University of Lethbridge's Participation in TREC-2007 QA Track

Yllias Chali and Shafiq R. Joty
Department of Mathematics and Computer Science
University of Lethbridge
4401 University Drive
Lethbridge, Alberta, Canada, T1K 3M4
E-mail: {chali,jotys}@cs.uleth.ca

1 Introduction

Question Answering (QA) is retrieving answers to natural language questions from a collection of documents rather than retrieving relevant documents containing the keywords of the query which is performed by search engines. What a user usually wants is often a precise answer to a question. For example, given the question "Who won the nobel prize in peace in 2006?" what a user really wants is the answer "Dr. Muhammad Yunus", in stead of reading through lots of documents that contain the words "win", "nobel","prize", "peace" and "2006" etc. This means that question answering systems will possibly be integral to the next generation of search engines.

The *Text Retrieval Conference, TREC*[1] QA track is the major large-scale evaluation environment for open-domain question answering systems. The questions in the TREC-2007 QA track are clustered by target, which is the overall theme or topic of the questions. The track has three types of questions: 1. *factoid* that require only one correct response, 2. *list* that require a non redundant list of correct responses and 3. *other* questions that require a non redundant list of facts about the target that has not already been discovered by a previous answer.

We took the approach of designing a question answering system that is based on document tagging and question classification. Question classification extracts useful information (i.e. answer type) from the question about how to answer the question. Document tagging extracts useful information from the documents, which will be used in finding the answer to the question. We used different available tools to tag the documents. Our system classifies the questions using manually developed rules.

2 System Overview

The global architecture of our system is shown in Figure **??**. Each of the modules of the system is described below.

2.1 Question Normalization

Before classifying the questions, our system changes some of the questions into a standard form that will be easier to classify.

Questions with *'s* after the question word means *is*. For example, the the question *Where's Montenegro?* is changed to *"Where is Montenegro?"* For any question starting with a preposition

[1] http://trec.nist.gov/

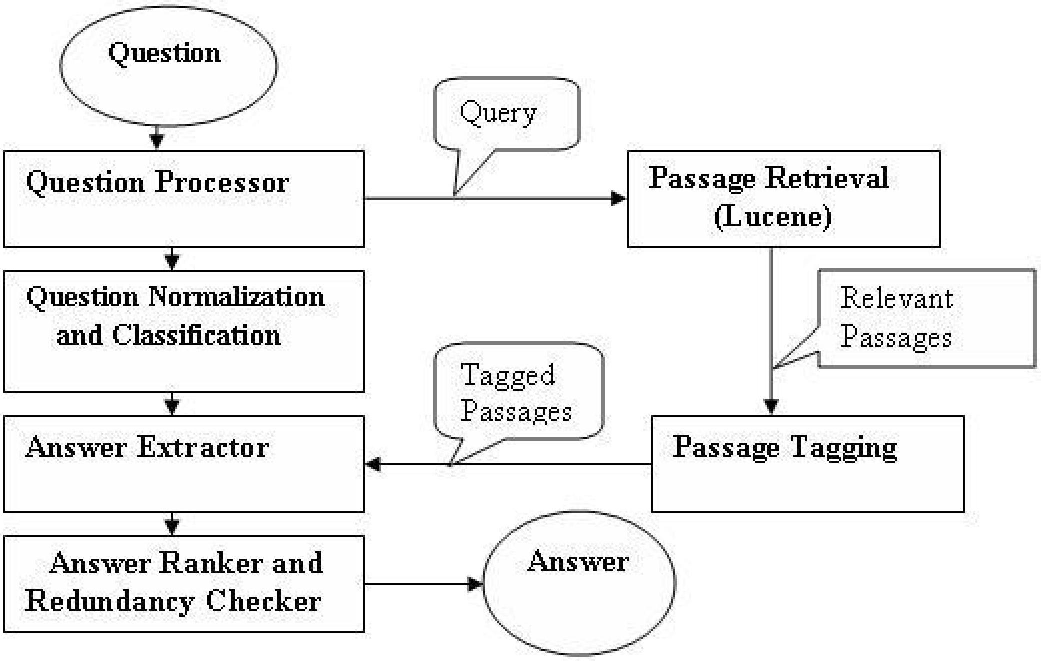
Question
Query
Question Processor
Passage Retrieval (Lucene)
Question Normalization and Classification
Relevant Passages
Tagged Passages
Answer Extractor
Passage Tagging
Answer Ranker and Redundancy Checker
Answer

