A Question Answering System Based on Vector Similarity

Takuya Kosugi, Hiroya Susuki, Hiroyuki Okamoto, Hiroaki Saito
Department of Information and Computer Science
Keio University
3-14-1, Hiyoshi, Kouhoku-ku, Yokohama, 223-8522, Japan
{kosugi,susuki,motch,hxs}@nak.ics.keio.ac.jp

Abstract

This paper reports on an implementation of a question answering system with the vector similarity scoring method. Our question answering system consists of four modules. The question analyzer classifies questions with manually created regular expressions. The document retrieval engine chooses the related articles using the vector space retrieval method. The named entity extractor finds answer candidates in the retrieved articles. The answer selector uses similarity score calculated by the document retrieval engine to decide the final answers to be presented to the user.

The evaluation of our system on NTCIR Question Answering Challenge 2 is 0.242 *in recall,* 0.095 *in precision,* 0.137 *in F-measure and* 0.231 *in MRR.*

Keywords: *Question Answering, Vector space search*

1 Introduction

Question Answering (QA) is to retrieve the exact answer for a natural language question rather than to present the document for artificial keywords. Thus, the main task of QA is real information retrieval. NTCIR held Question Answering Challenge (QAC) task to evaluate QA systems [4]. A QA system is usually equipped with a search engine to get articles which include the correct answer. Many systems use a full-text search engine for this task [5, 3, 6]. However, to use a full-text search engine for QA is not adequate. Making combination of keywords for a search engine is usually done in an ad-hoc way.

A vector space method is a document retrieval way which calculates similarity of two documents. Thus, it does not need to use ad-hoc way to combine keywords. This feature is suitable to QA, and the similarity score can also be used for answer selection.

2 The Structure of our QA System

Our QA system consists of four modules: the question analyzer, the article retrieval engine, the named entity extractor, and the answer selector. The relation of these modules is shown in Figure 1.

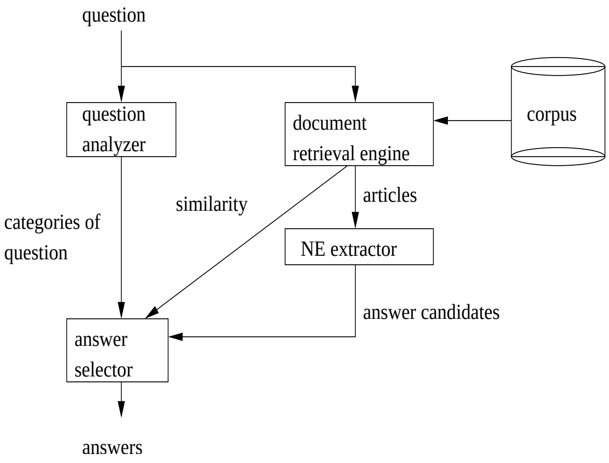

Figure 1. The structure of our QA system

2.1 Question analyzer

This module classifies the question into the predefined categories. The number of categories is nine, eight of which are the named entity (NE) categories in IREX [1]: ORGANIZATION, PERSON, LOCATION, ARTIFACT, DATE, TIME, MONEY, and PERCENT. Along with these categories, NUMBER category is added for general numerical expressions.

Question classification is performed by the manually created regular expression rules. If more than two rules match the question, all the matched categories are assigned. When no rule matches the question, the system assigns all the categories.

