Question – Answering Using Pronoun Disambiguation

Omdev Nagar
Department of Computer Science, YMCA University of science and technology, Faridabad & Haryana, India.

Dr. Sonali Gupta
Department of Computer Science, YMCA University of science and technology, Faridabad & Haryana, India.

Abstract –The Question Answering framework may be playing a huge part in internet searcher Also majority of the data extraction standard. The Question Answering System is mainly used to retrieve the exact answers from the database for the user query. The user query may be in the form of structured or unstructured manner. The research QA system has been started in the period of 1960s and they concentrated only on implementing the domain specific knowledge extraction [1],[2].

Index Terms – Question Answering System, QA, Pronoun Disambiquation

1. INTRODUCTION

The Question Answering framework may be playing a huge part in internet searcher Also majority of the data extraction standard. The Question Answering System is mainly used to retrieve the exact answers from the database for the user query. The user query may be in the form of structured or unstructured manner. The research QA system has been started in the period of 1960s and they concentrated only on implementing the domain specific knowledge extraction [1] , [2]. Researchers explored the research on QA system based on different categories like closed, pre-tagged [3] and Knowledge bases [4] , [5]. Jeeves [6] is the web based interface which accepts the user query in natural language and retrieve the information. This interface has the additional feature like recommending the list of relevant queries to the user. The user selects one of the matched queries from the list. Generally, the Question Answering system accepts the user query in natural language and retrieves the answer from the system. Earlier, the system followed the approach that permit the keywords as the query and find the matched answer based on the keywords.. The Word Sense Disambiguation is the task of identifying the correct sense of a polysemous word in a given input text. Words in English language can have completely opposite connotations in different contexts. For example, consider the following sentences: The album includes a few instrumental pieces. [Sense: relating to or designed for or performed on musical instruments} His efforts have been instrumental in solving the problem. [Sense: serving or acting as a means or aid} Both the above sentences convey a different meaning and context for the word 'instrumental'. The first sentence points at the sense which is related to musical instruments and the second sentence points at the sense - "serving or acting as a means or aid". Therefore for every word it is crucial to identify the correct sense it is pointing at according to the context of the text in which it is being used.

A knowledge based approach to Word Sense Disambiguation requires an external lexical database to specify the senses which need to be disambiguated. In this case we use, WordNet [14], which is a computational linguistics tool. It groups nouns, verbs, adjectives and adverbs into sets of cognitive synonyms each expressing a distinct concept. WordNet is a network of words which are interlinked by semantic and lexical relations. Many of the previous knowledge based approaches in the area of word sense disambiguation have used WordNet as their knowledge source.

2. RELATED WORK

2.1Question Processing

The Question Processing results are a list of keywords plus the information for asking point. For example, the question:

[1] Who won the 1998 Nobel Peace Prize? contains the following keywords: won, 1998, Nobel, Peace, Prize. The asking point Who refers to the NE type person. The output before question expansion is a simple 2-feature template as shown below:

[2] asking_point: PERSON

key_word: { won Nobel, Peace Prize in 1998 }

The following example shows the asking point does not correspond to any type of Named Entity in our definition.

[3] Why did Indrani ask the CBI for a word processor ?

The system then maps it to the following question template:

[4] asking_point: REASON

key_word: { ask, Indrani, CBI, word, processor }

The question is scanned by the question processor to search for question words (wh-words) and maps them into corresponding NE types/sub-types or pre-defined notions like REASON.

We adopt two sets of pattern matching rules for this purpose: (i) structure based pattern matching rules; (ii) simple key word based pattern matching rules (regarded as default rules).

It is fairly easy to exhaust the second set of rules as interrogative question words/phrases form a closed set.

In comparison, the development of the first set of rules are continuously being fine-tuned and expanded. This strategy of using two set of rules leads to the robustness of the question processor. The first set of rules are based on shallow parsing results of the questions, using Cymfony FST based Shallow Parser. The following is a sample of the first set of rules:

[5] Name NP (city | country | company) -->

CITY | COUNTRY | COMPANY

[6] Name NP (person_w) --> PERSON

[7] Name NP (org_w) --> ORGANIZATION

[8] Name NP (NOT person_w, NOT org_w) --> NAME

Rule [5] is used to check head words of the Noun Phrase.. Rule [6] works for cases like VG[Name] NP[the first private citizen] VG[to fly] PP[in space] as citizen belongs to the word class person_w. Rule [8] depicts the catch-all rule: if the Noun Phrase is person_w or org_w, then the asking point is a proper name (default NE), often realized in English in capitalized string of words. Examples include Name a film that has won the Golden Bear in the Berlin Film Festival.

The word lists org_w and person_w are currently manually maintained based on inspection of large volumes of text. An effort is underway to automate the learning of such word lists by utilizing machine learning techniques.

