

A Question Answering System Based on VerbNet Frames

Dunwei WEN ^{*†1}, Shen JIANG ^{#2}, Yangjian HE ^{‡3}

^{*}*School of Computing and Information Systems, Athabasca University
Athabasca, AB, T9S 3A3 Canada*

¹dunweiw@athabascau.ca

[#]*Department of Computing Science, University of Alberta
Edmonton, AB, T6G 2R3 Canada*

²sjiang1@cs.ualberta.ca

[‡]*School of Information Science & Engineering, Central South University
Changsha, 410083, China*

³yangjian.he@gmail.com

Abstract—The precision of the answer is now essential for a question answering system, because of the large amount of free texts on the Internet. Attempting to achieve a high precision, we propose a question answering system supported by case grammar theory and based on VerbNet frames. It extracts the syntactic, thematic and semantic information from the question to filter out unmatched sentences in semantic level and to extract answer chunk (a phrase or a word that can answer the question) from the answer sentence. VerbNet is applied in our system to detect the verb frames in question and candidate sentences, so that the syntactic and thematic information as well as semantic information can be therefore obtained. Our question answering system works well especially for answering factoid questions. The experiments show that our approach is able to filter out semantically unmatched sentences effectively and therefore rank the correct answer (s) higher in the result list.

I. MOTIVATION AND CONTRIBUTION

In the early stages of question answering, the responses to questions are from the knowledge encoded in preprocessed databases. Most of the question answering systems at that time are *structured knowledge-based* [1]. As the quantity of texts on the Internet increased dramatically in the last decades, the other branch of question answering, which is labeled as *free text-based* question answering, has become another main stream.

On the other hand, also because the information available online has been growing exponentially, locating the required information resources precisely by an information retrieval system (for example, question answering system, web search engine) becomes more difficult than it was in the past few decades. Taking the web search engine for example, for most of the queries issued by users there are tens of thousands documents returned. The ranking policy of the returned results has been a key to search engine user's satisfaction [2] (for instance, the ranking techniques applied by Google Inc. contribute greatly to its success), which proves that the requirement on precision is now very strict.

As we know, one of the main differences between a web search engine and a free text-based question answering system

is that the latter takes a complete question sentence as the input instead of several words or phrases. It is clear that there are more syntactic, thematic and semantic information that can be used by a question answering system. How to make use of the extra resources to improve the precision is crucial to free text-based question answering.

Issuing the extracted key words of the question sentence for retrieval is now not always sufficient to recognize a correct answer. For example, there are a question and a sentence as follows

Q: Who directed the 2008 Beijing Olympics opening ceremony?

S: Because Michael directed a documentary on the preparation of the 2008 Beijing Olympics, he was offered a ticket to the opening ceremony.

Although all key words of the question are included in the sentence, however, it is not a correct answer because of the incorrect positions of the key words (for example, *the 2008 Beijing Olympics opening ceremony* should be on the object position of *direct*). Syntactic, thematic and semantic information should be taken into account here to clarify the meaning of the sentence in order to make the decision on whether it is a correct answer or not.

With these in mind, we propose a practically feasible question answering system, which extracts syntactic and thematic as well as semantic information from the question and uses them to filter out incorrect answers. We also perform a set of comparative tests, and it is proved by the increased precision that our approach is an effective way to make use of the syntactic, thematic and semantic information in the question sentence. Our system is a free text-based question answering, therefore not like the structured knowledge-based question answering, it is not capable of deep reasoning.

The outlines of the paper are as follows. In section II, the free text-based question answering, the case grammar theory

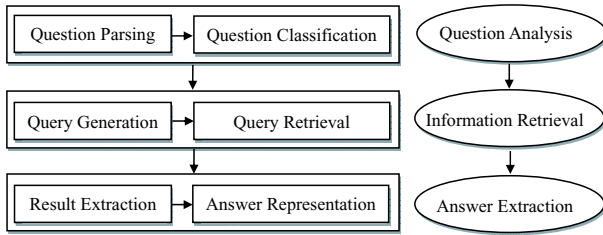


Fig. 1. A Popular Structure of Generic Text-based Question Answering System

and VerbNet will be introduced. In the next section, we present the design of our question answering system in detail. A complete example will be run through the section to visualize the processing flow. After that, we discuss two techniques that can probably enhance the system performance. In experiment section, the experiment setting and results are presented. The related work section is where we discuss the differences of our work and the other previous work. The paper ends with the conclusion and the future work.

