A Hybrid Unification Method for Question Answering in Closed Domains

Abolfazl Keighobadi Lamjiri, Leila Kosseim, Thiruvengadam Radhakrishnan
CLaC Laboratory
Department of Computer Science and Software Engineering
Concordia University, Montreal, Canada
{a_keigho,kosseim,krishnan}@cs.concordia.ca

Abstract

As opposed to factoid questions, questions posed in a closed domain are typically more open-ended. People can ask for specific properties, procedures or conditions and require longer and more complex answers. As a result, detailed understanding of the question and the corpus texts is required for answering such questions. In this paper, we present a unification-based algorithm for measuring syntactic and semantic similarity of a question to candidate sentences extracted by information retrieval. The algorithm first applies strict linguistic constraints in order to identify potentially similar sentences to the question, then uses a statistical method to measure the similarity of the question's subject and object to text chunks in each of these sentences. The algorithm has been evaluated on a closed domain in telecommunications and on the TREC 2003, 2004 and 2005 questions about the *AQUAINT* corpus for comparison. The evaluation shows a precision of 81.0% in our telecommunications domain, and 60% on the TREC non-copulative questions. This confirms our hypothesis of the applicability of deep syntactic analysis for closed domain QA.

1 Introduction

In this work, we present a technique for candidate answer ranking in question answering systems. Our focus is on closed domain question answering that often uses a document collection restricted in subject and volume. Questions asked in a specific domain usually are not factual questions: they tend to be more open-ended and ask for properties, procedures or conditions and their answers are longer and more complex. As a result, a system expert in that domain is expected to perform a detailed understanding of the question and the text to be able to extract the correct answer. Instead of being a single noun phrase, usually an answer should be an entire sentence to satisfy what is asked by the end user. Because of these characteristics, closed domain QA has recently been a hot topic of research in QA [Diego Molla, 2005; 2004].

Most current TREC [Voorhees and Tice, 1999] type question answering systems rely on redundancy of answers to rank candidates; i.e. a fact that occurs more frequently in the document collection is more likely to be true. In a document collection relevant to a specific topic, candidate redundancy is less present: the content of documents cover various issues instead of including repeated information [Doan-Nguyen and Kosseim, 2006]. Additionally, low candidate redundancy makes it very unlikely to find an answer with a straight forward grammatical and lexical similarity to the question. Finally, precision is very important in closed-domain question answering because of their practical applications, such as customer service and teaching, that requires high reliability.

In this paper, we present a *Hybrid Unification Method* for scoring and ranking candidate answers for a question. This method applies strict linguistic constraints in matching the question verb with a verb in a candidate sentence, then uses a fuzzy unification for matching the arguments of the verbs. In the later process, *matched syntactic links* in the parse tree introduce a strong linguistic feature, while the number of the matching words statistically contributes to the quality of the unification. Although each technique has been investigated individually in different types of text, to our knowledge, this rich combination of syntactic and statistical criteria is unique and new to the field.

In Section 2, we explain our hybrid unification method for candidate ranking; in particular, we will focus on paraphrases which introduce syntactic differences between two sentences that convey the same meaning. Section 3 gives the evaluation results: detailed error accumulated by the *Candidate Answer Extraction* module is reported separately from the error of the unifier. The evaluation shows a very high precision of our approach in a closed domain and the category of factoid TREC questions that have a non-copulative main verb. Section 4 provides a review of relevant work and finally, we analyze the applicability of this method in open domain in Section 5.

2 Hybrid Unification

Pure linguistic criteria for measuring the similarity of parse trees impose very strict syntactic constraints that result in low recall. This problem has been observed by researchers in the field, such as [Cui *et al.*, 2005].

Statistical approaches in QA inspired us in building a hybrid unification method: forcing critical syntactic roles, and

