Part-of-speech Tagging for Web Search Queries Using a Large-scale Web Corpus

Atsushi Keyaki
Tokyo Institute of Technology, 2-12-1 Ookayama
Meguro-ku, Tokyo 152-8550, Japan
keyaki@lsc.cs.titech.ac.jp

Jun Miyazaki
Tokyo Institute of Technology, 2-12-1 Ookayama
Meguro-ku, Tokyo 152-8550, Japan
miyazaki@cs.titech.ac.jp

ABSTRACT

This paper proposes an accurate part-of-speech (POS) tagging method for Web search queries using the sentence level morphological analysis results of a large-scale Web corpus. POS tagging is a fundamental technique for analyzing queries; however, the existing NLP tools often fail to correctly identify POS tags because queries are not based on natural language grammar. We propose a method not affected by the queries' characteristics lacking capitalization and free word order with the term-POS database (*TPDB*). Experimental results show that the proposed method significantly outperforms those using existing NLP tools and the state-of-the-art method. In addition, the data set we created is expected to be useful for future researches on both POS tagging systems to queries and IR systems leveraging POS tags.

CCS Concepts

•Information systems → Query representation; Web indexing; Test collections; •Computing methodologies → Lexical semantics;

Keywords

part-of-speech tagging; Web search query; term-POS database, global statistics

1. INTRODUCTION

Part-of-speech (POS) tagging is one of the most fundamental and important techniques in text processing. POS tagging is not only essential for more advanced natural language processing, but also quite useful in Web search systems. In the Web search system, it is reported that POS information can improve search accuracy or detect unnecessary data in some researches [6, 5], which is greatly beneficial for users. Crestani et al. found that search accuracy improved when search strategy is changed by POS tag of a query term [6]. Barr et al. prevented a drop in the search accuracy by using POS information in selecting important data [5]. These researches showed the potential of using POS information in IR systems. As a more concrete and intuitive example, a proper noun phrase "*discovery channel*" is composed of common nouns "*discovery*" and "*channel*". Common nouns "*discovery*" and "*channel*" can be used individually in a document with the different intent from the TV program "*discovery channel*", which may cause retrieving false positive results. Thus, accurate POS tagging to queries is very important and required because POS tagging accuracy directly affect the search effectiveness of that kind of IR systems.

Furthermore, advanced query analysis, such as semantic tagging [11], query task classification [9], and IR with word sense [17], suppose accurate POS tagging to queries. Hence, accurate POS tagging is a promising technique.

The biggest problem in utilizing POS information for accurate IR is that POS tagging to queries has not attained a level of accuracy sufficient for permitting its practical usage. This is because queries are not based on natural language grammar; this makes it difficult to correctly identify POS tags by using existing NLP tools, namely, morphological analysis trained with natural language documents. More precisely, queries have the following characteristics different from natural language documents: 1) the length is short, 2) capitalization is missing, and 3) word order is fairly free [7]. Accuracy is still low, although morphological analysis accuracy improves when trained with capitalization-labeled queries [2]. We observed the same trend through our preliminary investigations in Section 2 with the data set we constructed.

One of the solutions to this issue is to utilize the results of sentence level morphological analysis [3, 7]. This is because sentences are based on natural language grammar and these morphological analysis results are more reliable. Bendersky et al. [3] and Ganchev et al. [7] used sentences of search results and snippets from search logs, respectively. Specifically, POS tags of query terms are identified with morphological analysis results of these sentences. These researches mainly target the utilization of highly relevant and a small number of sentences. Hence, *global* statistics derived from a large-scale corpus are not fully inspected. Therefore, we investigate the potential of using global statistics derived from a large-scale Web corpus and build the term-POS database

SAC'17, April 3-7, 2017, Marrakesh, Morocco

http://dx.doi.org/10.1145/3019612.3019694

(*TPDB*) in a preprocessing step.