II. BACKGROUND INFORMATION

A. Free Text-based Question Answering

For a generic free text-based question answering system, each time a user issues a question as a query in natural language, the system usually parses and analyzes the question before classifying it. The system then retrieves one or more queries generated from the question against a large size corpus. From the returned results, one or multiple clear and precise selected answers, each in the form of a sentence or a piece of a sentence, are responded to the user. Figure 1 shows the three phases of a generic free text-based question answering system.

B. Introduction to Case Grammar

Case grammar is a grammar theory proposed by Charles J. Fillmore in 1966 [3]. It is a semantics oriented model for syntactic analysis, which pays primary attentions to the relation between the structure and the semantic meaning, focusing on the uniqueness of the surface layer structure and the generality of the inside structure. In a word, the case grammar focuses on the relation between the verb and the other parts of the sentence, by which it forms a deep semantic structure.

The case grammar depicts a sentence's meaning by describing its center predication and the object of the predication. A sentence (S) can be divided into two parts, the Proposition (P) and the Modality (M). A Proposition is the set of relations between the verb and nouns, while the modality is referred to the way in which the meaning of a sentence or clause may be modified through the use of a *modal verb*, such as *may*, *can*, *will*, *must*. As showed in the following equation.

$$S = M + P$$

TABLE I
ACCOMPANY-51.7 CLASS IN VERBNET

accompany-51.7		
MEMBERS	accompany, guide, lead, steer	
THEMROLES	Agent, Theme, Destination	
FRAMES	EXAMPLES	I steer a new car
	SYNTAX	<NP/Agent> <VP/VERB > <NP/Theme>
	THEMATICS	Agent, Theme

The Proposition is made of a core verb (V), as well as some nouns and noun phrases (C_i), as it is shown in the following equation

$$P = V + C_1 + C_2 \cdots C_n$$

However, Fillmore does not make comprehensive definitions for case as well as the related concepts. Many scholars have proposed their own theory systems, which causes disagreements in the research and in the instantiations of case grammar. In this paper, we adopt the thematic roles in VerbNet as an instance of the case grammar. The reason is that although the roles in VerbNet may not be complete semantic roles [4], the thematic roles together with the verb are able to offer plenty semantic information.

C. Introduction to VerbNet

VerbNet [5] is a hierarchical domain-independent, broad-coverage verb lexicon. It was based on the verb classification rules defined in Levin. It contains syntactic information as well as syntactic information. So far, the Extended VerbNet contains the frames for 5257 verb senses, which were classified into 274 frame classes.

Every VerbNet class includes 3 parts, which are denoted as MEMBERS, THEMROLES and FRAMES. The MEMBERS denotes the member verbs in this class. THEMROLES defines the possible thematic roles in the frames in this verb class. Now there are 23 thematic roles: Actor, Agent, Asset, Attribute, Beneficiary, Cause, Destination, Experiencer, Extent, Instrument, Location, Material, Patient, Predicate, Product, Proposition, Recipient, Source, Stimulus, Theme, Time, Topic, and Value.

FRAMES defines the verb frame of a verb class, it is also the core of VerbNet. Every Frame consists of 3 parts, EXAMPLES, SYNTAX, SEMANTICS. EXAMPLES contains the example sentences that match the Frame. SYNTAX contains the syntax structure and thematic roles of a sentence. SEMANTICS contains semantic information in the form of a conjunction of Boolean semantic predicates. Now we employ a specific VerbNet class to demonstrate its structure. The TABLE I illustrates the components of the *accompany-51.7* class.

For the use of our system, a VerbNet frame can be simply abstracted to two layers, which are the syntactic layer and the thematic layer. The syntactic layer depicts the syntactic structure of the frame. For example, in the frame of class

accompany-51.7, the syntactic annotation "NP VP NP" represents the syntactic layer, preserving the syntactic information. In the other layer, the thematic layer, the role annotation "Agent Verb Theme" represents the thematic layer. It describes the structure of the thematic roles in this frame.

The VerbNet is still consummating; currently it is mainly aiming at aiding the NLP tasks like thematic analysis and thematic annotation.

III. OUR APPROACH

The group of roles in a VerbNet frame can be regarded as the instances of the case in case grammar theory. In our system, verb frames in question and candidate sentences are detected, so that the thematic information can be therefore obtained. When one of the frames of the question is matched to one of the frames of a sentence in the retrieval corpus, this sentence is regarded as an *answer sentence* and a more specific *answer chunk* is extracted from the chunk whose thematic role is identical to the thematic role of the interrogative word in the question. In this way, the sentences with key words which do not have the matching thematic roles are filtered out.

