ASQA: Academia Sinica Question Answering System for NTCIR-5 CLQA

Cheng-Wei Lee[12], Cheng-Wei Shih[1], Min-Yuh Day[1], Tzong-Han Tsai[1], Tian-Jian Jiang[1], Chia-Wei Wu[1], Cheng-Lung Sung[1], Yu-Ren Chen[1], Shih-Hung Wu[3], Wen-Lian Hsu[1§]

[1]Institute of Information Science, Academia Sinica, Taiwan, R.O.C
[2]Department of Computer Science, National Tsing-Hua University, Taiwan, R.O.C
[3]Department of CSIE, Chaoyang University of Technology, Taiwan, R.O.C
§corresponding author

{aska,dapi,myday,thtsai,tmjiang,cwwu,clsung,yrchen}@iis.sinica.edu.tw,
shwu@cyut.edu.tw, hsu@iis.sinica.edu.tw

Abstract

We propose a hybrid architecture for the NTCIR-5 CLQA C-C (Cross Language Question Answering from Chinese to Chinese) Task. Our system, the Academia Sinica Question-Answering System (ASQA), outputs exact answers to six types of factoid question: personal names, location names, organization names, artifacts, times, and numbers. The architecture of ASQA comprises four main components: Question Processing, Passage Retrieval, Answer Extraction, and Answer Ranking. ASQA successfully combines machine learning and knowledge-based approaches to answer Chinese factoid questions, achieving 37.5% and 44.5% Top1 accuracy for correct, and correct+unsupported answers, respectively.

Keywords: *InfoMap, information retrieval, named entity recognition, question answering (QA), question focus*

1. Introduction

With the high level of information overload on the Internet, responding to users' questions with exact answers is becoming increasingly important. Many international question answering contests has been held at conferences and workshops, such as TREC [10], CLEF [11], and NTCIR [12]. Our proposed system, the Academia Sinica Question Answering System (ASQA), participated in NTCIR-5 CLQA C-C Task, which was the first "Chinese to Chinese" factoid question answering contest in the world.

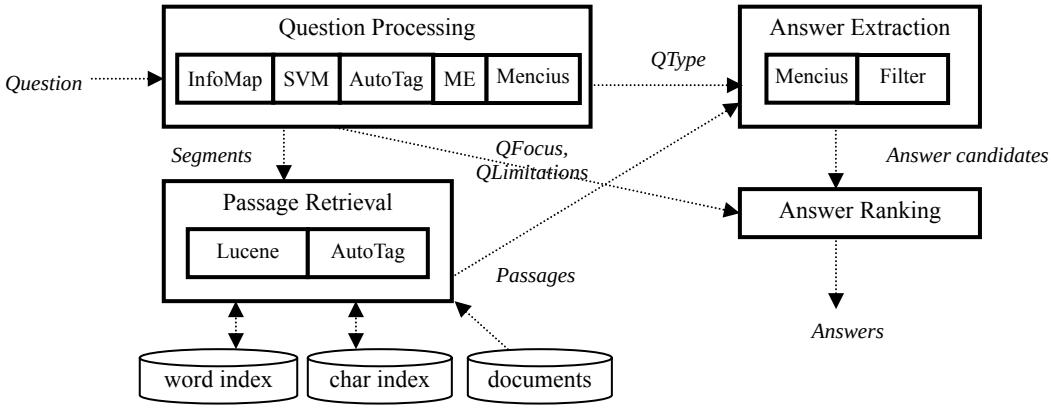

Figure 1. System architecture and data flow of ASQA. The outer rectangles are the four main modules. The inner rectangles are important sub-modules. The dashed arrows indicate the data flow between modules.

The architecture of ASQA comprises four main components: Question Processing, Passage Retrieval, Answer Extraction, and Answer Ranking. Questions are analyzed to obtain question types, segments, focuses, and other limitations. Through a simple mapping table, question types are used to constrain possible answer types. Documents are segmented and indexed by both characters and words. After question analysis, we extract query terms from the question segments and construct queries from the terms to retrieve possible document passages, which are then sent to a named entity recognition system to obtain answer candidates. Finally, answers are ranked according to the needs of the question focuses.

The remainder of this paper is organized as follows. Section 2 describes the system architecture and techniques used in each module. In Section 3, we discuss the system's performance and results. Finally, we present our conclusions in Section 4.