Shallow parsing helps us to capture a variety of question expressions. However, in some cases where simple key word based pattern matching would be enough to capture the asking point. That is our second set of rules. These rules are used when the first set of rules has failed to produce results. The following is a sample of such rules:

[9] who/whom --> PERSON

[10] when --> TIME/DATE

[11] where/what place --> LOCATION

[12] what time (of day) --> TIME

[13] what day (of the week) --> DAY

[14] what/which month --> MONTH

[15] what age/how old --> AGE

[16] what brand --> PRODUCT

[17] what --> NAME

[18] how far/tall/high --> LENGTH

[19] how large/hig/small --> AREA

[20] how heavy --> WEIGHT

[21] how rich --> MONEY

[22] how often --> FREQUENCY

[23] how many --> NUMBER

[24] how long --> LENGTH/DURATION

[25] why/for what --> REASON

asking_point: {because | because of | due to | thanks to | since | in order | to VB}

Key_word: { ask | asks | asked | asking | Indrani | CBI | word | processor }

The last item find an infinitive by checking the word to followed by a verb, as we know infinitive verb phrases are often used in English to explain a reason for some action.

2.2 Text Processing

In this ,we initially send the inquiries straightforwardly to an internet searcher keeping in mind the end goal to limit the archive pool to the main n, say 300, reports for IE preparing. Currently, this includes tokenization, POS tagging and NE tagging. Future plans include several levels of parsing as well; these are required to support CE and GE extraction. It should be noted that all these operations are extremely robust and fast, features necessary for large volume text indexing. Parsing is accomplished through cascaded finite state transducer grammars.

2.3 Text Matching

In This We attempt to match the question template with the processed documents for both the asking point and the keywords. There is a preparatory positioning standard built-in the matcher in order to find the most plausible answers. The essential rank is a check of what number of special watchwords are contained within a sentence. The auxiliary positioning is based on the order that the keywords appear in the sentence compared to their order in the question. The third, ranking is based on whether there is an correct match or a variant match for the key verb.

3. LIMITATION

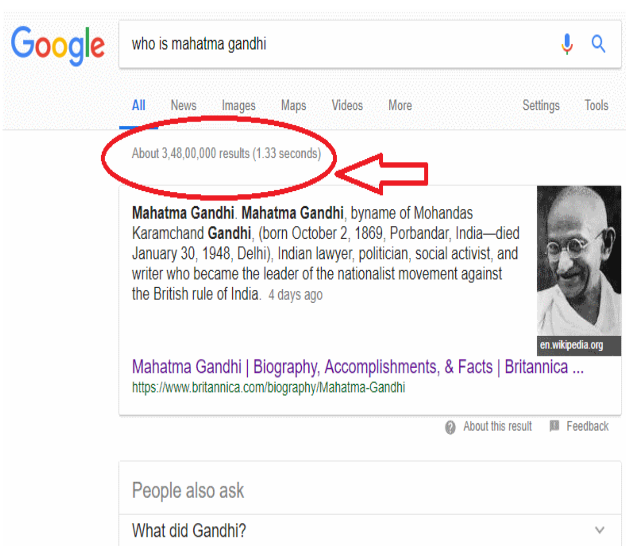
The first limitation comes from the types of questions. Currently only wh-questions are handled although it is planned that yes-no questions will be handled once we introduce CE and GE templates to support QA. Among the wh-questions, the why-question and how-question are more challenging because the asking point cannot be simply mapped to the NE types/sub-types.

The second limitation is from the nature of the questions. Questions like Where can l find the homepage for Oscar winners or Where can I find info on Shakespeare's works might be answerable easily by a system based on a well-maintained data base of home pages. Since our system is based on the processing of the underlying documents, no correct answer can be provided if there is no such an answer (explicitly expressed in English) in the processed documents. However, in the real world scenario such as a QA portal, it is conceived that the IE results based on the processing of the documents should be complemented by other knowledge sources such as e-copy of yellow pages or other manually maintained and updated data bases. The third limitation is the lack of linguistic processing such as sentence-level parsing and cross-sentential co-reference (CO). This problem will be gradually solved when high-level IE technology is introduced into the system.

4. PROPOSED WORK

When user search result on google, there they will have lots of pages for single search. Like if we consider this example ::

So as we can see here 3,48,000 total results found. So users need to decide which page to visit.Due to that users Will have to look so many results, and out of that so many of them are waste of time and of no use. So keep this thing in mind we did this work.We matching user query from our local corpus and responding to their question and generating answer in short .



5. IMPLEMENTATION

These three main modules are that are used to process answer with respect to question are:

1. Query Processing module
2. Document Processing module (information retrieval)
3. Answer Processing module

1. Question Processing Module:

First of all when user put any query based on question form. The query specifies the keywords that should be used for the IR system to use in searching for documents. For the given query as input, the function of query processing module is to analyze the query and then tokenize the given question and process by creating some representation in some format for the information required. So query processing module must have to do.

2. Document Processing module:

In this phase, tokens will be collected that were generated in question processing phase for given user question. After tokens will be collected, we match those tokens from our data, and represent the best suited answer corresponding to that token.

3. Answer Processing module:

The final phase of this architecture is Answer Processing module which is responsible for recognizable proof, extraction and approval of answers from set of generated ordered paragraph received from document processing module. In this phase, result will be generated based on user question and finally that answer will be shown to user.