The details of the system are described in the following subsections. Fig. 2 illustrates the process flow of our system with an example run through.

A. Question Processing

1) *Syntactic Analysis and Preprocessing*: First of all, we parse the question by Treebank parser to obtain structured Treebank chunks for further analysis. Although the Treebank chunks are in a tree structure, in most of the cases, we only consider them as a sequence of chunks. The exceptions will be presented in the subsection of *Answer Selection*. Only after the syntactic analysis is done, the VerbNet frames that exist in the question are able to be detected. The punctuation marks are removed and all characters are transformed to lower case.

The head verb of the question is determined in this stage (in the following of this paper, we refer the verb in a sentence to be the head verb in this sentence).

2) *Word Order Adjustment*: As we know, the word order in a question is different from that in a declarative sentence. Most questions and their corresponding declarative counterpart have distinct syntactic structures. The positions of the auxiliary verbs in the question and the interrogative words in a question bring difficulties when we are trying to find a verb frame for this question. For instance, there is such a question

"What does Linda give to a student?"

which is parsed to

[NP what/WP] [VP does/VBZ] [NP Linda/NNP]
[VP give/VBP] [PP to/TO] [NP a/DT student/NN]

There are two verbs (including *does*) in the sentence, and the object part, which is represented by the word *what*, is in the front instead of being in the position right to the verb *give*. While, the frame in VerbNet for this syntactic structure of the question's declarative counterpart is

NP/Agent VP/VERB NP/Theme PP/Prep
NP/Recipient

Therefore, we have to recover the word order in the question to its declarative counterpart.

[NP Linda/NNP] [VP give/VBP] [NP what/WP] [PP to/TO] [NP a/DT student/NN]

Now we can easily find the VerbNet frame that matches the syntactic structure of this question.

3) *Tense and Voice Unification*: There could be many tenses and voices of the question and sentences in the corpus. For the effectiveness of frame detection, we transform all tenses to simple present tense, all voices to active voice. In this way, we wipe off the modality part of a sentence, while the proposition part is preserved.

4) *VerbNet Frame Detection*: Once the question is well parsed, the order of each chunk is adjusted and the tense and voice of the question is unified, the frame detection for sentence can be done by matching the syntax of each frame with the syntactic Treebank annotations of the declarative question sentence. There are two layers for each frame, which are syntactic layer and thematic role layer. The matching in the syntactic layer indicates the matching in the thematic role layer. Therefore we obtain the thematic roles for each chunk, which is inspired by D. Gildea and D. Jurafsky's idea to automatically label semantic roles [6].

For some verbs, there are multiple frames that share the same syntax but with different thematic role structures. Therefore, a question can be mapped to multiple frames, and each chunk may be matched to more than one thematic roles.

For instance, there are such two frames for the verb *give* (not all frames for *give* are listed here):

Frame 1: NP/Agent VP/VERB NP/Theme PP/Prep
NP/Recipient

Frame 2: NP/Agent VP/VERB NP/Theme PP/Prep
NP/Asset

Frame 1 and Frame 2 have identical syntax. The adjusted question "*Linda give what to a student?*" will be mapped to both of them. Since only the syntactic is available for frame detection, hence we cannot distinguish which one of them is the better one.

To be clear, in the following of this paper, we refer the *frames of a sentence (or a question)* to the VerbNet frames each of which matches the syntactic structure of a sentence (or a question).

B. Retrieval

There is one query generated and issued for each frame of a question. The verb and all the nouns in the chunks that are mapped to roles in this frame are used to form a query for this frame. The query is a Boolean query, where the verb and nouns (they are referred as *key words* in the follows) are connected by logic operator AND.