2. System Description

Our system is comprised of four main components, as shown in Figure 1. Questions are first analyzed by the Question Processing module, and queries are constructed for Passage Retrieval. In the next phase, Answer Extraction is performed on the retrieved passages to obtain candidate answers, after which the Answer Ranking module is used to determine the top-ranked answers.

2.1. Question Processing

When ASQA receives a question, it is analyzed by the Question Processing module to obtain question segments, question types, question focuses, and other question limitations.

Chinese written texts do not contain word delimiters. Therefore, we incorporate a Chinese segmentation tool, CKIP AutoTag [8], to break a question into question segments comprising words and parts-of-speech (POS). With these question segments and other information, such as HowNet sense [2], we can identify 6 coarse-grained question types (PERSION, LOCATION, ORGANIZATION, ARTIFACT, TIME, and NUMBER) and 62 fine-grained question types. We adopt an integrated knowledge-based and machine learning approach for Chinese question classification.

We use InfoMap [3], which uses template rules to model Chinese questions as the knowledge-based approach, and adopt SVM (Support Vector Machines) [7] as the machine learning approach for a large collection of labeled Chinese questions.

Each question is classified into a question type(s) by the InfoMap and SVM module. The integrated module selects the question type with the highest confidence score from InfoMap or the SVM module.

Table 1. Taxonomy of Question Types.

Coarse-grained (6)	Fine-grained (62)
PERSON|人	APPELLATION|稱謂
	DISCOVERERS|發現者
	FIRSTPERSON|第一人
	INVENTORS|發明者
	OTHER|人其他類
	PERSON|人名
	POSITIONS|職位
LOCATION|地	ADDRESS|地址
	CITY|城市
	CONTINENT|大陸、大洲
	COUNTRY|國家
	ISLAND|島嶼
	LAKE|湖泊
	MOUNTAIN|山、山脈
	OCEAN|大洋
	OTHER|地其他類
	PLANET|星球
	PROVINCE|省
	RIVER|河流
ORG.|組織	BANK|中央銀行
	COMPANY|公司
	OTHER|組織其他類
	POLITICALSYSTEM|政治體系
	SPORTTEAM|運動隊伍
	UNIVERSITY|大學
ARTIFACT|物	COLOR|顏色
	CURRENCY|貨幣
	ENTERTAINMENT|娛樂
	FOOD|食物
	INSTRUMENT|工具
	LANGUAGE|語言
	OTHER|物其他類
	PLANT|植物
	PRODUCT|產品
	SUBSTANCE|物質
	VEHICLE|交通工具
	ANIMAL|動物
	AFFAIR|事件
	DISEASE|疾病
	PRESS|書報雜誌
	RELIGION|宗教
TIME|時間	DATE|日期
	DAY|日
	MONTH|月
	OTHER|時間其他類
	RANGE|時間範圍
	TIME|時間
	YEAR|年
NUMBER|數值	AGE|年齡
	AREA|面積
	COUNT|數字
	LENGTH|長度
	FREQUENCY|頻率
	MONEY|金額
	ORDER|序數
	OTHER|數值其他類
	PERCENT|比例
	PHONENUMBER|電話號碼、郵遞區號
	RANGE|數字範圍
	SPEED|速度
	TEMPERATURE|溫度
	WEIGHT|重量

Table 2. Examples of QFocus Analysis. All question focuses and limitations are in parentheses; "QF" means the question focus, "QFD" is the description of the question focus, "TI" represents time, and "NE" denotes the named entities in the sentence.

請問 [西元 2000 年 7 月/**TI**] [美方/**NE**] 派何人前往 [北京/**NE**] 對 TMD 以及其他全球戰略佈局與中方展開對話? July, 2000 USA Beijing *Who is the delegate of United States visiting Beijing to negotiate the TMD issue in July, 2000?*
請問 [2000 年/**TI**] 的 [G8 高峰會/**NE**] 在 [日本/**NE**] 何地舉行? Year 2000 G8 summit Japan *Which city in Japan hosted the G8 summit in 2000?*
請問 [芬蘭第一位女總統/**QF**] 爲誰? Finland's first woman president *Who is the Finland's first woman president?*
請問 [2000 年/**TI**] [沉沒於北極圈巴倫支海/**QFD**] 的 [俄羅斯核子潛艇/**QF**] 的名字? Year 2000 sank in the Barents Sea Russian nuclear submarine *Which Russian nuclear submarine sank in the Barents Sea in 2000?*
請問 [涉嫌竊取美國洛薩拉摩斯實驗室核武機密/**QFD**] 的 [華裔科學家/**QF**] 爲誰? accused of violating……National Laboratories Chinese scientist *Which Chinese scientist was accused of violating Atomic Energy Act because of his purportedly mishandling restricted data of Los Alamos National Laboratories?*

A detailed description of our question classification scheme can be found in [1].