The regular expressions for this module are shown in Figure 2.

```
((誰|だれ)|(本名|旧姓|((力士|首相) の名前)) は (何|なに|なん)|((何|なに|なん) という名前)) PERSON

((((どこ|何処) の|どの)(星|国|領|県|府|場所))|((星|国|領|県|府|場所|地|地名|首都|市|町|村|郡|山|川|湖) は (どこ|何処)(です|。|？)?)|((星|地名|山|川|湖).* は (何|なに|なん))|((何|なに|なん) という (星|国|領|県|府|場所|地|地名|首都|市|町|村|郡|山|川|湖))|((何|なに)(星|国|領|県|府|市|町|村|郡|山|川|湖))|(どこ|何処)) LOCATION

((((どこ|何処) の|どの)(会社|企業|組織|団体|.*社|.*店|チーム))|((社|会社|企業|組織|団体|店|主催者|派) は (どこ|何処)(です|。|？)?)|(どこ|何処) に (あり|ある)|((会社名|組織|団体|省庁).*は (何|なに|なん))|((何|なに|なん) という (会社名|組織|団体|省庁))|((何|なに)銀行)|(どこ|何処)) ORGANIZATION

(いつ|(何|なに|なん)(年|月|日)) DATE

((何|なに|なん)(時間|分|秒)) TIME

(いくら|(価格は (いくら|((どの|どれ)(くらい|ぐらい|位))))) MONEY

((何 (割|パーセント|%))|(率|割合) は.*(どの程度|((どの|どれ)(くらい|ぐらい|位))|どれだけ|(いくつ|いくら))|(どの程度|((どの|どれ)(くらい|ぐらい|位))|どれだけ|(いくつ|いくら)) の (率|割合)) PERCENT

何 (インチ|カラット|キロ|キログラム|キロメートル|キロリットル|グラム|センチ|センチメートル|ダース|トン|ノット|フィート|ページ|ポンド|マイル|ミリ|ミリメートル|メートル|ヤード|リットル|ヶ月|握|案|位|囲|羽|雨|駅|億|家族|架|箇|荷|画|回|回転|塊|海里|階|角|笠|株|冠|巻|竿|管|缶|貫|間|館|基|期|機|客|脚|球|級|鏡|局|曲|斤|句|区|区画|駒|軍|桁|件|県|軒|個|個所|戸|戸前|口|孔|校|構|行|号|合|座|座席|才|歳|冊|刷|札|皿|字|寺|時|時限|次|次元|社|尺|手|種類|首|周|週|週間|重|巡|升|小節|章|丈|場|条|畳|色|食|審|人|人前|寸|世|世紀|世帯|席|石|節|説|戦|選|銭|膳|糎|組|層|相|息|束|足|村|太刀|打|駄|体|対|袋|隊|代|台|卓|単位|担|段|着|丁|兆|帖|張|町|町歩|通|通話|坪|挺|提|締|艇身|摘|滴|店|点|度|投|棟|灯|当|等|等身|頭|堂|日|年|捻|把|杷|波|派|馬力|敗|杯|倍|拍|泊|箱|鉢|発|反|版|犯|班|晩|番|尾|琵|匹|筆|俵|票|品|斧|幅|分|文|文字|頁|編|辺|便|歩|包|房|本|枚|幕|枕|味|名|面|毛|目|匁|夜|葉|翼|絡|里|流|粒|両|稜|領|厘|輪|例|礼|列|話|椀|勝|敗|校) NUMBER

(広さ|面積|長さ|速さ|最高速|高さ|数|全長|震度|人口|座席|時差|量|温度|重さ|体積|幅|速度|最長|最短|距離|太さ|大きさ|小ささ|細さ|薄さ|電力|時速|密度|湿度)は.*(((どの|どれ)(くらい|ぐらい|位|程度))|いくら|いくつ) NUMBER

((どの|どれ)(くらい|ぐらい|位|程度)) の (広さ|面積|長さ|速さ|最高速|高さ|数|全長|震度|人口|座席|時差|量|温度|重さ|体積|幅|速度|最長|最短|距離|太さ|大きさ|小ささ|細さ|薄さ|電力|時速|密度|湿度) NUMBER

(ビル|建物|タワー|塔|城|ドラマ|映画|作品|代表作|続編|タイトル|賞|次回作|原作|楽器|著書|邦題|ダム|曲名|番組|酒).*は.*(何|なに|なん) ARTIFACT
```

Figure 2. Regular expressions used in the question analyzer

2.2 Article retrieval engine

This module retrieves articles which probably include the answer expressions. A vector space method is adopted for article retrieval. In this method a question and each article is represented by a vector. The similarity of a question vector and an article vector is measured by the cosine measure.

We use ChaSen [8] to analyze articles. For a vector space searching method, we use tf-idf weighting.

Top 5 articles are passed to the next module(NE extractor).

2.3 NE extractor

This module extracts answer candidates. In many cases, the answer candidates are NEs. Thus, to find answer candidates is extracting named entities from the retrieved documents. We use CaboCha [7] as an NE extractor. Although CaboCha is a dependency structure analyzer for Japanese, it can also extract NEs. When CaboCha extracts NEs, it labels 8 categories defined by IREX. We add an extraction rule for the NUMBER category, which is a very simple way which extracts all numerical expressions.

