NTT's CCLQA System for NTCIR-7 ACLIA

Ryuichiro Higashinaka and **Hideki Isozaki**
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Kyoto 619-0237, Japan
{rh,isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper describes our Complex Cross-Lingual Question Answering (CCLQA) system based on the technologies used in our past NTCIR systems for QAC and CLQA. We implemented a new rule-based English question analyzer to extract English query terms, which are translated into Japanese by translation dictionaries. For DEFINITION, BIOGRAPHY, and EVENT questions, we reused our definition module for QAC-4. For RELATIONSHIP questions, we developed a new module based on our why-QA approach for QAC-4. When these modules were not applicable, a simple sentence retriever was used. According to the organizers' evaluation results, although our EN-JA system performed rather poorly due to the low coverage of the translation dictionaries, our JA-JA system achieved the second best score among the four participants.

1 Introduction

Question answering (QA) has been extensively studied in the last decade. Although in the beginning of QA research most systems aimed to answer simple questions that request names (person names, location names, etc.) and numerical expressions (e.g., date, length etc.), the focus is now moving towards answering more complex questions that request definitions, relations, causes/reasons, procedures, and so forth.

This paper describes our Complex Cross-Lingual Question Answering (CCLQA) system for the NTCIR-7 Advanced Cross-lingual Information Access (ACLIA) task where the question types dealt with are DEFINITION, BIOGRAPHY, RELATIONSHIP, and EVENT. The task requests that English questions be answered by Japanese answers. Figure 1 shows the architecture of our system.

In building our system, our principle emphasized reusing most of our existing resources so that we could find modules that need improvement for CCLQA. Based on this principle, we reused the English question analyzer and the English-to-Japanese translation dictionaries we used for CLQA-1 [6]. The English question analyzer was extended to cope with CCLQA questions because only factoid questions were considered at CLQA-1.

For BIOGRAPHY, DEFINITION, and EVENT questions, we retrieved documents by Lucene with Okapi/BM25 and reused the definition answering module developed for QAC-4 [3]. For RELATIONSHIP questions, we developed and built a new answering module in the same manner as our why-question answering module [4]. For questions that cannot be classified, we simply used Lucene that returns a ranked list of single sentences based on Okapi/BM25. Since these answering modules work on Japanese, the input for them must be translated beforehand by the question analyzer and the translation dictionaries.

In Section 2, we describe how we translate English questions using the English-to-Japanese translation dictionaries. In Sections 3 and 4, we describe our answering modules for DEFINITION/BIOGRAPHY/EVENT and RELATIONSHIP questions. In Section 5, we describe our system's formal run results and possible improvements.

2 Dictionary-based Translation

For BIOGRAPHY, DEFINITION, and EVENT questions, target entities (hereafter, *targets*), such as names of persons, organizations, or events, must be extracted. For RELATIONSHIP questions, the two entities being related must be identified. To extract them, we used hand-crafted rules. Once targets/entities have been identified, we translate them using translation dictionaries. The dictionaries used are EDICT, ENAMDICT, and an in-house E/J dictionary developed for NTT's traditional machine translation system, ALT-J/E.

Although we are currently making progress with statistical machine translation (SMT) [11, 10], here, we used the dictionary-based method for its simplicity. We plan to introduce SMT techniques in the future.