In identifying query POS tags, we propose a method that not only ignores capitalization information, but also avoids reflecting the word order in a scoring function for co-operating queries' characteristics. In other words, we work on solving the problems caused by characteristics 2) capitalization is missing, and 3) word order is fairly free[1].

Experimental evaluations are conducted through two data sets; MS-251[2] which is used in previous studies [3, 7], and the POS-tagged Web track topics of TREC Web Track[3] which is created by the authors. The advantage of the data set we created is that the information need and search background of a query can be referred to, which are considerably useful in annotating reliable oracle POS tags to query terms.

The contributions of this study are as follows:

- We proposed a method for correctly identifying POS tags of queries with *TPDB* containing massive and accurate morphological analysis results derived from a large-scale Web corpus, and
- We confirmed that the data set composed of Web track topics of TREC Web Track is applicable for a task of POS tagging to Web search queries. This data set we constructed helps and encourages the development of IR systems leveraging POS tags.

2. PRELIMINARY INVESTIGATION

We conducted a preliminary investigation to explore under what circumstances morphological analysis tools fail to correctly identify POS tags of queries. We used Stanford Log-linear Part-Of-Speech Tagger[4] [16] as an existing morphological analysis tool. Then, we use the default English model and the caseless English model that do not consider capitalization information during training.

We use Web track topics (200 queries from 2009–2012), which is a query set used in TREC Web Track; it is intended at evaluating the effectiveness of IR systems. Note that we applied a stop-word processing with SMART stop list [15]. In experimental evaluations, we follow the classification of 14 POS tags by Ganchev et al. [7], which adds *proper noun* and *symbol* to the 12 universal POS tags by Petrov et al. [12], i.e., *verb*, *noun*, *pronoun*, *adjective*, *adverb*, *adposition*, *conjunction*, *determiner*, *number*, *particle*, *punctuation*, and *others*.

One of the authors and two employed annotators manually assigned the correct POS tag to each query term[5] using *description*. It describes the information need and search background of a query. The result of morphological analysis of *description* is also referred to. First, each of the three annotators assign POS tags to every query terms individually. After that, they discuss all query terms not agreeing to determine the most appropriate POS tags. Annotators attained high inter-annotator agreement (Fleiss' Kappa is 0.98), which is achieved by rich information about queries, or *description*. Various kinds of statistics, such as the ratio of each POS tag, show the same trend as [2], although capitalization occurs less in the POS-tagged Web track topics.

Table 1: Precision and recall of morphological analysis using the default model

POS	precision	recall
common noun	.550	.985
proper noun	1.0	.010
verb	.722	.867
adjective	.451	.958
all query terms	.547	.547

Table 2: Precision and recall of morphological analysis using the caseless model

POS	precision	recall
common noun	.789	.769
proper noun	.751	.640
verb	.733	.733
adjective	.690	.833
all query terms	.763	.763

2.1 Analysis using the Default Model

Table 1 shows precision and recall of morphological analysis results using the default model by POS tag[6]. Nearly half of the query terms were assigned the correct POS tags, which is much lower than that obtained with sentence level morphological analysis and is far from being practical useful.

The fact that the recall of proper noun is only 1% shows that almost all of the proper nouns are not identified. This is due to the lack of capitalization in Web search queries. From the results of deeper analysis, we found that 72 % of the errors are that proper nouns are mistakenly assigned as common nouns. For instance, a person's name such as "*obama*", name of a place such as "*india*", and facility such as "*ritz carlton*", are all identified as common nouns. Similarly, in some cases, the query is a proper noun but each term in it is a common noun such as "*discovery channel*". The conclusion is that some proper nouns are difficult to identify without context or external resources.

We also observed errors caused by using a partial grammatical rule. For example, "*lower*" in the query "*lower heart rate*" was identified as an adjective, although the correct POS tag is a verb judging from the query's *description*. Then, "*pay*" in the query "*gs pay rate*" ("*gs*" is the abbreviation of general schedule) is issued as a common noun, but identified as a verb. It seems these errors are caused by partial gram-

[1]Regarding the characteristic of 1) the length is short, this characteristic may be solved with users' continuous query logs, however, we target independently issued queries in this study. Thus, handing this characteristic is out of our focus.

