging accuracy for each corpus rapidly increased only for first 20–30 sentences, and after that the improvement speed drastically declined. Accordingly, in Figure 4, the line for the adapted parser with the adapted tagger (the line with triangle plots) rose rapidly for the first 20–30 sentences, and after that slowed down.

We explored the tagging and parsing errors, and analyze the cause of the initial accuracy jump and the successive improvement depression.

Error types	Gold POS	
	GENIA parser	Adapted parser
For GTREC overall (17 sentences)		
Range of NP	5	1.3
Range of verb subject	3	2.6
PP-attachment	3	2.7
Whether verb takes object & complement	3	2.9
Range of verb object	2	1
For GTREC imperatives (seven sentences)		
Range of NP	4	1.1
PP-attachment	2	1.6
Range of PP	2	0.3
Preposition / modifier	2	2
For GTREC questions (eight sentences)		
Coordination / conjunction	2	2.2
Auxiliary / normal verb	2	2.6
Failure in detecting verb	2	2.6

(The table shows only the types of parsing errors observed more than once for either of the parsers)

Table 6: Impact of adapting parser on parsing errors

Cause of initial accuracy jump

In Table 4, "VB → NN" tagging errors were observed only in imperative sentences and drastically decreased by the adaptation. In a imperative sentence, a verb (VB) usually appears as the first word. On the other hand, the GENIA tagger was trained mainly on the declarative sentences and therefore would often take the first word in a sentence as the subject of the sentence, that is, noun (NN). When the parser received a wrong NN-tag for a verb, the parser would attempt to believe the information ("failure in detecting verb" in Table 6) and could then hardly choose the NN-tagged word as a main verb ("root selection" in Table 6). By adapting the tagger, the correct tag was given to the verb and the parser could choose the verb as a main verb.

"WDT → WP" tagging errors were observed only in the question sentences and also drastically decreased. For example, in the sentence "What toxicities are associated with cytarabine?", "What" works as a determiner (WDT) which takes "toxicities", while the GENIA tagger often took this "What" as a pronoun (WP) making a phrase by itself. This would be because the training data for the GENIA tagger would contain 682 WP "what" and only 27 WDT "what". WP "what" could not make a noun phrase by taking a next noun, and then the parsing of the parsing would corrupt ("determiner / pronoun" in Table 5). By adapting the tagger, "WDT" tag was

given to "What", and the parser correctly made a phrase "What toxicities".

Since the variation of main verbs in GTREC imperatives is very small (see Section 2) and that of interrogatives is also very small, in order to correct the above two types of errors, we would require only small training data. In addition, these types of errors widely occurred among imperatives or questions, the accuracy improvement by correcting the errors was very large. The initial rapid improvement would thus occur.

Cause of improvement depression

"NN → NNP" tagging errors would come from the description style of words. In the GTREC queries, technical terms, such as the names of diseases or proteins, sometimes begin with capital characters. The GENIA tagger would take the capitalized words not as a normal noun (NN) but as a proper noun (NNP). By adaptation, the tagger would have learned the capital usage for terms and the errors then decreased.

However, in order to achieve such improvement, we would have to wait until a target capitalized term is added to the training corpus. "FW → NNP / NN / JJ", "NN → JJ", and several other errors would be similar to this type of errors in the point that, they would be caused by the difference in annotation policy or description style between the training data for the GENIA tagger and the GTREC queries.

"VB → VBP" errors were found in questions. For example, "affect" in the question "How do mutations in Sonic Hedgehog genes affect developmental disorders?" was base form (VB), while the GENIA tagger took it as a present tense (VBP) since the GENIA tagger would be unfamiliar with such verb behavior in questions. By adaptation, the tagger would learn that verbs in the domain tend to take base forms and the errors then decreased.

However, the tagger model based on local context features could not substantially solve the problem. VBP of course could appear in question sentences. We observed that a verb to be VBP was tagged with VB by the adapted tagger. In order to distinguish VB from VBP, we should capture longer distance dependencies between auxiliary and main verbs.