In addition to question segments and types, we conduct question focus (QFocus) analysis to extract the question focus and other question limitations (QLimitations) to fully capture the main purpose of the question. The concept of QFocus analysis has been adopted by many QA systems since 1999 [4]. A QFocus is a word or phrase in a question that represents the answer, and is more informative than the question type. In our system, we use a maximum entropy model and some empirical rules to find the QFocus and QLimitations of questions, such as time, related named entities, and descriptions of the QFocus, which are helpful in finding the most appropriate answer.

Table 2 shows some manually annotated examples of QFocus analysis. All the examples in this paper are taken from the NTCIR-5 CLQA development set or test set. The second example has no QFocus, but has three QLimitations: one of time and two of named entities. In contrast, the third example has only one QFocus and no QLimitations.

2.2. Passage Retrieval

Documents are preprocessed in the Passage Retrieval module to remove noise, and then segmented by CKIP AutoTag [8] to obtain words and parts-of-speech (POS). We split documents into small passages using three punctuation marks ",!?" and index them by Lucene [9], an open source information retrieval engine. Two indices are used in ASQA. One is based on Chinese characters and the other on Chinese words. Since AutoTag considers the principle of compositionality, the segmentation results it derives have some short words that are not suitable for passage retrieval. We use heuristic rules, as shown in Table 3 and Table 4 to combine such words to make them more meaningful. These rules were inspired by CKIP's analysis of Chinese morphologies [6].

At runtime, we utilize the question segments and POS to form query terms. The word combination rules also apply to the question segments and some resulting compound words are filtered out according to a stop words list. The filtered words are then used as query terms to create Lucene queries. In the current version, we do not use any query expansion techniques to enlarge query terms.

Lucene provides many constructs for building queries; however, we only use the *boosting* (^) and *required* (+) operators in our system to emphasize the

Table 3. POS combination rules for two words; at least one word must be 1-char long

Left word	Right word	Composite
FW	Neu	FW
VH13	Na, Nb, Nc, Nd	Na

Table 4. POS combination rules for two 1-char-long words

Left word	Right word	Composite
Na, Nb, Nc	Na, Nb, Nc	Na
A, VH, Neu, Nes, VH13	Na, Nb, Nc, Nd	Na
VJ	VH	Na
VC, VD	Na	Na
Nba	Nba	Nba
Dfa	VH, V_2	VH13
Nes, Neu	Neu	Neu
Neu, Nes, FW	Nf	Nf
Neu	VH	Nf
Nep	Nf, Nd	Nf

differences between types of query term for answering factoid questions.

By observation, we classify query terms into three types, and obtain their weights empirically from the training data. The quoted terms of a question, which usually contain the most important information, are boosted by 2; general conditions represented by nouns are set to 1.2; and verbs and adjectives are scored at 0.7, because they seem to be optional hints and may have more synonyms.

Two Lucene queries are constructed for each question. All the query terms are connected and weighted via Lucene's *boosting* operator. In the initial query, quoted terms and nouns are set as *required*. If this query does not return a result, we retry a relaxed version of it that does not assign any query term as *required*.

Taking the question:「請問台灣童謠「天黑黑」是由哪位作曲家所創作？」(Who was the composer of the Taiwanese nursery rhyme "Dark Dark Sky?") as an example, the initial query and the relaxed query are constructed as follows:

Initial query:
+"作曲家"^1.2 +"台灣"^1.2 "創作"^0.7 +"童謠"^1.2 +"天黑黑"^2

Relaxed query:
"作曲家"^1.2 "台灣"^1.2 "創作"^0.7 "童謠"^1.2 "天黑黑"^2

Queries are sent to both the character-based and word-based indices. The passages derived from the indices are merged after removing duplicate passages. The merged passages are then sorted according to the scores given by Lucene, and the top five are sent to next module for answer extraction.