[2]`http://code.google.com/archive/p/query-syntax/`

[3]`http://trec.nist.gov/data/webmain.html`

[4]`http://nlp.stanford.edu/software/tagger.shtml`

[5]For wide purpose usage, annotators use a more fined-grained POS tag classification used in Stanford POS tagger. `http://www.lsc.cs.titech.ac.jp/keyaki/dataSet/POSTaggedTREC-WebQuery.tsv`

[6]We omit low-frequency POS tags for simplification.

matical rules like adjectives having a higher probability of appearing before common nouns, and verbs having a higher possibility of appearing after a subject.

2.2 Analysis using the Caseless Model

Experimental results in the case where the caseless model was used are depicted in Table 2. Compared with the case where the default model was used, precision and recall improved overall. Proper nouns were identified using the caseless model, unlike using the default model; however, the accuracy did not reach a level sufficient for practical usage. The percentage of proper nouns being mistakenly identified as common nouns clearly decreased to 31%, whereas the percentage of common nouns being mistakenly identified as proper nouns drastically increased to 36%.

An example of the error is that a common noun term "*store*" is mistakenly identified as a proper noun in a query "*discovery channel store*". Most queries do not contain explicit segmentation between one phrase to another phrase. Thus, the tagger cannot split "*discovery channel*" and "*store*", which indicates that the problem of partial grammatical rules still exist, even though more of proper nouns are identified. Solutions for this issue can be word segmentation [14] or some other method that avoids reflecting the word order in identifying POS tags.

The accuracy of the caseless model is comparable to that of Stanford POS tagger trained with capitalization-labeled queries in a related study [2]. We suppose that both the caseless model and the model trained with capitalization-labeled queries are intended at achieving the same goal considering that both fill the gap of capitalization information between training data and queries. This suggests that the POS tagging difficulty of the POS-tagged Web track topics is the same as that of the data set used in [2] which is composed of search queries issued to Yahoo! search engine. Note that experimental evaluation with a data set used in the existing study [3, 7] is conducted in Section 4.2 for comparison.

3. POS TAGGING USING THE TPDB

3.1 Overview of the Proposed Method

Applying morphological analysis to queries is error prone as the former investigations have shown. Thus, we utilized the morphological analysis results of sentences in a large-scale Web corpus in POS tagging to queries, because the accuracy of sentence level morphological analysis has reached a level sufficient for practical usage. More precisely, we used sentences containing query terms from a large-scale Web corpus to determine the frequently assigned POS tags of query terms in the corpus intending to assign these POS tags to query terms. On this occasion, we consider a method that is not affected by free word order of queries. The process is shown below in more detail:

Building the *TPDB* We applied morphological analysis to each sentence in a large-scale Web corpus and stored the result, a combination of term-POS pairs, in the *TPDB*. This process is conducted offline, thus, it does not affect query processing time.

POS tagging to queries When a query is issued, sentences (precisely, combinations of term-POS pairs) containing two or more query terms are retrieved from the *TPDB*. Then, we identify the appropriate POS tags of the query terms. With regard to a single term query, the most frequently appearing POS tag in a large-scale Web corpus is tagged to the query term.

Note that terms in the *TPDB* and query terms are lowercased for handling missing capitalization of queries, which also helps easily matching of documents and queries.

We illustrate a concrete example of the proposed method with Figure 1. Suppose four sentences S_1, S_2, S_3, S_4 exist in the large-scale Web corpus. First, we apply morphological analysis to the sentences for assigning POS tags. Next, each combination of the term-POS pairs corresponding to a sentence is stored into the *TPDB*. In this case, a query "t_A t_C" is issued and all data containing t_A and t_C, that is, S_1, S_2, S_3 are retrieved. Concerning POS tags of t_A and t_C, t_A/P_1 (the POS tag of t_A is P_1) and t_C/P_3 appears twice, while t_A/P_1 and t_C/P_4 appears once. Eventually, query POS tags are identified as t_A/P_1 and t_C/P_3, because the combination occurs more frequently.