Figure 1: QA System

followed by the word *what* or *which*, the preposition is extracted. Another quick question reformulation is extracting *the name of* from questions that start with *What was the name of* or *What is the name of*. Some other major normalization involves:

2.1.1 Pronoun Resolution

The questions addressing a target often refer to the target by pronouns (i.e. he, she, it, they, his, her, its, their). We replace the personal pronouns (i.e. he, she, it, they) by the *target* and the possessive pronouns (i.e. his, her, its, their) by *target's*.

2.1.2 "What" Factoid Normalization

There are different ways to say the same thing. For example, the questions: *"Name a film in which Jude Law acted."*, *"Jude Law was in what movie?"*, *"Jude Law acted in which film?"*, *"What is a film starring Jude Law?"*, ask the same information. So, they should all be classified as *What* questions, and more specifically, as *What-Film* questions. Our system will handle questions that involve the word *which* as a *what* question. Questions, like "*Jude Law acted in which film?*" and "*Jude Law was in what movie?*", that do not start with a question word are changed to *"What film was Jude Law acted in?"* and *"What movie was Jude Law in?"*. For this method, our system will first put a *What* at the beginning of the question, followed by the second half the question, forming *What movie* in these examples. Then, *was* is added followed by the first half of the question, resulting in *What movie was Jude Law acted in?*.

2.1.3 "What" Definition Normalization

There are questions that ask for why someone was famous, and questions requiring a definition of what something was. We normalize them to the form "*What is X?*" or "*Who is X?*", where X is the target. For example, the questions: "*What is Colin Powell best known for?*" and "*Define thalassemia.*" are changed to "*Who is Colin Powell?*" and "*What is thalessemia?*" respectively.

2.1.4 "What" List Normalization

Some list questions include the number of entities to be included in the list of answers which is helpful in returning answers for these questions, so we extract the number, and reformat these questions to a general format. For each of these questions, the number is passed on to be used when the answer list is returned. Examples of normalization of these questions are: "What are titles of books written by Krugman?" *is normalized to* "What titles of books written by Krugman?" and "Name 10 auto immune diseases." *is normalized to* "What auto immune diseases?"

2.2 Question Classification

Our system classifies questions into one of the categories: who, when, why, how, where and what. Any question that does not include one of the first five stems (Whom is considered as Who) is considered as *what question*. Once the questions are categorized into one of the categories, a finer classification is done to extract the answer type of the question. Classifying questions in this way aids our system in finding the answer type of the question, as well as helping to create the query to retrieve passages from our information retrieval system. Our system classifies questions using rules that we derived through observation. Supervised machine learning can also be used to classify questions, using methods similar to those outlined by **?**) and **?**).

2.2.1 When Questions

When questions always ask for a *DATE*, but there are a few that are only asking for a day, and not a full year. The ones asking for a day will be asking for a special day that might appear on the same day each year, like a birthday, or Labour Day.

Question Type	Answer Type	Pattern	Example
When-Day	DATE without YEAR	"When is"	When is Father's day?
When-Year	DATE with YEAR	"All the rest"	When was Nimitz born?

Table 1: When Question Categories

2.2.2 Who Questions

There are three types of Who questions outlined in Table **??**. We pass the target directly to the answer extractor for the *Who definition* questions. The answer type for *Who list* is PERSON, which is passed to the answer extractor along with the number of persons (if present in the question). For *Who factoid* the answer extractor is given the information that the answer type is either PERSON or ORGANIZATION.

Question Type	Answer Type	Pattern	Example
Who-Definition	FACTS	"Who [is or was] [NAME]	Who is Peter Weir?
Who-List	PERSONS	"Who [are or were or have]	Who are professional female boxers?
Who-Factoid	PERSON or ORG.	All the rest	Who invented paper?