2.4 Answer selector

This module grades each answer candidate found out by the NE extractor and selects top 5 NEs to be presented as answers. First, this module checks if the categories classified by the question analyzer is included in the NE categories by the NE extractor. Second, the answer candidates which match the question categories is graded. The formula (1) is used for scoring,

$$\sum_{i=0}^{N} c_i(1 + \log \sum_{j=0}^{M} w_{ij}) \qquad (1)$$

where c_i is the similarity score of the ith article retrieved by the article retrieval engine, w_{ij} is the jth answer candidate in the ith article, N is the number of the extracted articles, and M is the number of the answer candidates. The formula above indicates that an NE which appears in many articles is more effective than it appears many times in an article.

The top 5 answer candidates are presented as the answers.

3 Result

This section describes the result of all the 200 questions in NTCIR QAC-2.

3.1 Question analyzer

We compare the output of the question analyzer with the correct classification by a person. The human analyzer successfully classified 153 questions into 9 categories; 47 questions cannot be classified into any categories. The result of the question analyzer is shown in Table 1.

Table 1. Result of the question analyzer

category	# of questions	# of classified articles	# of correct doc.	recall	prec.	F
ORGANIZATION	18	119	18	1.00	0.15	0.26
PERSON	47	118	45	0.96	0.38	0.54
LOCATION	36	125	36	1.00	0.29	0.45
ARTIFACT	20	82	20	1.00	0.24	0.39
DATE	8	84	8	1.00	0.10	0.18
TIME	3	75	2	0.67	0.03	0.06
MONEY	2	77	2	1.00	0.03	0.06
PERCENT	0	74	0	N/A	N/A	N/A
NUMBER	19	90	19	1.00	0.21	0.35
total	153	844	150	0.98	0.18	0.30

As mentioned earlier, our question analyzer assigns all the categories when it fails the classification into 9 categories. In QAC-2, 74 questions cannot be classified. This causes the low precision. If our analyzer assigns no category in that case, the precision rises sharply while the recall falls mildly as shown in table 2.

Table 2. Result of the question analyzer (modified)

category	# of questions	# of classified articles	# of correct doc.	recall	prec.	F
ORGANIZATION	18	45	11	0.61	0.24	0.34
PERSON	47	44	44	0.94	1.00	0.97
LOCATION	36	51	35	0.97	0.69	0.81
ARTIFACT	20	8	3	0.15	0.38	0.21
DATE	8	10	8	1.00	0.80	0.89
TIME	3	1	1	0.33	1.00	0.50
MONEY	2	3	2	1.00	0.67	0.80
PERCENT	0	0	0	N/A	N/A	N/A
NUMBER	19	15	12	0.63	0.80	0.70
total	153	177	116	0.76	0.66	0.71

3.2 Article retrieval engine

Top 5 articles chosen by the article retrieval engine contains the correct answer for 164 questions. The number of articles which has the answer is shown in Table 3.

Table 3. The number of documents which contain the correct answer

rank	Mainich 98	Mainichi 99	Yomiuri 98	Yomiuri 99
1	74	61	57	49
2	18	21	13	13
3	7	6	6	8
4	6	7	4	7
5	0	5	6	5

According to this result, when the article retrieval engine can find the article with the correct answer, 80 % of them contain the correct answer in the top 2 articles.

3.3 NE extractor

CaboCha performs NE extraction in our system. The score of NE extractor is evaluated by Yamada et. al. [9] The result of evaluation is shown in Table 4.

Table 4. Result of NE extraction [9]

Named Entity	F measure(CaboCha)
ARTIFACT	0.471
DATE	0.922
LOCATION	0.825
MONEY	0.943
ORGANIZATION	0.790
PERCENT	0.942
PERSON	0.863
TIME	0.832
total	0.832

The result above indicates that ARTIFACT extraction needs improvement.

3.4 Answer selector

To evaluate the answer selector alone, the questions which the retrieval engine cannot find the correct articles are eliminated. The result is shown in Table 5.

Table 5. Result of the answer selector

category	# of questions	MRR
ARTIFACT	3	0.7
DATE	8	0.2
LOCATION	35	0.23
MONEY	2	0.5
ORGANIZATION	11	0.20
PERCENT	0	0
PERSON	44	0.41
TIME	1	0.3
NUMBER	12	0.17
total	116	0.32

3.5 The entire system

The result of the entire system is shown in Table 6.

Table 6. NTCIR QAC2 result

# of correct answers	95
recall	0.242
precision	0.095
F measure	0.137
MRR	0.231

4 Discussion

In this section, we discuss the result of each module.