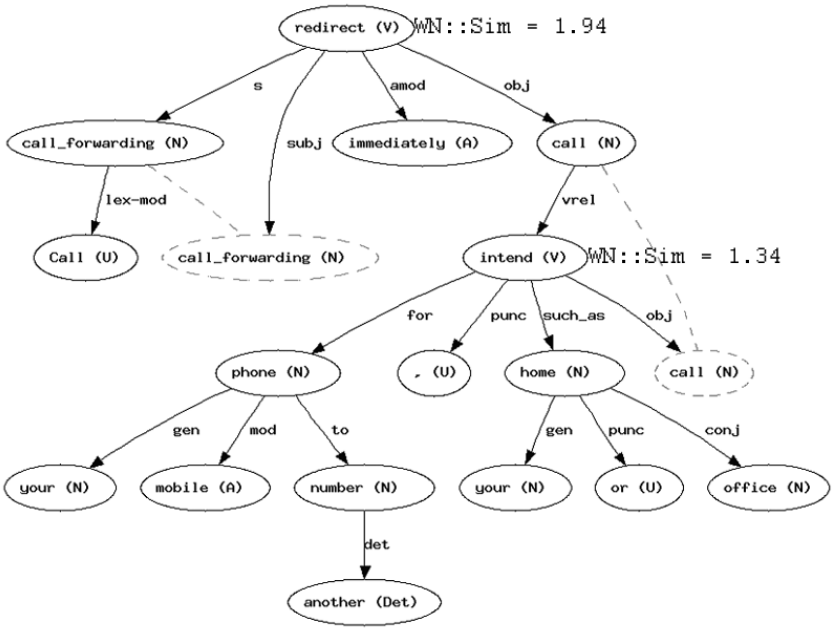

Figure 1: Finding a semantically similar verb in the sentence "*Call Forwarding immediately redirects calls intended for your mobile phone to another number, such as your home or office.*" to the question's main verb 'work'.

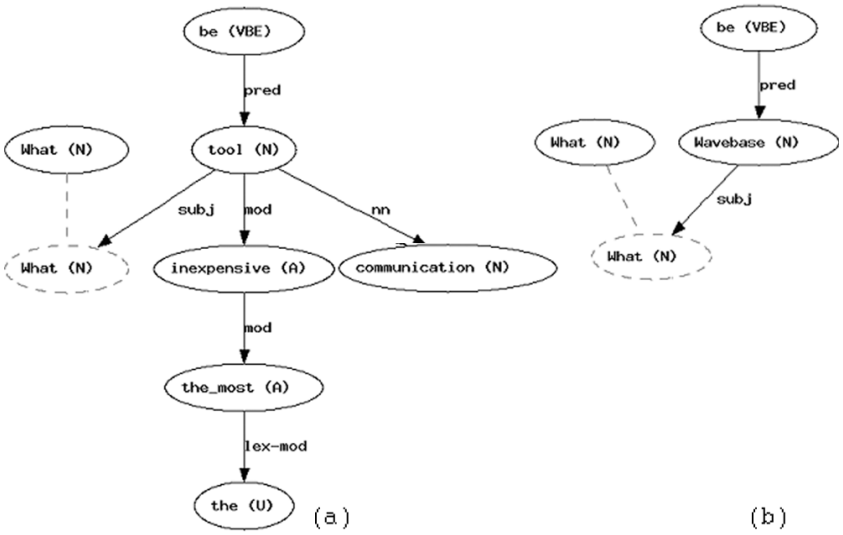

Figure 2: Parse tree of the questions (a) *"What is the most inexpensive communication tool?"* and (b) *"What is Wavebase?"*

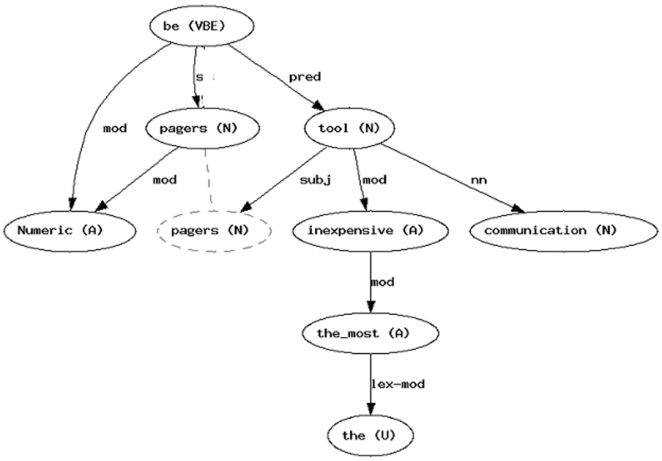

Figure 3: Parse tree of the sentence *"Numeric pagers are the most inexpensive communication tool."*

fuzzy scoring the remaining links.