Table 2: Who Question Categories

2.2.3 Where Questions

Most *where questions* require a type of location. There are other *where questions* that ask for a certain college or university someone went to. They are outlined in Table **??**. For questions asking for schools, the answer extractor will extracts all the schools. Other where questions ask for some kind of location more specifically the named entities (NE): CITY, PROVINCE, COUNTRY, GEOGRAPHICAL_LOCATION. For these questions, the answer extractor will extract these entities.

Question Type	Answer Type	Pattern	Example
Where-School	UNIVERSITY	college or university or school or degree	Where did Bill Gates go to college?
Where-Location	LOCATION	all the rest	Where is Las Vegas?

Table 3: Where Question Categories

2.2.4 Why Questions

These questions require reasons, which are not entities that our system currently recognizes.

2.2.5 How Questions

Most how questions are of the form, "*How X*", where adjective *X* gives clues about what kind of entity the question is asking for. The answer type for most of the *how questions* is straightforward. some sample categories are shown in Table **??**. The exception is the *how many questions* (not shown in table) because they require a count of a certain entity. These questions start with the phrase *How many* and after it comes the entity (nouns) needed to be counted. Examples of *How many* questions are: *How many rooms does the Las Vegas MGM Grand Hotel have?*, *How many layers of skin do we have?* To extract the entity, first we extract *How many* from the start of each question, then tag the question with a shallow parse. The reason we take off the *How many* first is because most taggers and parsers are trained on documents without question words and because of that, they do not do a very good job of tagging questions (**?**). The last noun of the first noun phrase of each tagged question will be the entity that needs to be counted. In the above example the entities will be, respectively: *rooms,* and *layers*.

2.2.6 What Questions

Some What questions are classified by pattern matching and, for the rest of questions, the question focus is used to classify the questions. The *what questions* that are classified by patterns are in Table **??**. *What definition* questions are handled similarly to the *"who definition"* questions, where the target is given to the answer extractor. *What Acro* questions are looking for either the acronym for a certain entity, or an acronym stands for. For the *"What Verb"* questions, the answer type is ambiguous. For example, there are many types of things that can be invented, discovered, and eaten. Our system does not answer these questions now. These questions can be answered using patterns in

Question Type	Answer Type	Pattern	Example
How-Large	AREA or VOLUME	How [big or large]	How big is Mars?
How-Accurate	PERCENTAGE	How [accurate]	How accurate is HIV tests?
How-Often	TIME PERIOD	How [often or frequent]	How often is someone murdered in USA?
How-Temp	TEMPERATURE	How [warm or cold or hot]	How hot is the sun?
How-Old	AGE	How old	How old is the universe?

Table 4: How sample Question Categories

Question Type	Answer Type	Pattern	Example
What-Definition	FACTS	What [is or are]	What is caffeine?
What-Acro	INSTITUTE or ACRO or PHRASE	" stand(s) for" or "acronym for" or "abbreviation for" or the ((acronym) or (abbreviation))	What does CNN stand for?
What-Verb	THING	question ends in a verb	What do manatees eat?

Table 5: What Simple Question Categories

the syntactic parse of the documents. Once we add a syntactic parse of the documents to our system, We will attempt to handle these questions.

The what questions that are not classified by pattern matching can be classified by discovering the focus of the question. In the question, "*What country is the leading exporter of goats?*", the question focus is *COUNTRY*. Once the questions are all tagged with chunked parts of speech, patterns are used to extract the question focus. The focus extracting patterns are: 1. *[NP What (type or kind or breed)] [PP of] [NP* X*]*, 2. *[NP What* X*] [NP What] [NP* X*]*, 3. *('s or ')* X*]*, 4. *[NP What] [VP (is or was)] [NP* X*]*, 5. *[NP Name* X*]*, and 6. *[NP Name] [NP* X*]*.

These focuses are then matched to a named entity, and that will be considered the answer type of the question. For each of these questions, the answer type is passed to the answer extractor.