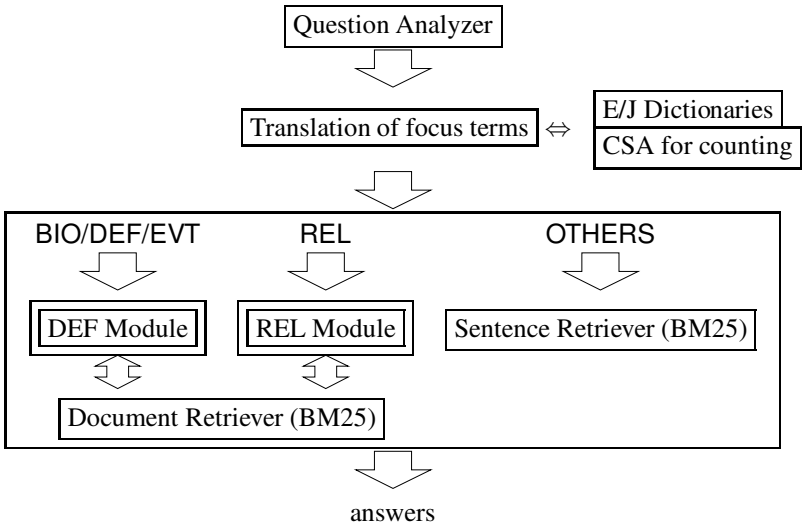

Figure 1. E/J System Architecture

2.1 Translating Targets

We suppose a target has a maximum of three words. When it has more than three words, we only focus on the initial three words.

When the target has three words (A, B, and C), we first consult the translation dictionaries to see whether they have such a three-word entry ("A B C"). If an entry is found, its translations are retrieved. If the look-up fails, we check whether the dictionaries contain the initial two words ("A B") as an entry. If this also fails, we check whether entry "A" exists in the dictionaries. When we find an entry for "A", we retain the translations for "A" and then look-up in the dictionaries using the longest match principle for the remaining words; that is, "B C" and then "B" if "B C" fails.

At the end of this process, we have n ($1 \leq n \leq 3$) parts of the target with multiple translation alternatives. We then compute all possible combinations of the translation alternatives. For example, when "A B C" is separated into two parts, "A B" and "C", with their alternatives $\{AB_1, AB_2, \ldots, AB_n\}$ and $\{C_1, C_2, \ldots, C_m\}$, we have $n \times m$ translations. The combinations are sorted by their frequency order from Mainichi newspaper articles (1998–2001).

To obtain the frequency of an arbitrary string efficiently, we used a variant of *Compressed Suffix Array* (CSA) [9]. CSA is also useful to check unknown words. In Japanese, inter-word spaces are not used. When the morphological analyzer does not know a certain word in a sentence, it often fails to determine word boundaries, leading to failures in commonly-used word-based inverted index search engines. CSA is robust in this sense because it performs a character-based search.

3 Definition Module

For DEFINITION, BIOGRAPHY, and EVENT questions, we reused our DEFINITION module for QAC-4 with a slight modification. In QAC-4, we used a simple pattern-based approach for definition questions. For example, we had "Y such as X" and "$Y(X)$" as our patterns, where X is the target and Y is the definition phrase to be extracted. Since we observed that most definitions we retrieved were obtained from a single pattern that extracts *rentai* (adnominal modification) or *renyou* (adverbial modification) clauses depending on X, we simply used this single pattern for answer extraction. In QAC-4, Harada et al. [2] and Murata et al. [8] also extracted modifying phrases as answer candidates.

For example, we obtained the following answer candidates for "President Suharto (スハルト大統領)":

- 32 年間にわたって，人口約 2 億人のインドネシアを牛耳ってきたスハルト大統領 (President Suharto, who has been ruling Indonesia with the population of 200 million for 32 years)
- 人口 2 億人を抱える大国で，32 年間も指導者であるスハルト大統領 (President Suharto, who has been the ruler of a major nation of over 200 million people for 32 years)

To extract adnominal or adverbial clauses, we used a tree-based search program called Tgrep2[1]. We first parsed the target using CaboCha[2] and made a dependency tree for the target (e.g., "スハルト (Suharto)" depending on "大統領 (President)"). Then, all subtrees containing the dependency tree were retrieved

[1] http://tedlab.mit.edu/~dr/TGrep2/index.html
[2] http://chasen.org/~taku/software/cabocha/

from Mainichi newspaper articles that had been converted into dependency trees using CaboCha in advance. To quicken the search process, we first identified document IDs containing the target using Lucene and only searched through the dependency trees of the documents for the IDs.