2.1 Linguistic Constraints

As events and states are typically realized by verbs, our first step is to verify the semantic relatedness of the question's main verb to each verb in the candidate answer. After finding a similar verb, the compatibility of its arguments will be tested to see if the two events are actually the same.

We use Leacock and Chodorow's similarity measure from WordNet::Similarity [Pedersen *et al.*, 2004] for evaluating relatedness of two verbs. Figure 1 shows the parse tree of a candidate answer for the question *"How does Call Forwarding work?"*. The verb *'redirect'* with a similarity of 1.94 is the closest in meaning to *'work'*, the question's main verb. We can now proceed with the unification by checking whether these two verbs relate to the same entities (subject and object in particular). A fuzzy statistical method evaluates how similar the two *'subj'* subtrees are, and likewise for *'obj'* subtrees. We will look at this method in detail in the following section.

The above linguistic selection of verbs does not work well with copulative sentences that have modal verbs. Copulative questions such as *"What is Wavebase?"*, *"What is the most inexpensive communication tool?"* and *"At what speed is the network compatible with?"* have an exceptional structure. They convey a state and not an action and their argument structure is more flexible. Although the first two questions are syntactically similar (see Figure 2), their answers come in different structures: *'ANS1'* in *"ANS1 are the most inexpensive communication tool."* has a *'subj'* role (Figure 3), while *'ANS2'* in *"Wavebase is ANS2"* comes in the *'pred'* subtree (Figure 4). This phenomenon led us to allow toggling of *'subj'* and *'pred'* arguments when unifying copulative structures.

2.2 Statistical Phrase Analysis

To unify two phrases (subtrees) marked by the linguistic method as the arguments of verbs, we apply a statistical process. This step uses two measures: number of overlapping words based on a bag-of-words approach and the number of overlapping links.

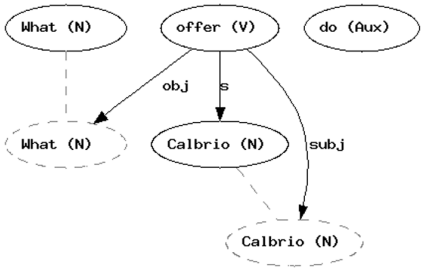

Figure 5: Parse structure for the question *"What does Calbrio offer?"*

The reason we relax our linguistic constraints at this stage is that we are focusing on a sentence that conveys a similar event or state to the question's; only a clue about similarity of its verb arguments is sufficient to conclude that its verb is affecting the same entities as the question. Syntactic differences of verb arguments should not critically affect our judgment. We reject a candidate if its arguments have no keyword based overlap with the question's. For example, the noun *"Calbrio"* in the question *"What does Calbrio offer?"* (see Figure 5) appears as *"The Calbrio workforce management system"* in the answer sentence, depicted in Figure 6. Here, a score of 1.0 is returned by matching the words ('Calbrio') and 0.0 for syntactic link overlap, since there is no common relation in the two subject subtrees. More formally, we compute $\alpha\ WordOverlap + (1-\alpha)\ LinkOverlap$ as the total unification score. The parameter α shows the relative importance of the two features: $\alpha = \frac{1}{2}$ assigns equal importance to either feature, while $\alpha = \frac{1}{3}$ (our configuration) considers the link-overlap feature to be twice as important as the bag-of-words feature. Note that the absolute value of the final score is not important since the scores are used only to rank the candidates and pick the best one.

By analyzing a few unification cases, we realized that matching of different types of links should have a variable contribution to the final unification score. Compare when a modifier *'mod'* link matches in the candidate *"wireless network"* as opposed to the noun complement *'nn'* link in the candidate *"home network..."* matches with the question *"a wireless home network ..."*. The second case shows stronger similarity since it narrows down more precisely the meaning of the noun *'network'*. A "lex-mod" link has the highest weight of 1.0 (they connect Proper Nouns), then "nn" and "mod" relating name-name or modifier-name get a weight of 0.5, and finally "determiner" and "gen" links are assigned a weight of 0.25. For the example in figure 6, the value of the $linksScore$ feature will be 0. This results in a total score of $\frac{1}{3} \times 1.0 + (1 - \frac{1}{3}) \times 0 = 0.66$ for the matching of 'subj' subtrees.

