University of Hyogo at NTCIR-11 TaskMine by Dependency Parsing

Takayuki Yumoto
Graduate School of Engineering, University of Hyogo
2167 Shosha, Himeji
Hyogo, Japan
yumoto@eng.u-hyogo.ac.jp

ABSTRACT

Our method for TaskMine consists of three steps. Firstly, we search for seed Web pages by using the query string with a word "houhou", which means "method". We collect more pages in consideration of the anchor texts in the seed pages. Then, we find pairs of chunks satisfying predefined patterns by dependency parsing on the sentences. We extract a target and a postpositional particle from a depending chunk, and extract an operation and a negation (if it exists) from a depended chunk. We regard a quadruplet of them as a subtask. Finally, we rank the extracted subtasks by their frequency based score.

Team Name

uhyg

Subtasks

TaskMine

Keywords

Dependency parsing, query expansion, site frequency

1. INTRODUCTION

We participated in the NTCIR-11 TaskMine subtask. This aims at finding subtasks for a given task[4]. Our method consists of three steps, collecting Web pages, extracting subtasks and ranking subtasks. In the first step, we collect Web pages that contain subtasks for a given task by a query expansion approach. In the second step, we extract subtasks as quadruplets of a target and a postpositional particle and an operation and a negation (if it exists) from collected Web pages. To extract subtasks, we focus on dependency relationship between words. In the third step, we rank subtasks by frequency based scores.

2. COLLECTING WEB PAGES

2.1 Query Expansion

We use a Web search API and obtain Web pages to extract subtasks. In TaskMine, query strings are given. Some query strings can be interpreted as not for task. To solve this problem, we take a keyword spice approach[5]. We combine a given query and a word "houhou" and make a search query. "Houhou" means "method". By this query, we limit obtained pages to pages describing some kinds of methods. We submit this query to Web search API, and we collect 50 Web pages. We used Bing Search API[1]. Here, we call these pages as seed pages.

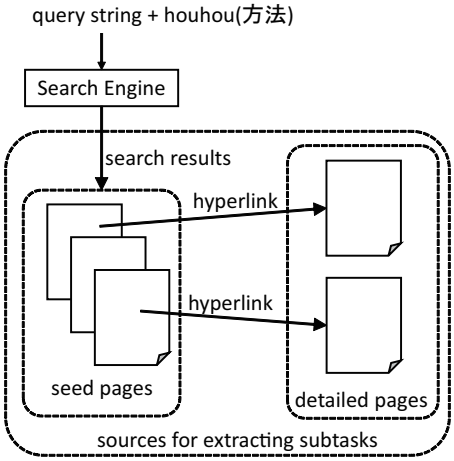

Figure 1: Web pages for extracting subtasks

2.2 Finding Detailed Pages

Seed pages are sometimes top pages that only contain menus without contents. Such pages contain hyperlinks to pages that describe actual content in the same URL domain. Therefore, we extract hyperlinks whose destination domain is the same as the source URL of the seed pages and whose anchor text satisfies either of following conditions.

- The anchor text contains all keywords extracted from a given query.
- The title of the seed page contains some keywords and the anchor text contains the rest of the keywords.

Keywords are nouns and verbs extracted by Japanese morphological analyzer Mecab. Then, we collect the destination pages. We call these destination pages as detailed pages. Multiple detailed pages can be extracted from one seed page. We extract subtasks from seed pages and their detailed pages. We show an image of Web pages for extracting subtasks in Figure 1.

[1] http://datamarket.azure.com/dataset/bing/search

3. EXTRACTING SUBTASKS

3.1 Subtask Model

We express a subtask as a quadruplet of follows :

- target : Target is a thing used to achieve a subtask. Target is a noun.
- postpositional particle : The postpositional particle follows a target word, and expresses a role of the target word. In Japanese sentence, most of nouns are followed by postpositional particles. A role of postpositional particles is similar to prepositions but they are often omitted in English.
- operation : Operation should be executed to achieve subtask. Operations are predicates in the sentences. In most cases, we consider that operation is a verb. Operation is sometimes a noun.
- negation : Negation expresses whether there is a negation in a chunk containing an operation. Here, "¬" denotes that a negation exists and "−" denotes no negations.

For example, from a sentence "masuku wo tsukeru", which means "wear a mask", a quadruplet (masuku, wo, tsukeru, −). "Masuku" means "(flu) mask", and "wo" has no corresponding words in English, and "tsukeru" means "wear".

3.2 Extracting Quadruplets

To extract subtask quadruplets, we use a dependency parser. We used Japanese morphological analyzer Mecab[2], and Japanese dependency parser Cabocha[3], [1].