For answer candidate evaluation, we used the sum of the scores of content words in C:

$$\mathrm{candscore}_{\mathrm{def}}(C) = \sum_{w \in \mathrm{CW}(C)} \mathrm{wordscore}_{\mathrm{def}}(w),$$

where $\mathrm{CW}(C)$ is the set of content words (verbs, nouns, and adjectives) in C.

It is reasonable to consider that a content word shared by many answer candidates indicates a better definition than another word shared by only a few candidates. Therefore, we defined the word score by the log of the count (term frequency without normalization) of word w in the set of all candidates $\{C_i\}$ found by Tgrep2:

$$\mathrm{wordscore}_{\mathrm{def}}(w) = \log(\mathrm{tf}(w; \{C_i\})).$$

4 Relationship Module

For RELATIONSHIP questions, we took the same approach as our why QA approach [4], which automatically mines patterns regarding a certain semantic relation (e.g., CAUSE) from a semantically tagged corpus, such as the EDR corpus[3], and uses the patterns to rank answer candidates. This approach resembles [1], which uses automatically acquired *soft-patterns* for ranking.

When the two entities whose relation we want to find are input to this module, they are first concatenated by a Japanese particle "と (and)" to create a search query. Then, using this query, we turned to Lucene to retrieve the top-20 documents. We used all the sentences in the retrieved documents as answer candidates. For each answer candidate, we calculated two kinds of scores: the *relation score* and the *similarity score*.

To obtain the relation score, we first created a classifier that distinguishes sentences expressing relationships from others by the following steps:

- We tagged each sentence in the EDR corpus with Goi Taikei's [5] semantic categories (word senses) and these sentences were transformed into tree structures in the same manner as [4].
- By using the semantic categories, each sentence in the corpus was classified into two classes depending on whether the sentence has one of the semantic categories corresponding to the word "関係 (relationship)".
- Then we removed these semantic categories from the trees, applied the training program of BACT, a tree-based boosting algorithm [7], and obtained a BACT classification model. We removed the categories from the training data to obtain implicit patterns that describe relationships.

We used BACT's output for answer candidate C as its relation score ($\mathrm{score}_{\mathrm{rel}}(C)$).

For the similarity score, we used the coverage of inverse document frequency (IDF) scores:

$$\mathrm{score}_{\mathrm{sim}}(C) = \frac{\sum_{w \in \mathrm{CW}(Q) \cap \mathrm{CW}(C)} \mathrm{idf}(w)}{\sum_{w \in \mathrm{CW}(Q)} \mathrm{idf}(w)},$$

where CW is a function that returns content words for question Q and answer candidate C. The overall score of C is simply the sum of the two scores:

$$\mathrm{score}(C) = \mathrm{score}_{\mathrm{rel}}(C) + \mathrm{score}_{\mathrm{sim}}(C).$$

5 Results and Possible Improvements

Table 1 shows our system's performance. Our JA-JA system was the second best among the four systems. However, our EN-JA system had difficulty answering many questions. Such poor performance was caused by the fragility of the English question analysis and dictionary-based translation modules. Here, we describe how we could improve these modules.

The system failed to solve the next question due to the fragility of the translation module.

- ACLIA1-JA-T10: "Please tell me about Martina Navratilova."

The translation module consulted the translation dictionary for "Martina Navratilova" and obtained "マルティナナヴラティロワ" as its translation. However, CSA rejected this translation because no such string appears in the given Japanese documents. Therefore, the system failed to answer this question.

After the formal run, we added a fail-safe mechanism to avoid this situation. Currently, the system tries to find other translations using the combination of words in the target. For example, each word of "Martina Navratilova" has the following translations:

- Martina: マルチナ, マーティナ, マルティナ
- Navratilova: ナブラチロワ, ナヴラティロワ

Then, the translation module checks all combinations of these and can successfully obtain "マルチナ・ナブラチロワ". Figure 2 shows the term frequencies given by CSA.

In this way, we can solve most of the failures. Sometimes, however, the dictionary does not have an appropriate entry. For example, the system failed to solve the next question because there is no entry for "Suharto".