For the frame

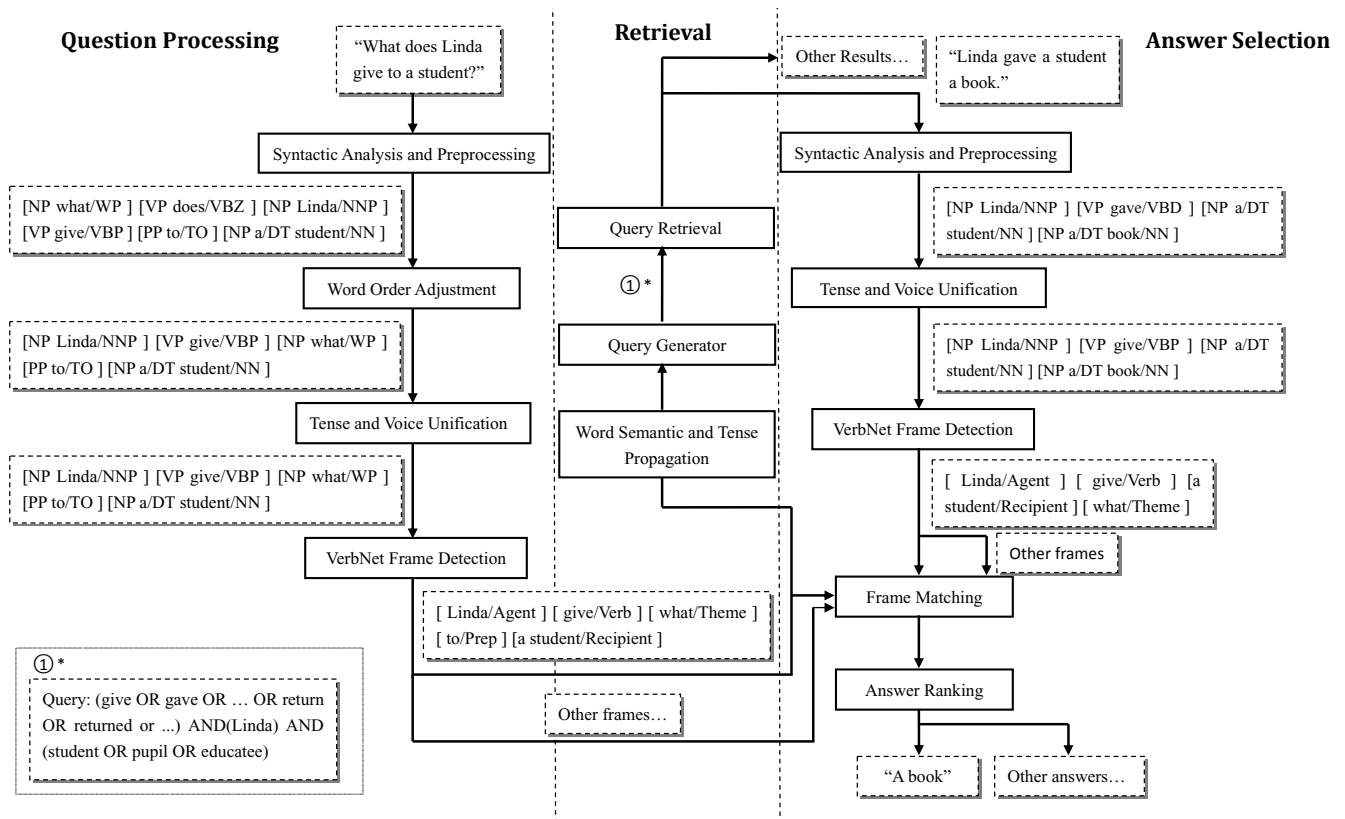


Fig. 2. The Processing Flow of Our System

NP/Agent VP/VERB NP/Theme PP/Prep
NP/Recipient

which is detected from the sentence "Linda give what to a student", the verb *give* and the nouns *Linda* and *student* form the query

(give AND Linda AND student).

However, the wording and expressions in the corpus may differ from those in the question. Applying this simple query may lose many answer candidates. Therefore we propagate the verb and the nouns. For each verb or noun, we look for no more than 20 semantically related words. Instead of directly put a verb or a noun into the query, we assemble an OR clause with the word and its semantically related words, and then AND all those OR clauses.

Take the question in the last example for instance, the verb *give* is semantically related to the word *pay*, *return*, *grant* and other words, and the word *student* is related to the word *pupil*, *educatee* and others, and there is no related word for the word *Linda*. Then the query will be

((give OR pay OR return OR grant OR ...) AND (Linda) AND (student OR pupil OR educatee)).

The semantically related words of a word *w* are the words in the *synsets* in WordNet, to which *w*'s senses belong. It means

that at least one sense of the related words of *w* has similar meaning to at least one sense of *w*.

We also propagate the tenses for the verb. Therefore, the query is updated to

((give OR gave OR ... OR return OR returned OR ...) AND (Linda) AND (student OR pupil OR educatee))

Instead of returning documents, the query retrieval system returns sentences that satisfy the clauses in the query for answer extraction.