There are questions that have a focus that is unique, and is not a named entity that is currently tagged by OAK system (**?**). For these questions, the hypernym set of WordNet is used to try to get a list of entities that are of the type of the focus.

2.3 Query Creation and Passage Retrieval

We are using java based Lucene [2] as our information retrieval system. **?**) notes that because of the Lucene system's ability to have a greater understanding of natural language, the passages retrieved can be more relevant to the query. We index all paragraphs in the document collection in case folded and stemmed fashion. Therefore, "Bank" will be indexed as the same as the "banking". The query in Lucene can be thought as "Vector-space on top of Boolean", that means it will satisfy the boolean true as well as it will rank documents. We query the documents using three methods:

We take all the nouns, verbs and adjectives from the question that are not stop words, and find synonyms for each of them, using WordNet. We use these synonyms to form a query, where each word in the synonym set for a word is separated by *OR*, and each synonym set is separated by *AND*. Our system does not include the words contained in the question focus when creating the expanded query. This query ensures that all documents retrieved should be somehow related to the question. These queries are often too restrictive and retrieve no documents.

If the first method retrieves no documents, then our system forms a boolean query with the proper nouns and dates from the question. This will usually retrieve the document with the answer to the question, but will also retrieve many documents unrelated to the question.

If the previous two methods retrieve no documents, then a vector-space search is performed taking all the important words in the query along with their synonyms.

For definition questions we do a query just on the subject (focus) to be defined.

2.4 Document Tagging

The document tagging module of our system tags useful information from the passages retrieved by Lucene. We use the Lingpipe[3] to resolve the coreference. We use OAK Tagger (**?**) to tag the passages with: 1. Part of Speech 2. Chunked Part of Speech and 3. Named Entity.

[2] http://jkarta.apache.org/lucene/

[3] http://www.alias-i.com/lingpipe

Each word in WordNet has multiple senses for the different ways the word can be used. Word sense disambiguation (WSD) is the process of determining in which sense a word is being used. Once the system knows the correct sense that the word is being used, WordNet can be used to determine synonyms. This is useful for seeing if a word from the question is associated with a word in the passage by being in the same synonym set. To tag the correct sense we developed a WSD system (**?**). It creates a list of important words in the passage. Then it measures the semantic relations (repition, synonym, hypernym, hyponym, gloss etc.) among the words using WordNet. We assign weights to the semantic relations. The sense that is mostly connected to the other words in the passage is said to be the sense it is being used in.

WordNet has sets for each word called hyponyms which are words that are examples of that word. This hyponym list can be loaded as a list of examples and a system can be made to tag these examples inside the document, similar to how OAK system tags NEs. To use this method, our system should first know what word's hyponym list is going to be used to tag the document. Knowing the NE our system is looking for will allow our system to tag only the entities relevant to the question. These entities have already been discovered in question classification. The hyponym list is then extracted and compared to the words of the document, and the entities that are in the list are tagged as possible answers.

2.5 Answer Extraction

2.5.1 Answer Extraction for List and Factoid Questions

Since the answer types and question types are known from the question classifier, and the documents are tagged with both shallow parse and named entities, extracting the information from the documents will use patterns to extract the named entities associated with the answer type. For some questions, the answer is not tagged and will be extracted using patterns.

2.5.2 "How Many" Questions

How many questions are answered using patterns and not named entities since these questions looks for a count of a certain entity. Answers will appear frequently in the tagged documents in the pattern "*[NP NUMBER ENTITY]*". For the question, *"How many hexagons are on a soccer ball?"*, the entity to be counted is *hexagons*, so the pattern it is trying to match is *[NP NUMBER hexagons].* WordNet is also used to get synonyms for the entity that is being counted. For example, given the question, *"How many floors are in the Empire State Building?"* the synset (*floor, level, storey* and *story*) of the word *floor* can be used to extract answers from the passage: *"The Empire State Building climbed to an unthinkable height of* 102 stories *in that city four years later."*

2.5.3 Names of People

The OAK system tags people in four ways: 1. PERSON, 2. LASTNAME, 3. FEMALE_FIRSTNAME and 4. MALE_FIRSTNAME. This presents a problem because the OAK tagger tags everything at once from a list of names. Some *LASTNAMES* are missing from the list of tags, and other names are also names of cities, such as the name *Paris*. The answer extracting module needs to extract the full name of the person when it is available. We used the patterns: "⟨*[A-Z_]NAME [A-Za-z]NNP* ⟩ *[A-Za-z]/NNP*" and "⟨ *NAME_TYPE NAME/NNP* ⟩ ⟨*NAME_TYPE NAME/NNP* ⟩" to extract the names.