In tagging, the fact that the above two types of errors occupied most of the errors other than the errors involved in the initial jump, would be related to why the accuracy improvement got so slowly, which would lead to the improvement depression of the parsing performances. With the POS candidates with probabilities, the possibilities of correct POSs would increase, and therefore the parser would give higher parsing performances than using only one-best POSs (see Table 3 and Figure 4).

Anyway, the problems were not substantially solved. For these tagging problems, just adding the training data would not work. We might need reconstruct the tagging system or re-consider the feature designs of the model.

5.2.2 Impact of parser itself on parsing

For the GTREC corpora, the GENIA parser with gold POSs lowered the parsing accuracy by more than three points than for the GENIA Corpus, while the adaptation of the parser recovered a few points for each domain (second and fifth column in Table 3). Figure 4 would also show that we could improve the parser's performance with more training data for each domain. For GTREC questions, the parsing accuracy dropped given the maximum size of the training data. Our training data is small and therefore small irregular might easily make accuracies drop or rise. [7] We might have to prepare more corpora for confirming our observation.

Table 6 would imply that the major errors for all of these three corpora seem not straightforwardly associated with the properties specific to imperative or question sentences. Actually, when we explored the parse results, errors on the sentence constructions specific to the two types of sentences would hardly be observed. ("Failure in detecting verb" errors in GTREC questions came from other causes.) This would mean that the GENIA parser itself has potential to parse the imperative or question sentences.

The training data of the GENIA parser consists of the WSJ Penn Treebank and the GENIA Corpus. As long as we searched with our extraction method in Section 4.3, the WSJ and GENIA Corpus seem respectively contain 115 and 0 imperative, and 432

[7] This time we could not analyze which training data affected the decrease, because through the cross validation experiments each sentence was forced to be once final test data. However, we would like to find the reason for this accuracy decrease in some way.

and seven question sentences. Unlike the POS tagger, the parser could convey more global sentence constructions from these sentences.

Although the GENIA parser might understand the basic constructions of imperative or question sentences, by adaptation of the parser to the GTREC corpora, we could further learn more local construction features specific to GTREC, such as word sequence constructing a noun phrase, attachment preference of prepositions or other modifiers. The error reduction in Table 6 would thus be observed.

However, we also observed that several types of errors were still mostly unsolved after the adaptation. Choosing whether to add complements for verbs or not, and distinguishing coordinations from conjunctions seems to be difficult for the parser. If two question sentences were concatenated by conjunctions into one sentence, the parser would tend to fail to analyze the sentence construction for the latter sentence. The remaining errors in Table 6 would imply that we should also re-consider the model designs or the framework itself for the parser in addition to just increasing the training data.

6 Related work

Since domain adaptation has been an extensive research area in parsing research (Nivre et al., 2007), a lot of ideas have been proposed, including un-/semi-supervised approaches (Roark and Bacchiani, 2003; Blitzer et al., 2006; Steedman et al., 2003; McClosky et al., 2006; Clegg and Shepherd, 2005; McClosky et al., 2010) and supervised approaches (Titov and Henderson, 2006; Hara et al., 2007). Their main focus was on adapting parsing models trained with a specific genre of text (in most cases PTB-WSJ) to other genres of text, such as biomedical research papers. A major problem tackled in such a task setting is the handling of unknown words and domain-specific ways of expressions. However, as we explored, parsing NL queries involves a significantly different problem; even when all words in a sentence are known, the sentence has a very different construction from declarative sentences.

Although sentence constructions have gained little attention, a notable exception is (Judge et al., 2006). They pointed out low accuracy of state-of-the-art parsers on questions, and proposed supervised parser adaptation by manually creating a treebank of questions. The question sentences are annotated with phrase structure trees in the PTB scheme, although function tags and empty categories are omitted. An LFG parser trained on the treebank then achieved a significant improvement in parsing accuracy. (Rimell and Clark, 2008) also worked on question parsing. They collected question sentences from TREC 9-12, and annotated the sentences with POSs and CCG (Steedman, 2000) lexical categories. They reported a significant improvement in CCG parsing without phrase structure annotations.

On the other hand, (Judge et al., 2006) also implied that just increasing the training data would not be enough. We went further from their work, built a small but complete treebank for NL queries, and explored what really occurred in HPSG parsing.