First, we find operations. In Japanese sentences, predicates locate at the end of sentences. Therefore, verbs in chunks at the end of sentences are candidates of operations. We also regard nouns at the end of sentences as the candidates, because some Japanese sentences end with nouns. On the other hand, one sentence sometimes contains multiple predicates. Therefore, we regard that verbs in chunks depending to candidates of operations are also candidates of operations. At first, we extract operation only from the chunk at the end of sentence. Then, we check chunks from the end of the sentence and add chunks depending to candidates of operations as a new candidate. If the chunk contains a negation, we extract it.

Next, we find targets and postpositional particles. We extract a target and a postpositional particle from a sequence of a noun and a postpositional particle that depends to candidates of operations. Then, we make a quadruplet.

We show two examples. First one is shown in Figure 2. "/" means a boundary of chunks, and arrows mean dependency relationship between chunks. We show meanings in English with parentheses. In this example, "dekakeru" is first candidate of operations. "Tsukete" is regarded as a candidate of operations because it depends to "dekakeru". "Tsukete" is also depended by "masuku wo", which contains a target "masuku". Therefore, it is a operation. On the other hand, "dekakeru" isn't depended by no chunks containing targets, and it is not a operation. From the sentence in Figure 2, we obtain a quadruplet (masuku, wo, tasukeru, −).

Second example is shown in Figure 3. In this example, the last chunk contains a negation. There are two chunks with targets, "soto ni" and "sentakumono wo". From this

Figure 2: Example of sentence with two predicates

sentence, we extract two quadruplets, (soto, ni, hosu, ¬) and (sentakumono, wo, hosu, ¬). "Hosu" means "hang".

Figure 3: Example of sentence with a negation

4. RANKING SUBTASKS

Extracted quadruplets are ranked based on their frequency. We regard that frequent quadruplets are better subtasks. We use site frequency instead of page frequency. This is because pages in the same site often contain same phrases in their headers or footers such as the copyright statements.

If words in quadruplets have same meaning but have different expressions, however, their frequency becomes lower. To solve this problem, we use Wikipedia corpus. If two anchor texts link to the same entry page in Wikipedia, we regard the two words as same.

We propose two ranking functions. In the first one, we focus on a pair of a target and an operation. If targets and operations in two quadruplets are same respectively, we regard the quadruplets as same in the first ranking function. By counting site frequency of quadruplets, we rank them in the descending order. We describe this algorithm as quadruplet-freq.

In the first ranking function, however, some proper but infrequent subtasks are ranked at low rank. We consider that frequent targets and frequent operations are likely to be important respectively. To collect various subtasks, we also count site frequency of targets and operations respectively. In the second ranking function, we use the minimum of two frequencies as the first key of the ranking and the maximum of two frequencies as the second key. In this ranking function, combination of frequent targets and frequent operations are highly ranked. We describe this algorithm as target-op-freq.

In both ranking, several quadruplets compose one group. To make a subtask string, we select one quadruplet from each group. First, we select quadruplets containing the most frequent postpositional particle in the group. Next, we select a quadruplet containing a negation if it exists. Otherwise, we select a quadruplet without a negation. This is for weighting extracted negations because our algorithm sometimes fails to extract negations.

Subtask strings are made by concatenating items in quadruplets. For example, a subtask string "masuku wo tsukeru" is obtained from a quadruplet (masuku, wo, tsukeru, −).

5. EXPERIMENTS AND DISCUSSION

5.1 Experimental Results

We submitted two runs as follows:

- uhyg1 : The results by target-op-freq
- uhyg2 : The results by quadruplet-freq

We show average of nDCG@k($k = 1, 5, 10, 50$) in Table 1. We compare nDCG@k($k = 1, 5, 10, 50$) of uhyg1 and uhyg2 in Table 2. From Table 1, uhyg2 is better than uhyg1 with regard to average of nDCG@k. From Table 2, uhyg2 is also better than uhyg1 with regard to the number of queries. Therefore, we conclude quadruplet-freq(uhyg2) is a better ranking score than target-op-freq(uhyg1).