2.3 Choosing a Seed Point

We observed that starting the unification method from the most similar verb of the candidate sentence to the question's main verb does not always lead to the correct place in the candidate; a strong verb similarity must co-occur with an entity match in the subtree. This suggests that a stronger seed point is the root of the subtree that contains the question's

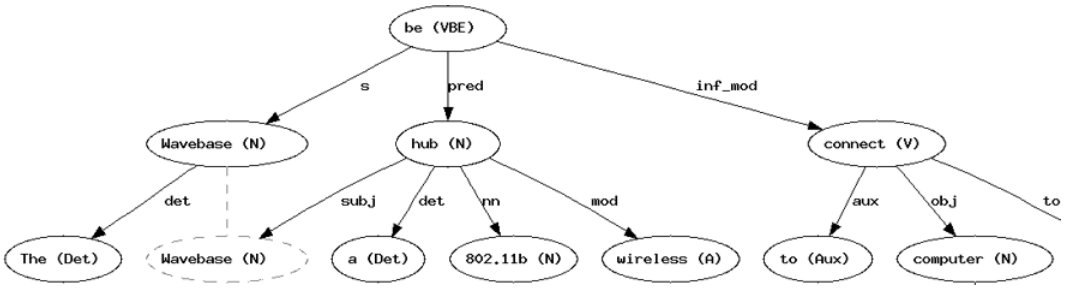

Figure 4: Parse tree of the sentence *"The Wavebase is a 802.11b wireless switched hub with 4 ports to connect up to 4 computers to ..."*

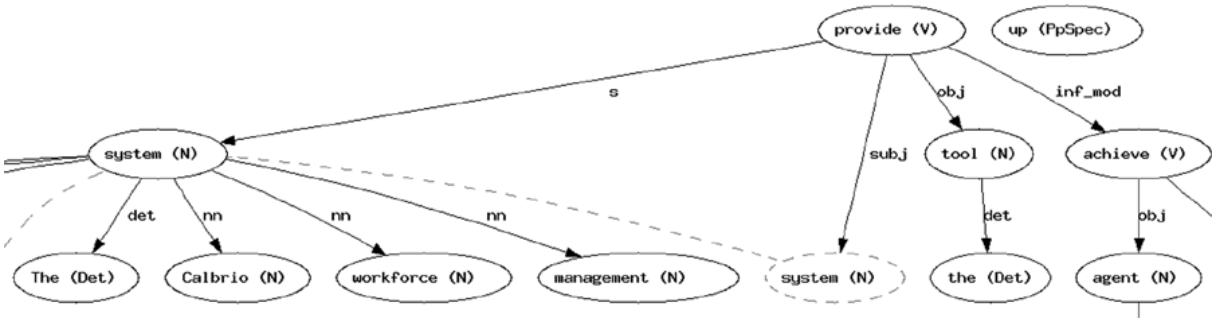

Figure 6: The parse structure for the sentence *"The Calbrio workforce management system, made up of fully integrated software modules, provides the tools to achieve optimal agent staffing ..."*

head noun phrase.

To choose the question head, we rank all noun phrases in the question and pick the one that contains the most valuable question keywords. If this head phrase is found in the candidate sentence, it is used as an anchor to find the relevant verb: we move up from the anchor to reach the first parent verb; this marks the most relevant subtree to be sent to the unification algorithm to be matched against the question. For example, *'Calbrio'* is the only noun phrase in the question shown in Figure 5. It is found in the left subtree of the verb *'provide'*. Moving up from this anchor skips the noun node *'system'* and marks the verb node *'provide'* as the seed point for performing unification. In such a long candidate sentence as this, using an anchor reduces the candidate verbs to the ones that include the question head or a reference to it.

3 Evaluation

To evaluate the performance of our unification method, we have implemented a *Question Analyser* module that extracts keywords from a question and feeds them to the Lucene IR engine[1]. A *Candidate Answer Extractor* module processes the n top documents returned by Lucene and marks all sentences that include question keywords. The output is finally given to the *Hybrid Unifier* discussed in Section 2 for ranking. These modules together build a *Question Answering* system. For our developments, we use the Gate framework[2] and the Minipar parser [Lin, 1993].