2.5.4 Dates

Some dates that are tagged by OAK do not fit into either of these categories and are considered relative dates. These include *today, this year, this month, next week* and *midnight*. These dates are not helpful in answering questions, and are eliminated right away. Also, answers to questions that are looking for a particular date should have a four digit year in them.

2.5.5 Quantities

OAK tags each quantity it sees as a quantity, but does not tag quantities together that are of the same measurement but in different units. For instance, it will tag the measurement *4 foot 2 inches* as ⟨ *PHYSICAL_EXTENT 4 foot* ⟩ ⟨ *PHYSICAL_EXTENT 2 inches* ⟩. We extracted the full quantity using patterns.

2.5.6 Other Types of Questions

For the rest of the questions, possible answers are extracted by extracting the named entities associated with the answer type of the question. This is done by pattern matching with "⟨*NE X* ⟩" where *X* is a possible answer if *NE* is a named entity tag corresponding the the answer type of the question.

2.5.7 Definition Questions

In fact finding questions, there is always a topic that is being sought. If the question is "*Who is X?*" (X being a name of a person), there will be different methods for finding facts for it, compared to a non-person entity that will be phrased as "*What is a Y?*" In order to find patterns to extract the facts, I took all the definition questions and the sentences in the collection that contain the answers to these questions from the previous TREC years. We chunk the sentences using OAK chunker. I discovered the patterns for answers from these chunked sentences. Our current definition patterns include, with *X* representing facts and *TARGET* representing the subject of the fact:1. "[NP *X TARGET*]", 2. "*[TARGET]* , *X* (, or . or ;)", 3. ”*[TARGET]* (is or are) *X* (. or ;)", 4. "*X* called *[TARGET]*", 5. "*[BEGINNING OF SENTENCE] [TARGET]*, *X*, is *X*" and 6. "*[TARGET]* and other [NP *X*]".

2.6 Answer Ranking

This module gives a score to each possible answer, and returns the one with the highest score, or the answers with the highest scores if a list of answers is required. Many systems ((**?**), (**?**) and (**?**)) use lexical patterns to extract answers from the documents, in contrast to our approach of extracting named entities. Our system used the extracting patterns instead to rank answers. The following patterns are used to rank the answers.

2.6.1 Date of Birth and Date of Death

The date of birth and date of death of a person are sometimes put in brackets, after a person's name, e.g. "PERSON (DATE - DATE)". For example, *Elvis Presley (1935-1977), James Dean (1931-1955) are the new men.* contains such patterns.

2.6.2 "What" Location Questions

When a location is the answer type of the question, and the question contains a preposition like *in, on, from* or *near*, those words will frequently appear before the location entity that is the answer. For questions that contain *on* and *from*, *in* can also appear before the answer.

2.6.3 "When" Questions Ending in a Verb

Some questions end with a verb that represents the particular action that is being asked about. Examples of these questions are: *When was Microsoft established?*, *When was Hiroshima bombed?* etc. For these questions, an answer is usually found in the following pattern, "*VERB in* ⟨*DATE*⟩", where *VERB* represents a verb with a stem that is a synonym of the last verb from the question.

2.6.4 Who Action

Who questions often contain the action of *being*. For example: *Who was Khmer Rouge's first leader?* There are some questions that contain a physical action that are not being: *Who wrote "Dubliners"?* These actions will be represented in the passages that the answers are in, but might take different forms. For these types of questions, our system will only take the pattern of the whole action. For example, the question "*Who wrote"Dubliners"?*", has the action "*wrote "Dubliners"*", and will be the pattern our system uses to rank.