C. Answer selection

As discussed in the motivation and contribution section, the number of the results returned from the retrieval usually is huge. Now it is time to use the information extracted from the question sentence to select correct answers. The idea is to filter out the sentences whose frames do not match any of the frames of the question or do not have the key words in correct positions. The key words are in correct positions here means they have the same thematic roles as they do in the question. Before those steps, we have to parse all the sentences in the retrieval result list and detect their frames.

The answer extraction of our system processes each sentence returned by the search engine basically the same way as it processes the question. Those returned sentences are denoted as *answer candidate sentences* in the following sections.

Firstly, each answer candidate sentence is parsed and pre-processed by Treebank parser. There is a difference here. There may be one or more subordinate clauses in an answer candidate sentence (there may be clauses in questions but only in very few cases). In these cases, we treat each clause as a sentence and add it to the set of answer candidate sentences. Furthermore, in this sentence we view the clause as a noun phrase. This may not be the perfect way to handle the clause, but it does make sense. In each clause, there are complete components of a normal sentence. If the clause matches the frame, and all the key words are in the right position, it is feasible to return the clause as an answer sentence. In practical, we return the original sentence containing the clause instead for completeness. On the other hand, the clauses in the original sentence usually serve like a noun phrase. In the cases where there are clauses in the question sentence, we simply regard the clauses as noun phrases and do not treat each clause as a new question since they are not complete questions.

There is no word order adjustment for declarative answer candidate sentences. Then the tenses and voices are unified for each answer candidate sentence, and their VerbNet frames are detected. There are also multiple frames that could be found for one answer candidate sentence.

1) *Frame Matching*: The final step is to judge whether a question semantically matches an answer candidate sentence and to extract the answer chunk from the sentence. Only if at least one of the frames of the question matches at least one of the frames of the answer candidate sentence, the answer candidate sentence is regarded as one *answer sentence*. And the *answer chunk* is extracted from the answer sentence. There are only two requirements for two frames being matched. To illustrate, we denote the frames of the question as

$$F_Q = F_{Q,1}, F_{Q,2}, \dots, F_{Q,|F_Q|}$$

and the frames of the answer candidate sentence as

$$F_S = F_{S,1}, F_{S,2}, \dots, F_{S,|F_S|}$$

Suppose the frame of question and the frame of the answer to judge are $F_{Q,i}$ and $F_{S,j}$. The requirements are:

(1) Each thematic role of key word (noun and verb in the frame) in the frame $F_{Q,i}$ should be the same as it is in the frame $F_{S,j}$.

For example, there are following two frames as follows

Question:

[Linda/Agent] [give/Verb] [what/Theme] [to/Prep]
[a student/Recipient]

Answer Candidate Sentence 1:

[Linda/Agent] [give/Verb] [a student/Recipient] [a book/Theme]

The noun *Linda* is an Agent in the frame of the question, and the noun *student* is a Recipient. And in the answer candidate

sentence, they are also Agent and Recipient.

The following frame of the answer candidate sentence does not match the frame of the question, because the role of the noun *Linda* and the noun *student* are swapped, therefore this answer candidate sentence is filtered out.

Answer Candidate Sentence 2:

[a student/Agent] [give/Verb] [a present/Theme]
[to/Prep] [Linda/Recipient]

(2) The role of the interrogative word in the frame $F_{Q,i}$ should exist in the frame of $F_{S,j}$. In the example, the interrogative word *what* has a thematic role Theme in the question. So there should be a thematic role Theme (which is mapped to the answer chunk to this question) in the frame $F_{S,j}$, otherwise the two frames are not considered to be matched.

Once $F_{Q,i}$, $F_{S,j}$ match, we extract the answer chunk, which is the chunk mapping to the role of the interrogative word, from the answer sentence. Store the answer together with the answer sentence and the sum of the sizes of $F_{Q,i}$ and $F_{S,j}$ as a record into the result list for ranking. We do this for all valid i and j to find out all possible matching frames and answers.

2) *Answer Ranking*: The answer chunks and the answer sentences from which the answer chunks are extracted from are all returned to the user as a result representation. Before returning the answer chunks and the answer sentences, we rank them to promote better answers. Here by answer, we mean the pair of answer sentence and answer chunk.

Each answer is ranked by the sum of the sizes of the frame pair from which the answer is extracted. The size of a frame is defined to be the number of roles (including the Verb) in this frame. The intuition is that the size of the frame probably reflects the amount of thematic information this frame keeps. A matching of two long frame indicates that the answer sentence match the question to a high degree.