Come to think of queries composed of three or more terms, retrieving data containing all query terms and extracting their POS tags for assigning them to query terms is natural when a number of sentences contain all query terms. However, when such a sentence that contains all query terms hardly exists and specific two query terms co-occur frequently, the POS tags of the two terms worth being emphasized. In the next section, we discuss a method to correctly identify POS tags under the circumstances mentioned above.

3.2 Scoring Function for POS tagging of Queries

Data retrieved from *TPDB* are broken down and classified by the query term pair. This is because when three or more query terms co-occur frequently, arbitrary pairs of these query terms also co-occur frequently. Thus, we comprehensively consider all query term pairs in identifying POS tags. Suppose a query is "t_A t_B t_C" as illustrated in Figure 2. Derived query term pairs are t_A:t_B, t_A:t_C, and t_B:t_C.

Next, we gather the term-POS pairs by each query term pair, and count the frequency of each term-POS pair by POS tag combination. Regarding a query term pair t_A:t_B in Figure 2, the frequency of t_A/P_1 and t_B/P_2 is 5, whereas the normalized frequency is 0.33 ($\frac{5}{5+3+7}$) which is divided by the total frequency of t_A:t_B as indicated in Figure 2.

The important property of this approach is that term-POS pairs in a sentence containing three or more query terms have more impact on scoring. Suppose a sentence composed of t_A/P_1, t_B/P_2, and t_C/P_2 exists. This sentence is broken down into t_A/P_1 and t_B/P_2, t_A/P_1 and t_C/P_2, and t_B/P_2 and t_C/P_2, that is, each of term-POS pairs is counted twice. The pseudo code of the process is described in Algorithm 1.

Hereinafter, we devised three methods to determine POS tags.

MaxFreq In *MaxFreq*, the most frequently assigned POS tag is regarded as an appropriate POS tag. Thus, let