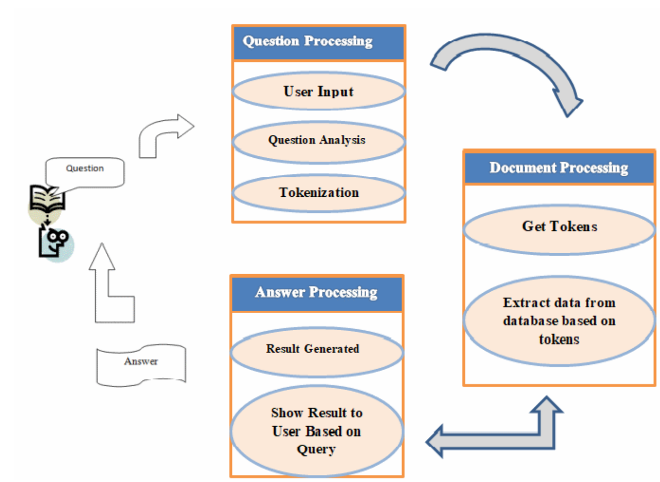


Fig -1: Answer Processing Diagram

After text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this made record, feature every one of the substance and import your readied content document. You are presently prepared to style your paper.

6. CONCLUSIONS AND FUTURE SCOPE

The PDP (Pronoun Disambiguation Problem) task doesn't have direct training data. By applying state-of-the-art co-reference resolvers to answer the PDP problems, we find the performances are very poor (close to a random guess). So by Generating small queries system performance will increase. First we tokenize user question into tokens, After that tokens match content from our database. After matching tokens with our database content , it produces result in Quick time.

In future we are planning to get answer based on speech recognisation. When user writes text it will consume some of time, So better that user find their answers by speeking only. It will reduced even more time to generate answer based on user query.

REFERENCES

[1] Z. Zheng, "AnswerBus Question Answering System," In Proceedings of Second International Conference on Human Language Technology Research, San Francisco, CA, USA, 2002.

[2] C. T. Kwok, O. Etzioni, and D. S. Weld, "Scaling Question Answering to the Web," In Proceedings Tenth World Wide Web Conference, Hong Kong, China, 2001.

[3] S. Harabagiu, M. Pasca, and S. Maiorano, "Experiments with open-domain textual question answering," In Proceedings of 18th International Conference on Computational Linguistics, 2000.

[4] J. Budzik and K. J. Hammond, "Learning for Question Answering and Text Classification: Integrating Knowledge-Based and Statistical Techniques," In Proceedings of AAAI Workshop on Text Classification. Menlo Park, CA, 1998.

[5] P. Clark, J. Thompson, and B. Porter, "A knowledge-based approach to question answering," In Proceedings of AAAI'99 Fall Symposium, Orlando, Florida. 1999.

[6] Askjeeves:http://askjeeves.com/2000

[7] Seonyeong Park, Soonchoul Kwon, Byungsoo Kim, Sangdo Han," Question Answering System using Multiple Information Source and Open Type Answer Merge", Proceedings of NAACLHLT 2015, pages 111–115,Denver, Colorado, May 31 – June 5, 2015.

[8] Wang, C., M. Xiong, Q. Zhou and Y. Yu, 2007. PANTO: A portable natural language interface to ontologies. Proceedings of the 4th European Semantic Web Conference, (ESWC' 07), Publication post of DBLP, pp: 473-487.

[9] Damljanovic, D., M. Agatonovic and H. Cunningham,2010. Natural language interfaces to ontologies:Combining syntactic analysis and ontology-based lookup through the user interaction. Semantic Web:Res. Appli., 6088: 106-120. DOI: 10.1007/978-3- 642-13486-9-8.

[10] Kaufmann, E., A. Bernstein and R. Zumstein, 2006.Querix: A natural language interface to query ontologies based on clarification dialogs.Proceedings of the 5th International Semantic Web Conference, (ISWC' 2006), Citeulike, pp: 980-981.

[11] Cimiano, P., P. Haase, J. Heizmann, M. Mantel and R.Studer, 2007. Towards portable natural language interfaces to knowledge bases-the case of the ORAKEL system. Data Know. Eng., 65: 325-354. DOI: 10.1016/j.datak.2007.10.007

[12] Hogan, A., A. Harth, J. Umbrich, S. Kinsella and A.Polleres et al., 2011. Searching and browsing linked data with SWSE: The semantic websearch engine. J. Web Semantics, 9: 365-401. DOI: 10.1016/j.websem.2011.06.0

[13] Toba, Hapnes, et al. "Discovering high quality answers in community question answering archives using a hierarchy of classifiers." Information Sciences 261 (2014): 101-115.

[14] C. Fellbaum, editor, "Word Net: An Electronic Lexical Database," MIT Press 1998.

[15] Michael Lesk, "Automatic Sense Disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone", Proceedings of SIGDOC'86. 1986

Authors

Omdev Nagar
M.Tech, Dept. of CSE
YMCAUST, Faridabad
Om581895@gmail.com.

Dr. Sonali Gupta
Assistant Professor
Department of CSE
YMCAUST, Faridabad
sonali.goyal@yahoo.com