7 Conclusion

In this paper, we explored the problem in parsing queries. We first attempted to build a treebank on queries for biological knowledge and successfully obtained 196 annotated GTREC queries. We next examined the performances of the POS tagger and the HPSG parser on the treebank. In the experiments, we focused on the two dominant sentence constructions in our corpus: imperatives and questions, extracted them from our corpus, and then also examined the parser and tagger for them.

The experimental results showed that the POS tagger's mis-tagging to main verbs in imperatives and *wh*-interrogatives in questions critically decreased the parsing performances, and that our small corpus could drastically decrease such mis-tagging and consequently improve the parsing performances. The experimental results also showed that the parser itself could improve its own performance by increasing the training data. On the other hand, the experimental results suggested that the POS tagger or the parser performance would stagnate just by increasing the training data.

In our future research, on the basis of our findings, we would like both to build more training data for queries and to reconstruct the model or reconsider the feature design for the POS tagger and the parser. We would then incorporate the optimized parser and tagger into NL query processing applications.

References

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II style — Penn Treebank project. Technical report, Department of Linguistics, University of Pennsylvania.

John Blitzer, Ryan Mcdonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia.

A. B. Clegg and A. Shepherd. 2005. Evaluating and integrating treebank parsers on a biomedical corpus. In *Proceedings of the ACL 2005 Workshop on Software*, Ann Arbor, Michigan.

Tadayoshi Hara, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Evaluating impact of re-training a lexical disambiguation model on domain adaptation of an hpsg parser. In *Proceedings of 10th International Conference on Parsing Technologies (IWPT 2007)*, pages 11–22.

William R. Hersh, Ravi Teja Bhupatiraju, L. Ross, Aaron M. Cohen, Dale Kraemer, and Phoebe Johnson. 2004. TREC 2004 Genomics Track Overview. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004*.

William R. Hersh, Aaron M. Cohen, Jianji Yang, Ravi Teja Bhupatiraju, Phoebe M. Roberts, and Marti A. Hearst. 2005. TREC 2005 Genomics Track Overview. In *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005*.

William R. Hersh, Aaron M. Cohen, Phoebe M. Roberts, and Hari Krishna Rekapalli. 2006. TREC 2006 Genomics Track Overview. In *Proceedings of the Fifteenth Text REtrieval Conference, TREC 2006*.

William R. Hersh, Aaron M. Cohen, Lynn Ruslen, and Phoebe M. Roberts. 2007. TREC 2007 Genomics Track Overview. In *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007*.

John Judge, Aoife Cahill, and Josef van Genabith. 2006. Questionbank: Creating a Corpus of Parsing-Annotated Questions. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 497–504.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert Macintyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of ARPA Human Language Technology Workshop*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 337–344, Sydney, Australia.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic Domain Adaptation for Parsing. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the ACL*, pages 28–36, Los Angeles, California.

Takashi Ninomiya, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2007. A log-linear model with an n-gram reference distribution for accurate hpsg parsing. In *Proceedings of 10th International Conference on Parsing Technologies (IWPT 2007)*, pages 60–68.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic, June. Association for Computational Linguistics.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Laura Rimell and Stephen Clark. 2008. Adapting a Lexicalized-Grammar Parser to Contrasting Domains. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 475–584.

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 126–133, Edmonton, Canada.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 331–338, Budapest, Hungary.

Mark Steedman. 2000. *The Syntactic Process*. THE MIT Press.

Yuka Tateisi and Jun'ichi Tsujii. 2006. GENIA Annotation Guidelines for Treebanking. Technical Report TR-NLP-UT-2006-5, Tsujii Laboratory, University of Tokyo.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. 2005. Syntax Annotation for the GENIA corpus. In *Proceedings of the Second International Joint Conference on Natural Language Process-*

ing (IJCNLP 2005), Companion volume, pages 222–227.

Ivan Titov and James Henderson. 2006. Porting statistical parsers with data-defined kernels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 6–13, New York City.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics*, volume LNCS 3746, pages 382–392, Volos, Greece, November. ISSN 0302-9743.