2.3. Answer Extraction

We perform a two-step answer extraction process. First, an online named entity recognition (NER) system is used to retrieve passages and obtain answer candidates. Second, the extracted named entities are filtered based on the expected answer types derived in the question processing phase.

We use a Chinese NER engine, Mencius [5], to identify both coarse-grained and fine-grained NEs. Mencius incorporates various types of linguistic features into maximum entropy (ME) models to discriminate coarse-grained NE types, such as PERSON, ORGANIZATION, and LOCATION. In addition, it encodes NE templates and NE lists in InfoMap to recognize fine-grained NEs and other coarse-grained NEs, such as TIME and NUMBERs.

We note that some NE types, like movie titles or TV program titles in Chinese news documents, are usually enclosed by quotation marks. Therefore, instead of employing the methods mentioned in the previous paragraph, we identify these NEs by quotation marks and classify them with nearby keywords.

After NER processing, the extracted named entities are filtered according to their NE categories to select answer candidates. To do this, we use a manually constructed mapping table containing information about question types and their corresponding expected answer types. NEs whose types are not found among the expected answer types are removed. The remaining NEs are the answer candidates. More information about expected answer types can be found in [1]. In addition to filtering with a mapping table, we remove stop words from a question and send the remainder to Internet search engines to obtain highly coherent sentences. Answer candidates not contained in the retrieved sentences are eliminated.

2.4. Answer Ranking

In the answer ranking phase, we use QFocus and QLimitations to sort the answer candidates derived from the Answer Extraction module. An answer candidate is given a ranking score if it fits the answer focus or limitations of the question. The candidate with the highest score is the one that fits the most clues of the question, and is therefore regarded as the top 1 answer to the question. The ranking score of answer candidate a_{ij} in passage p_i is calculated as follows :

$$\text{Score}(a_{ij}) = \frac{\sum_{k=1}^{m}\text{Exist}(p_i, ne_k)}{NE\_Number} + \frac{\sum_{l=1}^{o}\text{Exist}(p_i, cue_l)}{CUE\_Number}$$

$$+\text{QFI}(a_{ij}) + \text{QFA}(a_{ij}),$$

where

- p_i is the selected passage and a_{ij} is the *j-th* answer candidate extracted from p_i ;
- $NE = \{ne_1, ne_2, \cdots, ne_m\}$ is the named entity set appearing in the question;
- $CUE = \{cue_1, cue_2, \cdots, cue_o\}$ are other question limitations, except named entities.
- $\text{Exist}(p_i, ne_k) = \{1,0\}$, which represents the matching bonus score of related named entities. If the source passage p_i of answer candidate a_{ij} contains $ne_k \in NE$, then $\text{Exist}(p_i, ne_k) = 1$; otherwise $\text{Exist}(p_i, ne_k) = 0$.
- $\text{Exist}(p_i, cue_l) = \{1,0\}$, which is the answer cue's matching bonus score. The calculation of $\text{Exist}(p_i, cue_l)$ is similar to that of $\text{Exist}(p_i, ne_k)$.

- NE_Number and CUE_Number are the number of named entities in NE and the number of cues in CUE respectively.
- QFI(a_{ij}) indicates the extra score if answer candidate a_{ij} comprises the question focus string.
- QFA(a_{ij}) indicates the extra score if answer candidate a_{ij} is adjacent to the question focus string.

3. Performance and Error Analysis

In the CLQA C-C Task, we achieved 37.5% and 44.5% overall accuracy for correct answers, and correct+unsupported answers, respectively.

We observe that the distribution of question types in the development set of questions is different from the distribution in the test set of questions. The largest variation is in the number of ARTIFACT questions. There are 10% more questions about ARTIFACT in the test set than in the development set, as shown in Table 5. Although the type distribution in the two sets does not have to be the same, a large variation in distribution is not a practical way to evaluate a system's performance, since each system is developed according to a problem's specific characteristics.

Table 5 Question type distribution of development set questions and test set questions. Some types in the development set are combined.