[1]Available at `http://lucene.apache.org/`

[2]`http://www.gate.ac.uk/`

3.1 The Document Collection

The algorithm was first tested on the restricted domain of telecommunications. To prepare our question-answer data set, we asked 15 students to assume themselves to be Bell Canada[3] customers and compose questions relevant to the document collection. The document collection is made up of 250 web pages, internal documents and manuals of Bell Canada. In total, it accounts for 500KB of text. Around 15 documents were assigned to each student. In general, the questions collected varied in style, length and complexity; however, as we expect in a closed domain (see Section 1) most questions are long, and complex (compare 'Bell Qs' and 'TREC Qs' columns in Table 1) and include mostly 'what' and 'how' type questions. For answering, some questions need knowledge of the domain and acronyms or synonyms, while some are very similar, lexically and syntactically to the answer sentence. In total, we randomly chose 45% of the questions for development, and the rest were kept for testing. For testing on open domain, we used the factoid questions from TREC-2003, TREC-2004, and TREC-2005 sets.

In order to have an idea of the complexity of the document collection, we compared them to several other text genres. The most popular *Readability Measures* [Greenfield, 2004] show that our texts have on average longer sentences and hence are of lower reading ease compared to *Short Stories*[4], news articles from the *AQUAINT* corpus [Voorhees and Tice, 1999], and grade 5 reading comprehension texts typically used in QA domain [Molla, 2003]. Table 1 compares

[3]The leader telecommunications service provider in Canada

[4]We took 5 classic short stories from `http://www.bnl.com/shorts/`

Statistics	Bell Qs	TREC Qs	Bell Corpus	Short Stories	AQUAINT	Grade 5
Number of words	344	396	10,947	9,672	10,088	1,525
Number of sentences	39	59	551	460	637	174
Average # characters per word	4.36	4.35	4.89	4.44	4.66	4.21
Average # syllables per word	1.43	1.46	1.71	1.49	1.55	1.36
Average # words per sentence	8.82	6.25	19.87	21.03	15.84	8.76
Readability Measures						
Gunning Fog index	6.55	6.73	13.83	12.08	10.95	4.79
Flesh Kincaid Grade level	4.73	3.93	12.30	10.18	8.93	3.85
Automated Readability Index	3.53	2.18	11.55	9.99	8.44	2.78
SMOG	8.04	7.78	13.64	11.94	11.24	6.74
Flesch Reading Ease	76.88	78.06	42.27	59.56	59.26	83.05

Table 1: Complexity of the Bell corpus compared to other genres of text.

the relative complexity of randomly chosen 60KB of text from each of these collections. The *Gunning Fog index* indicates the number of years of formal education that a person requires in order to easily understand the text on the first reading. The *Flesh Kincaid*, *Automated Readability Index (ARI)* and *SMOG* are approximate representations of the U.S. grade level needed to comprehend the text. Finally for *Flesch Reading Ease* measure, scores of 90-100 are considered easily understandable by an average 5th grader; 8th and 9th grade students can easily understand texts with a score of 60-70 (which is the case of Short Stories and *AQUAINT*), and texts with results of 0-30 are best understood by college graduates[5].

3.2 Results

As the upper bound on accuracy, we first computed the accumulated error in extracting candidate answer sentences. Since these are the sentences sent to our unifier algorithm, they impose a limit on the expected final result. Table 2 shows an accuracy of 39.3% at the sentence retrieval level for our closed-domain test set (the "IR Acc." column); this represents the percentage of questions for which at least one correct candidate sentence is retrieved and sent to the unifier by the IR. In open domain, the accuracy at this stage is around 35%.

The final accuracy of the QA system is given in the MRR measure [Voorhees and Tice, 1999] for the development and test data sets[6]. Although we have a relatively low accuracy at the sentence extraction level, the results show a high performance of the candidate ranking algorithm in telecommunications closed domain (81.0% shown in the "Unif. Acc." column). Our unifier is favored however because the students who composed the questions, saw the documents beforehand. This may have unintentionally led them to choose a syntactic structure for their questions that is close to the structure of the answer sentence.

For factual open domain questions however, the precision of the unifier drops to around 60% for the questions with a main content verb (non-copulative questions): more candidate answers are sent to the unifier (on average 11 sentences with a standard deviation of around 10) compared to closed domain (on average 6 sentences with a standard deviation of 3). The higher number of candidates makes the ranking process harder and sensitive to the weights used in the scoring. The low accuracy for the copulative TREC questions (without a main content verb) shows the important role of the main verb in our method.