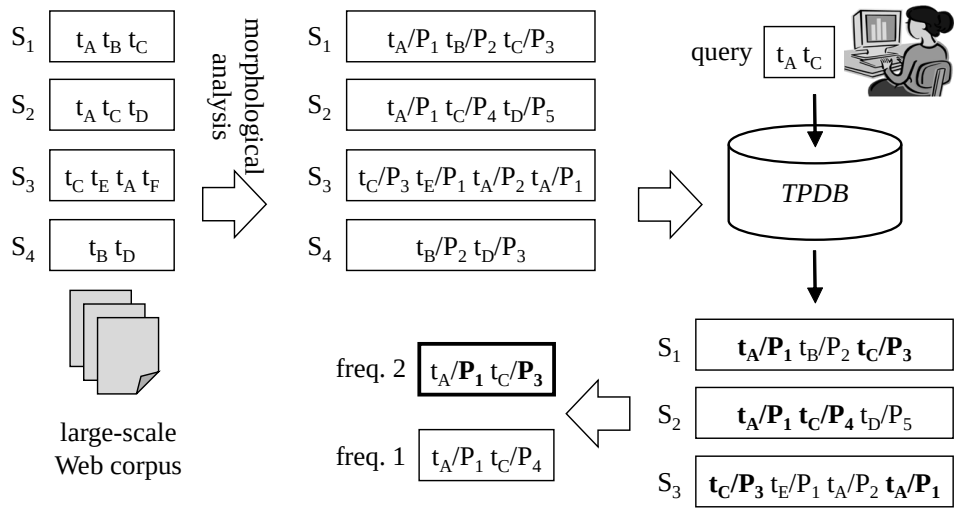

Figure 1: POS tagging with the *TPDB*

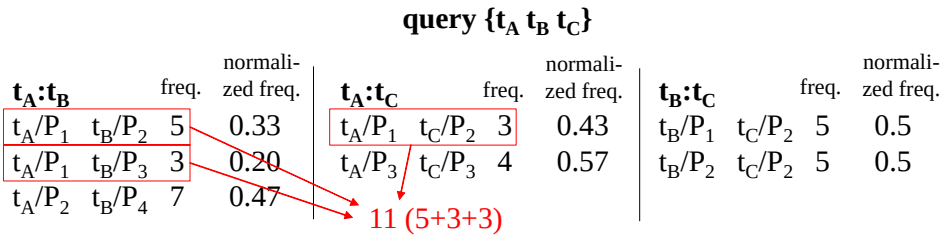

Figure 2: Statistics for identifying POS tags

the determined POS tag be the most frequently appearing POS tag in all term-POS pairs. This is the most straightforward and simplest method of determining POS tags. The POS tag of t_A is P_2 because the frequency of t_A/P_2 and t_B/P_4 is 7 which is the highest. We show the pseudo code of *MaxFreq* in Algorithm 2.

MostLikelihood There is a possibility that a term frequently appearing in a corpus dominate the result with *MaxFreq*. We therefore consider normalization of frequency in *MostLikelihood*. Let the determined POS tag be the POS tag with the highest normalized frequency. The POS tag of t_A is P_3 because the normalized frequency of t_A/P_3 and t_C/P_3 is 0.57 which is the highest. We show the pseudo code of *MostLikelihood* in Algorithm 3.

AllCombi The abovementioned methods only focus on a POS tag with the highest frequency or normalized frequency. In *AllCombi*, let the determined POS tag be the POS tag with the highest sum of the term-POS frequency. With this approach, more diversified context including long tail can be considered. The POS tag of t_A is P_1 because the sum of its frequency is 11 (5+3+3) which is the highest. We show the pseudo code of *MostLikelihood* in Algorithm 4.

4. EXPERIMENTS

We evaluated the proposed method with three variations of scoring functions *MaxFreq*, *MostLikelihood*, and *AllCombi*, and three existing methods, i.e., *Stanford*, *Caseless*, and *SingleFreq* for comparison. *Stanford* and *Caseless* are methods using the Stanford POS tagger with the default model and the caseless model, respectively. The POS tag appearing the most frequently in the corpus for each term is assigned to a query term with *SingleFreq*, which contributes accurate POS tagging [2].

We use ClueWeb09 Category B as a large-scale Web corpus. It is composed of 50 million English Web documents and the search target document collection of Web track topics. The data sets used in the experimental evaluations are the POS-tagged Web track topics and MS-251 which is a Microsoft search log and used in previous studies [3, 7].

4.1 Experiments with the POS-tagged Web Track Topics

Algorithm 1 Common process of the scoring functions

```
1: input: Q // a set of query terms
2: output: POSSet // returns resulting term-POS pairs
3: for all q1 ∈ Q do
4:   for all q2 ∈ Q (q2 ≠ q1) do
5:     // a query "q1 q2" is issued to TPDB.
6:     termPOSPairSet ← ScanDB(q1, q2)
7:     for all TPP ∈ termPOSPairSet do
8:       // count the number of term-POS pair TPP
9:       if count(TPPSet, TPP) = 0 then
10:        TPPSet_TPP ← 1
11:      else
12:        TPPSet_TPP ← count(TPPSet, TPP) + 1
13:      end if
14:    end for
15:    // an arbitrary scoring function is applied
16:    POSSet ← scoreTermPOS(TPPSet, POSSet)
17:  end for
18: end for
19: return POSSet
```

Algorithm 2 MaxFreq

```
1: // in case MaxFreq is applied
2: input: TPPSet, POSSet
3: output: POSSet
4: for all TPP ∈ TPPSet do
5:   // get term-POS of query term q1 from TPP
6:   TP_q1 ← getTermPOS(TPP, q1)
7:   if POSSet_q1 = φ then
8:     POSSet_q1 ← TP_q1
9:   else if count(POSSet, q1) < count(TPPSet, TPP)
     then
10:    POSSetq1 ← TP_q1
11:  end if
12:  // same process with query term q2 (lines 6–11)
13: end for
14: return POSSet
```