2.6.5 Word Association

Lexical chains (**?**) are created when we associate words together that have a similar theme. **?**) discussed using WordNet to help with the associations between words. Besides synonym, hypernym, hyponym and gloss each word in WordNet has a derived form as well. The derived form for a noun is a verb that is associated with the noun. With these tools there can be a path formed from the question words to the words of the passage of a possible answer.

In order to find this association, we extracted the important words(i.e. nouns, verbs, adjectives and adverbs) from the passage with candidate answer as well as from the question. Then we perform our WSD algorithm on these words(where the passage words preceeds the question words) and form lexical chains. Now, we check for each of the question words whether we have got any chain or not. A chain indicates there is a association between the question word and the passage words. In this way we count the number of association. For example,

Question: For which newspaper does Krugman write?

Passage: Paul Krugman is also an author and a columnist for The New York Times.

the passage words *author* and *columnist* are associated with the question word *write* by their semantic relation(gloss of *author* and *columnist* in this case). So, here the number of association is two one is for Krugman(repitition) and second one is for write(gloss).

2.6.6 WordNet Glossary

Our system extracts the WordNet glossary entry for the proper nouns in the question. For definition questions, the glossary entry for the target is used. For factoid questions, our system tags the glossary entry for named entities, and if it contains the answer type, that answer is given a higher rank. For definition questions, the WordNet glossary entry is passed to the redundancy checker to rank facts. For the question, *"Where is Belize located?"*, *Belize* is a proper noun and its WordNet gloss is: "*a country on the northeastern coast of Central America on the Caribbean; formerly under British control*. The gloss tagged with POS and NE is:

> a/DT country/NN on/IN the/DT northeastern/JJ coast/NN of/IN ⟨ GEOLOGICAL_REGION Central/NNP America/NNP ⟩ on/IN the/DT ⟨ GEOLOGICAL_REGION Caribbean/NNP ⟩ ;/: formerly/RB under/IN ⟨ NATIONALITY British/JJ ⟩ control/NN.

For *where questions*, *GEOLOGICAL_REGION* is an accepted answer type, and both *Central America* and *Caribbean* are acceptable answers for this question.

2.6.7 Target distance

One other method that is commonly used to rank answers is the distance between a possible answer and keywords from the question. Our system calculates distance as the count of words and punctuation between the important words from the question and the possible answer. If there are no such entities in a question, then this method is not used to rank answers for that question. For example, the question "*What does Kurt Vonnegut do for a living?*", has a target of *Kurt Vonnegut*. The passage contains the answer to this question: " Trust a crowd, says *author Kurt Vonnegut*, to look at the wrong end of a miracle every time." The answer *author* is considered, by our system, to be one word away from the target of the question.

2.6.8 Redundancy Checking

Answers that appear frequently is given a higher rank if the retrieved documents are related to the question. If the question requires more than one answer, each answer returned by the system should be unique. When this module returns an answer, it performs a check to see if the answer has already been included in the final answer list. This check is performed by checking whether the answer contains nouns that have already appeared in a previous answer.

For definition questions, an answer can contain more than one fact. For example for the target *Jane Goodall*, the phrase, *the British primatologist*, gets extracted. This contains the fact that she is *British*, and the fact that she is a *primatologist*. If this is the case, then the phrase that contains two or more facts is added to the list of answers, and if either of the contained facts were found in the list, their score will be added to the score of the combination fact.

2.6.9 Answer Ranking Formula

Factoid and List Questions: The ranking formula is developed by reviewing which weights for the methods above yielded the highest accuracy on the corpus of questions and their answers. For *Factoid and List Questions* our answer ranking formula is:

score of an occurance of a candidate answer = $(6 \times w_1) + (w_2) + (3 \times w_3) + \frac{1}{w_4}$ where,

w_1 denotes whether the answer is found in a pattern associated with the question type. The value will be 1 if it was found in such a pattern, and 0 if it was not.

w_2 denotes how many words from the question are represented in the passage with the answer, plus 3 more points for each word that is represented by a disambiguated word.

w_3 denotes if the answer appears in the WordNet glossary for important words from the question. (value of 0 if it does not and 1 if it does)

w_4 denotes the distance between the important words from the question and the answer.