Table 1: Average of nDCG@k

	$k=1$	$k=5$	$k=10$	$k=50$
uhyg1	0.098	0.119	0.132	0.166
uhyg2	0.109	0.150	0.171	0.191

Table 2: Comparison of nDCG@k between uhyg1 and uhyg2 (uhyg1 > uhyg2 : the number of queries where nDCG@k of uhyg1 is higher than the one of uhyg2, uhyg1 < uhyg2 : the number of queries where nDCG@k of uhyg1 is lower than the one of uhyg2, not found : the number of queries where answers are not found at top k)

	$k=1$	$k=5$	$k=10$	$k=50$
uhyg1>uhyg2	4	14	14	15
uhyg1<uhyg2	5	21	32	34
not found	39	13	4	1

For further analysis, we show nDCG@k for categories by uhyg1 and uhyg2 in Table 3 and 4 respectively. We also compare uhyg1 and uhyg2 from following aspects in Table 5:

- not found : the number of the queries where any answers cannot be found in top 10
- @5<@10 : the number of the queries where nDCG@10 is better than nDCG@5
- @10<@50 : the number of the queries where nDCG@50 is better than nDCG@10

The nDCG@k increases as k increases in many queries. This means that our algorithm ranks the correct answers at lower ranks. The reasons are discussed in later.

Table 3: nDCG@k for categories by uhyg1

	$k=1$	$k=5$	$k=10$	$k=50$
Health	0.042	0.076	0.107	0.145
Education	0.100	0.137	0.130	0.167
Daily life	0.183	0.118	0.113	0.143
Sequential	0.123	0.186	0.202	0.231

Table 4: nDCG@k for categories by uhyg2

	$k=1$	$k=5$	$k=10$	$k=50$
Health	0.140	0.141	0.173	0.191
Education	0.000	0.075	0.107	0.161
Daily life	0.167	0.153	0.150	0.154
Sequential	0.100	0.237	0.253	0.259

Table 5: Comparison of nDCG@k

	not found	@5<@10	@10<@50
uhyg1	9	27	34
uhyg2	8	26	35

5.2 Discussion

The best average of nDCG of our method was 0.191, and it should be improved. To improve our algorithm, there are following three problems to be solved:

- the part where subtasks are extracted
- same subtask in different expressions
- limitation of model

The first problem is that quadruplets are extracted from the part where tasks aren't described. By our query expansion approach, we succeed to collect pages containing task descriptions. However, these pages also contain other information. For example, the page about a cold contains not only information to prevent a cold but also explanation of mechanism to catch a cold. To improve our algorithm, we need to specify the part where tasks are described.

The second problem is that some subtasks can be expressed by several forms. In this case, frequency becomes lower than actual. To solve this problem, we used Wikipedia corpus, and aggregate synonyms. However, this isn't sometimes enough. In the context of subtasks for vision correction by contact lens, "use", "wear" and "buy" are operations for the same subtasks but aren't synonyms. We need to develop a method to aggregate them.

The third problem is that some subtasks cannot be expressed by quadruplets. For example, "exercise" is a subtask to lose weight but it has no target. On the other hand, a subtask "tane ni kizu wo tsukeru(scratch a seed)" to grow pavillon contains two targets "tane" and "kizu". To solve this problem, our subtask model needs to be more flexible.

6. CONCLUSION

We proposed a method to find subtasks by using a query expansion and dependency parsing. Our method consists of three steps, collecting Web pages, extracting subtasks and ranking subtasks. In the first step, we collect Web pages that contain subtasks for a given task by a query expansion approach. In the second step, we extract subtasks as quadruplets of a target and a postpositional particle and an operation and a negation (if it exists) from collected Web pages. To extract subtasks, we focus on dependency relationship between words. In the third step, we rank the subtasks. We proposed two ranking functions quadruplet-freq and target-op-freq, that are based on site frequency. From

the evaluation, we found that quadruplet-freq is better than target-op-freq. The nDCG@50 by our method was 0.191, and it should be improved. We found several problems for it.

7. ACKNOWLEDGMENTS

This work was supported by Grant-in-Aid for Young Scientists (B) 24700097 from JSPS.

8. REFERENCES

[1] Cabocha - Yet another Japanese dependency structure analyzer. https://code.google.com/p/cabocha/. Accessed: 2014-08-21.

[2] MeCab: Yet another part-of-speech and morphological analyzer. http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html. Accessed: 2014-08-21.

[3] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In Proceedings of the 6th Conference on Natural Language Learning 2002, pages 63–69, 2002.

[4] Y. Liu, R. Song, M. Zhang, Z. Dou, T. Yamamoto, M. Kato, H. Ohshima, and K. Zhou. Overview of the NTCIR-11 IMine task. In Proceedings of the NTCIR-11, 2014.

[5] S. Oyama, T. Kokubo, and T. Ishida. Domain-specific web search with keyword spices. IEEE Transactions on Knowledge and Data Engineering, 16(1):17–27, January 2004.