For example, for the question

[Linda/Agent] [sells/Verb] [what/Theme] [to/Prep]
[a student/Recipient] [for/Prep] [one dollar/Asset]

There are two answer candidate sentences:

[NP Linda/NNP] [VP sells/VB] [NP a/DT stu-
dent/NN][NP a/DT book/NN] [PP for/IN] [NP
one/CD dollar/NN]

[NP Linda/NNP] [VP sells/VB] [NP a/DT stu-
dent/NN][NP a/DT pen/NN]

The answer from the first answer sentence is ranked topper than the answer from the second one.

IV. DISCUSSION

There are two other techniques discussed in this section. In our current system, we only apply the first technique in our system so far.

TABLE II

SOME OF THE THEMATIC ROLES ALLOWED FOR INTERROGATIVE WORDS

who	Agent, Actor, Recipient ...
what	Theme, Topic, Cause ...
where	Location, Source, Destination ...
when	Time
why	Cause
how	Theme
which	Theme, Topic, Agent ...
whose	Theme, Topic, Agent ...

A. Noise Filter

It is observed that there are noises in the chunks of a sentence, which fails the frame detection. The modal verb and/or the auxiliary verb as well as peripheral adjuncts in some cases appear between the subject and the verb or between the verb and the object so that the sequence of the chunks does not match the syntax of the corresponding VerbNet frame.

To deal with this problem, we design a filter to recognize and exclude those noises when detecting the frame. Excluding those noises usually does not change the meaning of the sentence to a big degree, therefore applicable.

B. Frame Filter

In the frame detection, there may be multiple frames found for a sentence, since the syntactic information is not sufficient for us to clarify which frame is the right one. There are some frames that have the same syntactic layer but different thematic role layer.

The interrogative word in a question gives us a hint. The reason is that when the question is a *who*-question, we know that the thematic role of the word *who* cannot be a Theme nor an Asset. Therefore, we can use this thematic information to build a frame filter to filter out impossible frame for a question.

TABLE II list some of the thematic roles allowed for interrogative words. This table is not guaranteed to be accurate. It is the table we manually built for simple test.

We do not use the frame filter so far because we do not have an accurate table, which can tell us all the possible thematic roles allowed. A simple test does not show the simple table we created helps. But if there is a way we can obtain a complete such table, we may have a chance to improve the system performance.

V. EVALUATION

We design a small test platform, on which we test the performance of our system. The questions are manually formed from sampled sentences in the Reuters-21578 Distribution 1.0 text categorization test collection¹. The test we use in the experiments includes 100 sentences and corresponding 100 pairs of ground truth answer sentence and answer chunk which we manually built.

Table III shows the performance of our system. In this table, *Answer Sentence Only* is correct means the ground truth

sentence is returned in top 20 of the list of answer sentences. If the ground truth sentence is not returned by our system or ranked lower than 20 (exclusively), the case is judged as an *incorrect* case. *Answer Sentence and Answer Chunk* is correct means not only the ground truth sentence is returned in top 20, but also the ground truth answer chunk is returned in the top 20 too. We pick 20 as the threshold simply because users are likely to only view the first 20 results returned in search engines [7]. We can see from the table that the system can correctly answer 62% of the questions, and it is able to extract correctly 43% answer chunk. There are 19% percentage cases where the correct answer sentence is found but the correct answer chunk is not successfully extracted. A possible explanation is that in the ground truth, some of the correct answer chunks contain postmodifiers, while the answer chunks generated by OpenNLP Treebank parser [8] usually separate the nouns and postmodifiers in two chunks.

Table IV shows the success rate of each functional module of our system. The question processing is judged fail generally in two cases: (a) the head verb or interrogative word is not found; (b) the frame detection fails. One of the reasons why the first case happens is that VerbNet does not include all verbs. If the head verb of the question is not included in VerbNet, the question processing fails. The other reason why the head verb is not found is that the OpenNLP Treebank parser does not always parse a sentence correctly. Sometimes, the actual verb is parsed as a noun (especially when it is a word that can be a noun in other context). On the other hand, in some sentences, the verb collocations (for example collocations in a format as *somebody wants to take something*) or some other irregular but legal word usage makes the verb and its arguments (such as subjects, objects) not in the same order as what is required by the syntax of the VerbNet frame. It is observed that this is the main reason for the failure of the frame detection. To handle the cases when the question processing fails, we can choose to apply another question answering system (or simply retrieve the query consisting of the key words and then return the retrieval results) in practical applications. The retrieval success rate is 100% indicates that there are always sentences returned for the queries generated. The 10.96% failure rate of answering selection means that there are 10.96% cases where there is no sentence returned after the filtering.