Each answer's final rank is the sum of the ranks of the occurrences of that answer.

Definition Questions: We rank answers to definition questions, giving them one point for repetition and four points if found in the WordNet gloss.

3 Evaluation

Out of 360 *FACTOID* questions in TREC-2007, 16 had no known correct response in the document collection, hence “NIL” is the correct answer for these questions. Table **??** shows our scores for FACTOID questions.

Total	Globally Correct	Locally Correct	Wrong	Unsupported	Inexact	Accuracy	“NIL” Precision	“NIL” Recall
360	93	5	231	12	19	0.258	8/153 = .052	8/16 = .5

Table 6: UofL Score for FACTOID questions

For the 85 *LIST* questions in TREC 2007, our system achieved the “average F-measure“ score of 0.132 with recall and precision weighted equally ($\alpha = 0.5$). Our system achieved the “average pyramid F-score” (with $\beta = 3$) of 0.030 over 70 *OTHER* questions. In TREC 2007, there were questions about 70 topics (targets or series). Our system achieved average per-series score of 0.141.

4 Conclusion

This year we have improved our rank in all three types of questions. This overall improvement was primarily because of our expanded classification and the addition of word dependencies to answer ranking. Our system extracted the answers from the top 100 (50 AQUAINT-2 + 50 BLOG06) relevant documents provided by TREC instead of using the whole collection. Our system did not find answers (i.e. NIL) for 153 FACTOID questions. According to the TREC-2007 evaluation there were 16 questions with "NIL" answers. Our system could find 8 correct "NIL" answers. This clearly indicates that if we would use the whole document collection, our system could have found more answers (i.e. less number of "NIL" answers) hence, we could have scored very good results in TREC 2007.

References

Chali, Yllias and Shafiq R. Joty. (2007). Word sense disambiguation using lexical cohesion. In *Proceedings of the 4th International Conference on Semantic Evaluations*, pages 476–479, Prague. ACL.

Echihabi, A., U. Hermjakob, E. Hovy, D. Marcu, E. Melz, and D. Ravichandran. (2003). Multiple-engine question answering in textmap. In *Proceedings of the Twelfth Text REtreival Conference (TREC 2003)*, pages 772–781, Gaithersburg, Maryland.

Hermjakob, U. 2001. Parsing and question classification for question answering. In *Proceedings of the Association for Computational Linguistics 39th Annual Meeting and 10th Conference of the European Chapter Workshop on Open-Domain Question Answering*, pages 17–22, Toulouse, France.

Li, X. and D. Roth. (2002). Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)*, Taipei,Taiwan.

Moldovan, D., S. Harabagiu, C. Clark, M. Bowden, J. Lehmann, and J. Williams. (2004). Experiments and analysis of lcc's two qa systems over trec2004. In *Proceedings of the 13th Text REtrevial Conference (TREC 2004)*, Gaithersburg, Maryland.

Moldovan, D. and A. Novischi. (2002). Lexical chains for question answering. In *Proceedings of COLING 2002*, pages 674–680, Taipei, Taiwan.

Morris, J. and G. Hirst. (1991). Lexical cohesion cornputed by thesaural relations as an indicator of structure of text. *Computational Linguistics*, 17(1):21–48.

Prager, J., J. Chu-Carroll, K. Czuba, C. Wlty, A. Ittycheriah, and R. Mahindru. (2003). Ibm's piquant in trec2003. In *Proceedings of the Twelfth Text REtreival Conference (TREC 2003)*, pages 283–292, Gaithersburg, Maryland.

Ravichandran, D. and E. Hovy. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual meeting of the association for Computational Linguistics (ACL)*, pages 41–47, Philadelphia, PA.

Sekine, S. (2002). Proteus project oak system (english sentence analyzer), http://nlp.nyu.edu/oak.