Q Set	#Q	IR Acc.	MRR	Unifier Acc.
Telecom (Devel)	96	34.9%	29.6%	85.0%
Telecom (Test)	120	39.3%	32.0%	81.0%
2003 copulative	192	24.5%	7.7%	23.4%
2003 non-copul	94	23.4%	15.7%	59.1%
2004 copulative	147	36.1%	19.0%	35.8%
2004 non-copul	64	45.3%	28%	58.6%
2005 copulative	250	32.4%	16.9%	37.0%
2005 non-copul	110	33.6%	24.5%	62.3%

Table 2: Accuracy at sentence extraction and unification levels.

3.3 Analysis

Among the sources of error in the unification phase we observed the following with our closed domain:

Ellipsis in Lists List structures introduce a gap in the flow of text: the heading sentence introduces an entity and the sentences that follow, provide features for that entity, without explicitly including that entity's name. No score is considered for such elided constituents. For a few questions, appositive structures similarly cause a gap in the subject or object of a sentence.

Incorrect parse tree for the question The Minipar parser is not specifically designed to parse sentences in question form. Proceeding with unification based on a wrongly parsed question obviously results in extracting incorrect answers.

Co-reference Sometimes the answer and its supporting context come in two consecutive sentences. We need coreference resolution to identify the answer in such cases.

[5]Reader's Digest magazine has a readability index of about 65, Time magazine scores about 52, and the Harvard Law Review has a general readability score in the low 30s.

[6]Note that based on the performance of the last years TREC submissions, the questions in TREC-2003 were harder to answer, with an average precision of 12.2% for the year 2003, compared to 15.5% in the 2004 and 16.7% in the 2005 years.

4 Related Work

To compare our technique with other closed-domain systems we should note that as it was shown in Table 1, the textual genre/complexity differs significantly among closed domain works: in reading comprehension done by Molla, et.al. [Molla, 2003], sentences are short and easy to parse; each document (story) has 5 questions related to it, meaning a precise document retreival for the task. Even the evaluation in closed domain can be subjective and done on different subsystems (ex. Templates in WEBCOOP [Benamara, 2004] used in the touristic domain.) Molla in [Molla, 2003] reports an MRR of around 40% in the best system configuration for the reading comprehension task. However, it is necessary to evaluate the performance of a QA system for real-world texts.

Bnamara in WEBCOOP system uses three types of relaxation in order to cope with complex questions in the for touristic domain [Benamara, 2004]: relaxations based on cardinality, the type of the question focus, and finally the constants. These relaxations are hard-coded in the QA system. Although they cover a large proportion of queries for touristic domain, they may not work for the categories that are not predicted by the system developers.

To compare our technique with other syntactic based approaches, we can mention the university of Singapore QA [Cui *et al.*, 2005] system that base their answer extraction module on pre-extracted syntactic patterns and approximate matching of dependency relations. They calculate the cost of transforming the parse tree of the question to a candidate parse tree. In addition to bag-of-words however, we keep stop words and find the common syntactic links (ex. *"his red car"* versus *"the car at his red door"*; we gave this syntactic feature twice as much contribution in the subtree (ARG) scoring by $(1 - \alpha) = 2/3)$.

Katz et al. bring up the need for introducing syntactic constraints after applying a bag-of-words passage retrieval engine [Katz and Lin, 2003]. They focus on the following two problems and adapt a first order predicate logic formalism to address them:

1. *Semantic symmetry:* such as in *"What do frogs eat?"* and *"What eats frogs?"* which are similar at the word level.
2. *Ambiguous modification:* for example in *"the largest volcano in the Solar System?"* and *"the largest planet in the Solar system"* and *"Even the largest volcanoes found... backyard, the Solar System"*, the adjective *'largest'* modifies different entities.

Applicability of this comprehensive state-of-the-art method is shown successfully on five questions. Breaking the text into small grains in predicate-logic form is less feasible to apply in large scale and open-domain.

Salvo et al., in [de Salvo Braz *et al.*, 2005] introduce a hierarchical knowledge representation for Meaning Entailment: a sentence is entailed by a paragraph if its context graph can be unified with that of the paragraph. A cost function determines the goodness of a unification. Unified nodes must be at the same level in the hierarchy, and the cost of unifying nodes at higher levels dominates those of the lower levels. Nodes in both hierarchies are checked for subsumption in a top-down manner: The hierarchy level H_0 consists of verbs that unify if they are synonyms based on WordNet and their constituent phrases at H_1 level unify. Hierarchy set H_2 corresponds to word-level nodes. As it can be seen, syntax is used only at the topmost level H_0.