[3] http://www2.nict.go.jp/r/r312/EDR/index.html

Table 1. Scores of our system

answer class	DEFINITION	BIOGRAPHY	RELATIONSHIP	EVENT	ALL
our JA-JA system's score	0.2888	0.1788	0.2209	0.0915	0.1873
best score of other JA-JA systems	0.4201	0.1900	0.2332	0.0937	0.2201
our EN-JA system's score	0.1699	0.0932	0.0476	0.0023	0.0676

Table 2. Term frequencies given by CSA

マルチナ	332	マルチナナブラチロワ	0
マーティナ	0	マルチナ・ナブラチロワ	18
マルティナ	9	マルティナナブラチロワ	0
ナブラチロワ	39	マルティナ・ナブラチロワ	0
ナヴラティロワ	0		

- ACLIA1-JA-T3: "Please tell me about President Suharto."

We can obtain the correct translation "スハルト" for "Suharto" easily by modifying our *back-transliteration module* that we introduced for CLQA J-E[6]. We have not implemented an E-J transliteration module yet, but the J-E back-transliteration module returned "Suharto" as the second best candidate for "スハルト". The best candidate is "Soeharto", which is an alternative transliteration of Suharto.

Even the J-E transliteration module is useless for new expressions such as "embryonic stem cells" in the following question because its translation, "ES 細胞", cannot be derived from transliteration.

- ACLIA1-JA-T106: "I would like to know about the relationship between stem cells and embryonic stem cells."

For such cases, we must obtain new words and their translations from multilingual resources such as Wikipedia.

6 Concluding remarks

This paper described our Complex Cross-Lingual Question Answering (CCLQA) system based on the technologies used in our past NTCIR systems for QAC and CLQA. Our JA-JA system showed relatively good performance, but our EN-JA system did not work as well due to the fragility of the English question analysis and E-J translation modules. We discussed possible improvements for these modules.

References

[1] H. Cui, M.-Y. Kan, and T.-S. Chua. Soft pattern matching for definitional question answering. *ACM Transactions on Information Systems (TOIS)*, 25(2), 2007.

[2] M. Harada, Y. Kato, K. Takehara, M. Kawamata, K. Sugimura, and J. Kawaguchi. QA system Metis based on semantic graph matching at NTCIR-6. In *Working Notes of the NTCIR Workshop Meeting (NTCIR)*, pages 448–459, 2007.

[3] R. Higashinaka and H. Isozaki. NTT's question answering system for NTCIR-6 QAC-4. In *Working Notes of the NTCIR Workshop Meeting (NTCIR)*, pages 460–463, 2007.

[4] R. Higashinaka and H. Isozaki. Automatically acquiring causal expression patterns from relation-annotated corpora to improve question answering for why-questions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 7(2), 2008.

[5] S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Oyama, and Y. Hayashi, editors. *Nihongo Goi Taikei – A Japanese Lexicon*. Iwanami Publishing, 1997. (in Japanese).

[6] H. Isozaki, K. Sudoh, and H. Tsukada. NTT's Japanese-English Cross-Language Question Answering System. In *Working Notes of the NTCIR Workshop Meeting (NTCIR)*, pages 186–193, 2005.

[7] T. Kudo and Y. Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 301–308, 2004. http://chasen.org/˜taku/software/bact/.

[8] M. Murata, S. Tsukawaki, T. Kanamaru, Q. Ma, and H. Isahara. A system for answering non-factoid japanese questions by using passage retrieval weighted based on type of answer. In *Working Notes of the NTCIR Workshop Meeting (NTCIR)*, pages 477–482, 2007.

[9] K. Sadakane. Compressed text databases with efficient query algorithms based on the compressed suffix array. In *Lecture Notes in Computer Science*, volume 1969, pages 410–421, 2000.

[10] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki. Online large-margin training for smt. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Lanugage Learning (EMNLP/CoNLL)*, 2007.

[11] T. Watanabe, H. Tsukada, and H. Isozaki. Left-to-right target generation for hierarchical phrase-based translation. In *Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association of COmputational Linguistics (COLING/ACL)*, 2006.