Algorithm 3 MostLikelihood

```
1: // in case MostLikelihood is applied
2: input: TPPSet, POSSet
3: output: POSSet
4: for all TPP ∈ TPPSet do
5:   // normalized with the number of all term-POS pairs
6:   tmpScore = count(TPPSet, TPP)/countAll(TPPSet)
7:   TP_q1 ← getTermPOS(TPP, q1)
8:   if POSSet_q1 = φ then
9:     POSSet_q1 ← TP_q1
10:  else if getScore(POSSet_q1) < tmpScore then
11:    POSSetq1 ← TP_q1
12:  end if
13:  // same process with query term q2 (lines 7–12)
14: end for
15: return POSSet
```

Algorithm 4 AllCombi

```
1: // in case AllCombi is applied
2: input: TPPSet, POSSet
3: output: POSSet
4: for all TPP ∈ TPPSet do
5:   tmpCnt ← count(TPPSet, TPP)
6:   TP_q1 ← getTermPOS(TPP, q1)
7:   if TPCnt_q1 = 0 then
8:     TPCntSet_q1 ← tmpCnt
9:   else
10:    TPCntSet_q1 ← TPCntSet_q1 + tmpCnt
11:  end if
12:  // same process with query term q2 (lines 6–11)
13: end for
14: for all TPCnt_q1 ∈ TPCntSet_q1 do
15:  if POSSet_q1 = φ then
16:    POSSet_q1 ← getTermPOS(TPCnt_q1)
17:  else if count(POSSet, q1) < TPCnt_q1 then
18:    POSSet_q1 ← getTermPOS(TPCnt_q1)
19:  end if
20: end for
21: // same process with query term q2 (lines 14–20)
22: return POSSet
```

Table 3: Experiments with the POS-tagged Web track topics

	precision	improvement
MaxFreq	.814	1.07
MostLikelihood	.814	1.07
AllCombi	.821	1.08
Caseless	.763	1.00
SingleFreq	.702	0.98
Stanford	.547	0.56

Table 3 shows the precision[7] with all query terms of each method and the improvement ratio compared with the most accurate method in the comparison methods *Caseless*. The result shows that the accuracy of *MaxFreq* and *MostLikelihood* are the same. This suggests that it does not make much difference whether we employ frequency or probability in accurate POS tagging to queries. Then, *AllCombi* attained the highest precision of the scoring functions. This suggests that considering more diversified context is useful in correctly identifying POS tags. Note that sign tests confirmed that every of the proposed method, i.e., *AllCombi* ($p < 0.01$), *MaxFreq* ($p < 0.05$), and *MostLikelihood* ($p < 0.05$), significantly outperformed *Caseless*.

AllCombi succeeded in identifying more proper nouns including those listed in Section 2.1, as compared to existing methods. In addition, *AllCombi* diminished the chances of being applied in a partial grammatical rule of no use. As a result, "*pay*" in the query "*gs pay rate*" is correctly identified as a common noun. However, "*lower*" in the query "*lower heart rate*" is still mistakenly identified as an adjective. In

[7]Under such a situation that all POS tags are aggregated, values of precision and recall are the same.

Table 4: Precision of each method by POS tag

precision	common noun	proper noun	verb	adjective
MaxFreq	.825	.833	.769	.647
MostLikelihood	.825	.833	.769	.647
AllCombi	.825	.860	.714	.629
Caseless	.789	.751	.733	.690
SingleFreq	.775	.670	.533	.581
Stanford	.550	1.0	.722	.451

Table 5: Recall of each method by POS tag

recall	common noun	proper noun	verb	adjective
MaxFreq	.846	.765	.667	.917
MostLikelihood	.846	.765	.667	.917
AllCombi	.872	.755	.667	.917
Caseless	.769	.740	.733	.833
SingleFreq	.636	.755	.533	.750
Stanford	.985	.010	.867	.958

addition, as a example of the negative effect of the proposed method, "*president*" in the query "*president united states*" was mistakenly identified as a proper noun, although it was correctly identified as a common noun with *Stanford*. The fact that "*president*" in the corpus are often identified as proper nouns causes the error. As a solution for this issue, we need to normalize term weights, which is a part of future work.