4.1 Question analyzer

The result of the question analyzer affects the performance of the entire system. In the evaluation, the recall rate was high, but the precision was very low. This was due to the questions which do not match the regular expression patterns. There are two reasons for this:

- The patterns for classification are not enough.
- The question itself cannot be classified into the predefined categories.

The number of questions which do not match any patterns was 74. There are 47 questions which cannot be classified even by the humans. Therefore, new categories should be created as well as more patterns. We should be careful, however, not to add new patterns indiscreetly, because it could decrease the precision performance.

4.2 Article retrieval engine

The article retrieval engine can collect articles which contains the correct answer in 80 % of the questions. And 80 % of them are ranked within top 2.

Some questions cannot be retrieved correctly due to the misleading information. For example, QAC2-10131-01: “The world’s longest bridge is the Second Lake Pontchartrain Causeway in the United States. What is the world’s longest bridge for railroad?” has lots of unnecessary information. This kind of misleading information should be eliminated.

Another cause of retrieving wrong articles is that the question is too short to contain enough information. QAC2-10080-01: “What does ‘flugels’ of the Yokohama Flugels mean?” has only two content words; “flugels” and “Yokohama”. Questions which do not contain sufficient information tend to make the article retrieval engine find unrelated articles. To find correct articles from scarce information, filtering like syntactical structure matching could be effective.

To make the article retrieval engine better, a sophisticated retrieval method might help. For example, to use Latent Semantic Indexing (LSI) instead of a simple vector space method would be a good idea. LSI is a way of dimensionality reduction on the vector space retrieval [2], since the vector space retrieval with LSI can exploit co-occurrence information as well.

4.3 NE extractor

This module only uses an existing NE extractor, and the result of it is shown in Table 4. The result is good except ARTIFACT. ARTIFACT includes the title of dramas, literary work and prise. In many cases, these expressions are recognized as an unknown word in morphological analysis. An unknown word has little information to other morphemes, thus this leads to classification failure.

The number of categories seems too few for QA; this classification is based on the NE task of IREX. More fine grained categories for QA would be effective. But to add new categories to CaboCha, learning by corpus with the new categories will be needed.

4.4 Answer selector

The scoring formula (1) has room for improvement. First, the NEs which occur in a question sentence have high scores. In many cases, the NE in a question sentence do not fit as the correct answer. The simplest way of avoiding this problem is to eliminate NEs in a question from the answer candidates. But some NEs which appear in a question can be the correct answer.

Second, this scoring method gives a high score to the answer candidate which appear in two or more articles. There are many articles happened in ’98 and ’99, since QAC2 uses the newspaper in ’98 and ’99 as corpora. There are few articles in other years. Therefore, if a question does not ask the event in ’98 or ’99, the system hypothesizes wrong answers frequently.

Third, this task must show the article ID with the answer. Our system gets the article ID from the top ranked article. Thus, although the system can successfully find the correct article, it may show the wrong article ID.

5 Conclusion

We have implemented and evaluated a question answering system which selects answers based on the vector similarity. This system has attained MRR=0.231 in QAC2.

References

[1] IREX (NE) http://www.csl.sony.co.jp/person/sekine/IREX/NE.

[2] H. S. Christpher D. Manning. *FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING*. MIT Press, 1999.

[3] J. Fukumoto, T. Endo, and T. Niwa. Ritsqa: Ritsumeikan question answring system used for qac-1. *Working Notes of the Third NTCIR Workshop Meeting. PartIV: Question Answering Challenge*, pages 113–116, October 2002.

[4] J. Fukumoto and T. Kato. Question answering challenge (qac-1) an evaluation of question answering task at ntcir workshop 3. *Working Notes of the Third NTCIR Workshop Meeting. PartI: Overview*, pages 77–86, October 2002.

[5] A. Ikeno and H. Ohnuma. Oki qa system for qac-1. *Working Notes of the Third NTCIR Workshop Meeting. PartIV: Question Answering Challenge*, pages 17–20, October 2002.

[6] K. Kawata, H. Sakai, and S. Masuyama. A question and answering system using newspaper corpus as a knowledge base. *Working Notes of the Third NTCIR Workshop Meeting. PartIV: Question Answering Challenge*, pages 25–29, October 2002.

[7] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[8] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. *Japanese Morphological Analysis System ChaSen version2.2.1 Manual*. 2000.

[9] H. Yamada, T. Kudo, and Y. Matsumoto. Japanese named entity extraction using support vector machine. *Transactions of IPSJ(in Japanese)*, 43(1):44–53, 2002.