In table V, we can see how the syntactic, thematic and semantic information can improve the precision of the result. In this table, *With Filtering* here refers to applying the filter which only allow the sentences that match at least one of the frame of the question with the matched thematic roles for key words. While *Without Filtering* means returning the sentences in the retrieval result without using the semantic information to do filtering. Note that, we consider the correctness of only the *answer sentence* here, since without the filtering, we cannot extract answer chunks from the answer sentences. We can see that, without filtering, only 41.10% of the ground truth sentences are returned to the first place of the result list, while our approach actually increase this percentage by 17.80% to 58.90%. The other observation is that there are more cases

¹Available from <http://www.research.att.com/~lewis>

TABLE III
CORRECTNESS

	Correct	Incorrect
Answer Sentence Only	62%	38%
Answer Sentence and Answer Chunk	43%	57%

TABLE IV
SUCCESS RATE

	Success	Fail
Question Processing	73%	27%
Retrieval	100%	0%
Answer Selection	89.04%	10.96%

TABLE V
THE EFFECT OF THE FILTERING

	Top 1	Top 3	Top 20	Beyond 20 (exclusively) or Not Found
Without Filtering	41.10%	61.64%	90.41%	9.59%
With Filtering	58.90%	71.23%	84.93%	15.07%

where the one with filtering ranks the ground truth answer lower than 20 or fails to return the correct answer, compared with the one without filtering. An explanation is that in some cases, the frame detection for the correct answer sentence fails and therefore the correct answer sentence is filtered out in the frame matching phase.

VI. RELATED WORK

Free text-based question answering has been studied and most of free text-based question answering systems tend to share common pipelines with ours, such as the ones presented in Prager et al. [9], Xu et al. [10], Hovy et al. [11], Moldovan et al. [12]. There is a category of approaches (including ours) try to find patterns associated with questions to identify correct answers or re-rank answer candidates. Some of related works in this category are reviewed in the following.

Kaisser and Webber [4] developed a Question System, which is based on dependency tree and employs some similar techniques with ours. One of the major differences is that we semantically propagate the key words (verbs and nouns in the question) so that even the wording or expressions differs in question and in corpus, the answers are still able to be found. The other major difference is that, in the query generation phase, the queries generated by their system are all in a form of quoted *answer template*, which follows a verb argument structure with an unknown argument. They introduce all possible tenses and voices into the queries in order to deal with the mismatching of tenses and voices in question and answer candidate sentences. While, in our approach, the generated queries are not in a form of a template, but a sequence of key words (in the question) and propagated words. The queries in their system are quoted, which require a strict matching in every character. It is possible that some answer candidate sentences with extra adverbs or auxiliary words in the verb argument structure are excluded.

The system described in Brill et al. [13] also builds their

queries in a form of *answer template*. The user's question is reformulated to a declarative sentence (similar to what our Word Order Adjustment module does). Then the interrogative word is left out and the rest of the sentence is retrieved. For instance, "What is relative humidity?" would be rewritten as "Relative humidity is what", then "Relative humidity is" is retrieved as a query.

Novischi and Moldovan [14] combine lexical chains and verb argument structures to re-rank answer candidates. The aim is to deal with varying syntactic roles when the head verb of an answer sentence is different from that of the question sentence. Their approach semantically propagates the verb to obtain a wider coverage of verb argument structures. In this way, they make use of the thematic information from roles to re-rank an in-house question answering system. It is reported that their approach achieve a performance gain of 2.4% on factoid questions of TREC 2004. Their work is similar to our semantic extension on verbs. In our approach, for each verb, we only use the words in the same WordNet synset for propagation, while they create the lexical chains with four relations: HYPERNYM, TROPONYM, ENTAILMENT and CAUSATION. There is a tradeoff here between the coverage and the semantic similarity of words. Experiments are expected to find the balance point.

In Hermjakob et. al. [15], the authors introduce their Weblopedia question answering system. This system is able to pinpoint brief phrasal answers within the documents returned by IR engine. They build ISI's QA Typology which contains 140 Qtargets. Qtarget is an equivalence class of all phrasings for the answers, serving as an answer type. Once the IR engine returned a number of documents, a matcher module then compares their parse trees to those of the original questions in terms of the Qtargets. The matcher module has a backup policy if the matching of Qtargets fails. Their system won the second place in 2000 and fourth in 2001. The Qtargets in their system works in a similar way as the thematic roles in our system do.

They introduce Qtargets to pinpoint the answers within the retrieval result, while we make use of thematic roles to filter out sentences that are not semantically relevant.