For deeper analysis, we examined precision, recall, and F-measure (harmonic mean of precision and recall) of each method by POS tag as shown in Tables 4, 5, and 6. *AllCombi* correctly identified weather a noun is a common noun or a proper noun. On the other hand, *MaxFreq* and *MostLikelihood* show better accuracy for other POS tags such as verb and adjective. Therefore, there is a possibility that the proposed method can be improved when the scoring functions are utilized concurrently. For example, *MaxFreq* or *MostLikelihood* assign POS tags as a first step. Then, terms tagged noun (common noun or proper noun) are re-tagged with *AllCombi*.

4.2 Experiments with MS-251

Both the previous studies [3, 7] used MS-251 in the experiments; however, they differ in the POS tag classification. POS tag classification in this study, 14 types of POS tags, are introduced from [7], while [3] defines three types of POS tags, *noun*, *verb*, and *other*. Note that the authors of [7] re-annotated oracle tags of the data set. As a result of re-annotation, the POS tags of some terms drastically changed. For example, "*ask*" in the queryID 3000057 is re-annotated from a verb to a proper noun. Because it is fairly difficult to judge which label is more appropriate, we just ignored queries containing any conflict between the definitions of [3] and [7]. Remaining queries are 189 in total[8].

As shown in Table 7, the order of precision is different from the one with Web track topics. In this experiment, *MostLikelihood* is the most accurate. Accuracy of each method by POS tag shows largely the same trend of the former experiment. The reason why the order of accuracy is changed is that the ratios of POS tags in this data set are different from the first data set. The accuracy of the proposed methods is better than the best method in the previous study [7], although these methods cannot be compared strictly because the queries used to evaluate these methods are different.

In summary, the results through preliminary investigations and experimental evaluations, the POS-tagged Web track topics largely have the same trend and property as other data sets composed of search logs [2, 3, 7]. Judging from accuracy of existing methods, the level of difficulty is largely the same as a data set for a POS tagging task to queries. Therefore, we conclude that the data set constructed in this study is applicable for the task. Moreover, this data set can be directly applied to the researches of developing IR systems leveraging POS tags.

5. RELATED WORK

Studies of POS tagging to queries are classified into three: 1) knowledge based [8], 2) comprehending query structure [1, 2, 10, 4, 13], and 3) leveraging results of sentence level morphological analysis [3, 7]. Our study is included in the third one. A knowledge base is required with the first approach, while manually judged labels for supervised learning methods are needed for the second approach. Building a knowledge base or annotating labels are not always easy.

Regarding the third approach, the top results of pseudo-relevance feedback [3] or snippets of user-browsed documents [7] have been used in the previous studies. These researches simply count frequency of POS tags, while the proposed method is focused on co-occurrence of query terms. These studies also differ from ours in that they target a limited number of documents while our study utilize a large-scale Web corpus. As an actual fact, our experiments suggest

[8] `http://www.lsc.cs.titech.ac.jp/keyaki/dataSet/MS251noConflict.tsv`

Table 6: F-measure of each method by POS tag

F-measure	common noun	proper noun	verb	adjective
MaxFreq	.835	.798	.714	.759
MostLikelihood	.835	.798	.714	.759
AllCombi	.848	.804	.690	.746
Caseless	.779	.746	.733	.755
SingleFreq	.699	.710	.533	.655
Stanford	.706	.020	.788	.613

Table 7: Experiments with MS-251

	precision	improvement
MaxFreq	.890	1.04
MostLikelihood	.895	1.04
AllCombi	.893	1.04
the best method in [7]	.858	1.00

that global statistics are useful for accurate POS tagging. Additionally, it is much easier to accumulate a large-scale Web corpus compared with collecting user's search logs.