QType	(D)ev. Set	(T)est Set	(T) – (D)
ARTIFACT	16.5%	6.5%	-10.0%
LOCATION	27.5%	26.5%	-1.0%
MONEY	1.0%	2.0%	1.0%
NUMEX	3.5%	6.0%	2.5%
ORG.	10.0%	9.0%	-1.0%
PERSON	35.0%	40.0%	5.0%
DATE	6.0%	10.0%	4.0%
REASON	0.5%	0%	-0.5%

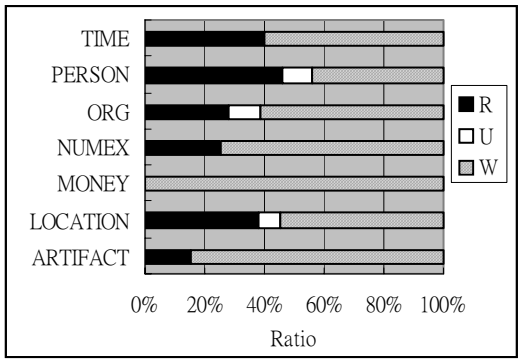

Figure 3. Accuracy of QA system by question type. R: correct answers, U: unsupported answers, W: wrong answers

The performance of each question type is shown in Figure 3. We observe the variation in accuracy between question types. PERSON is the best type with 56% R+U accuracy, and MONEY is the worst type with all answers wrong. The high F-measure of our NER module, shown in Figure 2, is the major factor in the performance of the PERSON type. Furthermore, we only return the top 5 passages and do not consider question type information in the passage retrieval phase; thus, rare answer types, such as MONEY and NUMEX, are often not included in the top passages. We will add a passage filter later to overcome this problem.

A strange phenomenon can be observed in Figure 3, Only three types, PERSON, ORGANIZATION, and LOCATION, have unsupported answers. Although the reason is unclear, we think it could be due to the high degree of accuracy of the NER module. The complex relationship between the concepts of these types in news documents may be another reason for this phenomenon.

We made another interesting observation when comparing our system with other C-C Task participants. Using our approach, the percentage of unsupported answers was 7%, which was approximately double that of other participants. The results of other QA contests, such as TREC QA Track [13], reveal the same phenomenon, which suggests there is a pattern that requires further study.

As well as examining the system's performance based on question types, we also studied the performance of each module.

Question classification is important in QA systems. The accuracy of our question classification module [1] for InfoMap and SVM is 88% and 73.5% respectively. By combining the two methods, 92% accuracy can be achieved in question type classification.

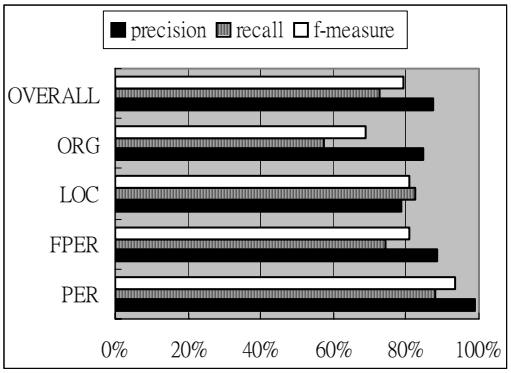

Figure 2. NER overall performance, plus the performance of organization names, location names, person names and foreign person names

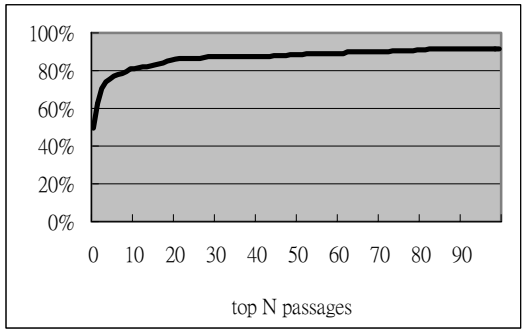

Figure 4. TopN passage retrieval accuracy. Both correct and unsupported answers are included.