All the approaches above including ours share a common idea, which is to find a pattern that links the question and answer(s) and make use of this pattern in a certain way to locate the answer. There may be cases that the pattern in the question and the pattern in the answer(s) are not verbatim identical. Lin and Pantel [16] present a method to deal with this problem. The inference rules found by an unsupervised training are actually carried by the paraphrasing of an identical or similar meaning. For example, “X writes Y” implied “X is the author of Y” is an inference rule. In this rule, the “X is the author of Y” is a paraphrasing of “X writes Y”. In this way, the problems caused by the differences in pattern can be solved in some cases. Their experimental results show that their approach can find many such inference rules that are missed by humans.

VII. CONCLUSION AND FUTURE WORK

In this paper, we present a complete and practically feasible query answering system, which is able to extract the syntactic, thematic as well as semantic information for semantic level matching.

Questions are parsed and matched to the VerbNet frames so that the syntactic and thematic information can be used later to filter out semantically unmatched sentences. WordNet is used to propagate the key words in semantic level. Then a query containing the key words and their various forms in different tense as well as related words is generated and issued. At last, the syntactic, thematic and semantic information is used to filter out semantically unmatched sentences in the form of frame matching and thematic role matching.

The result shows that the system is able to rank higher the correct answers by filtering out those sentences with unmatched meanings, so that the correct answer can be easily located by the user. There are 41.10% percent correct answers sentences are returned in the first place of the retrieval result, while after the filtering 58.90% correct answers sentences are ranked 1st, proving that the way we make use of syntactic and thematic as well as semantic information is feasible and effective.

One future work is to decrease the failure rate of the question processing. Since we cannot just require the parser or VerbNet to be perfect, the focus is to solve the problem caused by the Verb collocations and irregular word usage, aiming to achieve a higher frame detection success rate.

ACKNOWLEDGEMENT

This work is partly supported by Mission Critical Research (MCR) fund and Research Incentive Grant (RIGS) from Athabasca University.

REFERENCES

- [1] D. Mollá and J. L. Vicedo, “Question answering in restricted domains: An overview,” *Comput. Linguist.*, vol. 33, no. 1, pp. 41–61, 2007.
- [2] B. Yaltaghian and M. H. Chignell, “Effect of different network analysis strategies on search engine re-ranking,” in *CASCON '04: Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 2004, pp. 308–317.
- [3] C. Fillmore, “Towards a modern theory of case,” *Project on Linguistic Analysis*, 1996.
- [4] M. Kaisser and B. Webber, “Question answering based on semantic roles,” in *Proceedings of the 5th Workshop on Important Unresolved Matters*. Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 41–48.
- [5] K. Ripper, H. T. Dang, and M. Palmer, “Class-based construction of a verb lexicon,” *AAAI2000*, 2000.
- [6] D. Gildea and D. Jurafsky, “Automatic labeling of semantic roles,” *Comput. Linguist.*, vol. 28, no. 3, pp. 245–288, 2002.
- [7] B. J. Jansen and A. Spink, “An analysis of web documents retrieved and viewed,” in *International Conference on Internet Computing*, 2003, pp. 65–69.
- [8] [Online]. Available: <http://opennlp.sourceforge.net>
- [9] J. Prager, J. Chu-Carroll, and K. Czuba, “Question answering using constraint satisfaction: Qa-by-dossier-with-constraints,” in *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2004, p. 574.
- [10] J. Xu, A. Licuanan, and R. Weischedel, “Trec 2003 qa at bbn: Answering definitional questions,” in *Proceedings of TREC 2003*, 2003.
- [11] E. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C. yew Lin, “Question answering in webclopedia,” in *In Proceedings of the Ninth Text REtrieval Conference (TREC-9, 2000)*, pp. 655–664.
- [12] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus, “The structure and performance of an open-domain question answering system,” in *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2000, pp. 563–570.
- [13] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng, “Data-intensive question answering,” in *In Proceedings of the Tenth Text REtrieval Conference (TREC, 2001)*, pp. 393–400.
- [14] A. Novischi and D. Moldovan, “Question answering with lexical chains propagating verb arguments,” in *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*. Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 897–904.
- [15] U. Hermjakob, E. H. Hovy, and C. Y. Lin, “Knowledge-based question answering,” in *Text Retrieval Conference (TREC)*, 2000.
- [16] D. Lin and P. Pantel, “Discovery of inference rules for question-answering,” *Nat. Lang. Eng.*, vol. 7, no. 4, pp. 343–360, 2001.