6. CONCLUSION

We propose a POS tagging method for queries using the results of sentence level morphological analysis of a large-scale Web corpus. We devised three variations of scoring functions, namely, *MaxFreq*, *MostLikelihood*, and *AllCombi*. The experimental results showed that the proposed method outperforms those using existing tools and the state-of-the-art method. Then, strong and weak points are different by scoring function. This suggest that a combination of scoring functions has a possibility to improve accuracy.

In future, we will normalize the term weight to prevent errors from occurring by using the proposed method. In addition, we plan to apply the data set we created to IR researches. A proposal of database schema design for fast POS tagging is also a part of future work.

7. ACKNOWLEDGEMENTS

This work was partly supported by JSPS KAKENHI Grant Numbers 15H02701, 15K20990, 16H02908, and 26280115.

8. REFERENCES

[1] J. Allan and H. Raghavan. Using Part-of-speech Patterns to Reduce Query Ambiguity. In *Proc. of SIGIR*, pages 307–314, 2002.

[2] C. Barr, R. Jones, and M. Regelson. The Linguistic Structure of English Web-Search Queries. In *Proc. of EMNLP*, pages 1021–1030, 2008.

[3] M. Bendersky, W. B. Croft, and D. A. Smith. Structural Annotation of Search Queries Using Pseudo–Relevance Feedback. In *Proc. of CIKM*, pages 1537–1540, 2010.

[4] M. Bendersky, W. B. Croft, and D. A. Smith. Joint Annotation of Search Queries. In *Proc. of HLT*, pages 102–111, 2011.

[5] A. Chowdhury and M. C. Mccabe. Improving Information Retrieval Systems using Part of Speech Tagging. Technical report, Univ. of Maryland, 1993.

[6] F. Crestani, M. Sanderson, M. Theophylactou, and M. Lalmas. Short Queries, Natural Language and Spoken Document Retrieval: Experiments at Glasgow University. In *Proc. of TREC-6*, pages 667–686, 1998.

[7] K. Ganchev, K. Hall, R. McDonald, and S. Petrov. Using Search-Logs to Improve Query Tagging. In *Proc. of ACL*, pages 238–242, 2012.

[8] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou. Short Text Understanding Through Lexical-Semantic Analysis. In *Proc. of ICDE*, pages 495–506, 2015.

[9] I.-H. Kang and G. Kim. Query Type Classification for Web Document Retrieval. In *Proc. of SIGIR*, pages 64–71, 2003.

[10] X. Li. Understanding the Semantic Structure of Noun Phrase Queries. In *Proc. of ACL*, pages 1337–1345, 2010.

[11] X. Li, Y.-Y. Wang, and A. Acero. Extracting Structured Information from User Queries with Semi-Supervised Conditional Random Fields. In *Proc. of SIGIR*, pages 572–579, 2009.

[12] S. Petrov, D. Das, and R. McDonald. A Universal Part-of-Speech Tagset. In *Proc. of LREC*, 2012.

[13] J. Pound, A. K. Hudek, I. F. Ilyas, and G. Weddell. Interpreting Keyword Queries over Web Knowledge Bases. In *Proc. of CIKM*, pages 305–314, 2012.

[14] R. S. Roy, Y. Vyas, N. Ganguly, and M. Choudhury. Improving Unsupervised Query Segmentation using Parts-of-Speech Sequence Information. In *Proc. of SIGIR*, pages 935–938, 2014.

[15] G. Salton. *The SMART Retrieval System–Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[16] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *in Proceedings of NAACL*, pages 173–180, 2003.

[17] Z. Zhong and H. T. Ng. Word Sense Disambiguation Improves Information Retrieval. In *in Proc of ACL*, pages 273–282, 2012.