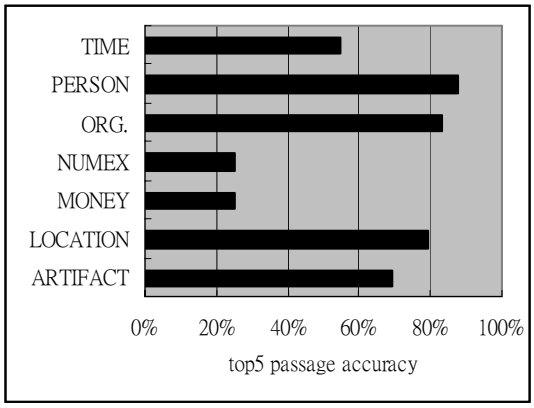

Figure 5. Top5 passage retrieval accuracy by QTypes; both correct and unsupported answers are included.

In passage retrieval, we use an open source IR engine without query expansion. Even so, the passage retrieval performance is acceptable. As Figure 4 shows, the passage accuracy of the Top5 passages we use in our system is approximately 80%. If we extend the number of passages to 100, 92% passage accuracy can be achieved. Comparing Figure 3 and Figure 5, the performance variance of passage retrieval between question types is not as large as the overall system performance. PERSON is still the best performing type in this module.

Since we use question types along with the NE types of answers to eliminate the number of answer candidates, the performance of the NER module is crucial. Errors in NE type identification may cause unexpected removal of correct answers. We have therefore improved the Mencius NER engine so that it has a better recognition rate than the previous version [5]. The performance of coarse-grained NE types is shown in Figure 2. We achieve 93%, 81%, and 69% F-measure scores for the PERSON, LOCATION, and ORGANATION types respectively. These scores seem to be proportional to the QA system performance shown in Figure 3. This suggests that further development of the NER module would enhance the QA system.

4. Conclusion

Question answering systems are extremely complex, but their potential is unlimited. This year, we tried to minimize the cost and complexity by using available modules, such as our own Mencius, InfoMap, CKIP's AutoTag, and the Lucene module. By modifying these modules slightly and adding two critical parts, question classification and answer ranking, we created the Academia Sinica Question Answering System for the NTCIR CLQA C-C Task.

In addition, various machine-learning methods and manually collected or created data were incorporated, achieved 37.5% and 44.5% Top1 accuracy for correct and correct+unsupported answers respectively. We believe that machine-learning methods are just tools; human knowledge or rules are the key to creating intelligent systems.

It is very helpful to have a forum to compare the performance of various Asian language QA systems. We hope that it will be a regular event so that more researchers can share their experience and expand this research domain further.

5. Acknowledgments

This research was supported in part by the National Science Council under GRANT NSC94-2752-E-001-001-PAE.

We would like to thank the Chinese Knowledge and Information Processing group (CKIP) in Academia Sinica for providing us with AutoTag for Chinese word segmentation.

6. References

[1] M.-Y. Day, C.-W. Lee, S.-H. Wu, C.-S. Ong, W.-L. Hsu. An Integrated Knowledge-based and Machine Learning Approach for Chinese Question Classification. In *Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLPKE)*, 2005.

[2] Z. Dong and Q. Dong, HowNet, http://www.keenage.com/, 2000.

[3] W.-L. Hsu, Y.-S. Chen, S.-H. Event Identification Based on the Information Map – INFOMAP. In *Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLPKE)*, 2001.

[4] D. Moldovan, S. Harabagiu, M. Paşca, R. Mihalcea, R. Goodrum, R. Gîrju and V. Rus. Lasso: A Tool for Surfing the Answer Net. In *Proceedings of The Eighth Text REtrieval Conference*, 1999.

[5] T.-H. Tsai, S.-H. Wu, C.-H. Lee, C.-W. Shih,

W.-L. Hsu. Mencius: A Chinese Named Entity Recognizer Using Maximum Entropy-based Hybrid Model. In *Computational Linguistics & Chinese Language Processing* 9, 65-82. 2004.

[6] H.-H. Tseng, K.-J. Chen, Design of Chinese Morphological Analyzer. In *Proceedings of SIGHAN*, 2002.

[7] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.

[8] CKIP AutoTag, Academia Sinica. http://ckipsvr.iis.sinica.edu.tw/

[9] Lucene, http://lucene.apache.org/

[10] Text REtrieval Conference (TREC), http://trec.nist.gov/

[11] Cross Language Evaluation Forum (CLEF), http://www.clef-campaign.org/

[12] NTCIR Workshop, http://research.nii.ac.jp/ntcir/

[13] TREC 2004 Question Answering Results, http://trec.nist.gov/pubs/trec13/appendices/qa.results.html