On the other hand, PiQASso [Attardi *et al.*, 2001] and AnswerFinder [Molla and Gardiner, 2004] compute the match between a question and a candidate answer using a metric which computes the overlap in their dependency relations. A similar work by Nyberg [Durme *et al.*, 2003] introduces a light-weight fuzzy unification as an extension to their earlier work, JAVELIN [Nyberg *et al.*, 2003]; here, counterpart syntactic links and their head and tail tokens contribute to the final match score. Unlike the PiQASso system, syntactic links are weighted so that a matching 'subject' link has higher contribution than a 'determiner' link. For this linguistic work however, no evaluation result is provided.

Finally, as another popular statistical method for unifying parse trees, we would like to refer to, Raina et al. [Raina *et al.*, 2005]. They learn weights for matching subtree structures at the source and destination nodes: matching of the modifier of two verb nodes may contribute less than matching of their subjects. We consider this as a linguistic feature in our unification method.

5 Conclusion and Future Work

In this paper we showed how to impose simple linguistic constraints to select only the candidates that refer to the same event or state that the question asks for and at the same time, syntactically chunk these candidate sentences. A fuzzy statistical measure then computes the similarity of each chunk in a candidate to its counterpart in the question. The similarity of the event and the main entities show high semantic resemblance of that candidate to the question and the answer is extracted and returned from that candidate.

We evaluated this algorithm on a closed domain in telecommunications and on the TREC *AQUAINT* corpus for comparison. The evaluation shows high precision in our telecommunications domain that confirms our hypothesis of the necessity of deep syntactic analysis for closed domain QA. We obtained relatively lower precision on the TREC non-copulative questions. Finally, since our method relies on the semantic similarity of the question's main verb with the candidates', it does not perform well on the copulative questions. Around one third of the TREC factoid questions are non-copulative. Finding an appropriate mapping from a copulative parse tree to a non-copulative parse tree would be interesting for the non-copulative questions that are answered by a copulative sentence, and vice versa.

Special attention should be given to parsing the question; for example, converting the question to the positive form or to use more than one parser to realize when the question is not parsed correctly should be studied. We are currently studying the combination of answer redundancy prevalent in open domain with our linguistic method to get higher performance in TREC questions.

Acknowledgement

This research was financially supported by a grant from NSERC and Bell University Laboratories.

References

[Attardi *et al.*, 2001] Giuseppe Attardi, Antonio Cisternino, Francesco Formica, Maria Simi, and Alessandro Tommasi. PiQASso: Pisa Question Answering System. In *Proceedings of the Twelfth Text REtrieval Conference (TREC-12)*, 2001.

[Benamara, 2004] Farah Benamara. Cooperative question answering in restricted domains: the webcoop experiment. In *ACL Workshop: Question Answering in Restricted Domains*, Spain, 2004.

[Cui *et al.*, 2005] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question Answering passage retrieval using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 400–407, Salvador, Brazil, 2005.

[de Salvo Braz *et al.*, 2005] Rodrigo de Salvo Braz, Roxana Girju, Vasin Punyakanok, Dan Roth, and Mark Sammons. An inference model for semantic entailment in natural language. In *AAAI05*, Illinois, USA, 2005.

[Diego Molla, 2004] (editor) Diego Molla. *ACL Workshop on Question Answering in Restricted Domains*. Spain, 2004.

[Diego Molla, 2005] (editor) Diego Molla, Jose Luis Vicedo. *Special Issue of Computational Linguistics on Question Answering in Restricted Domains*. 2005.

[Doan-Nguyen and Kosseim, 2006] Hai Doan-Nguyen and Leila Kosseim. Using Terminology and a Concept Hierarchy for Restricted Domain Question Answering. In *Research on Computing Science, Special issue on Advances in Natural Language Processing*, 2006.

[Durme *et al.*, 2003] Benjamin Van Durme, Yifen Huang, Anna Kupsc, and Eric Nyberg. Towards light semantic processing for Question Answering. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 54–61, NJ, USA, 2003.

[Greenfield, 2004] Jerry Greenfield. Readability Formulas for EFL. In *JALT Journal*, 2004.

[Katz and Lin, 2003] Boris Katz and Jimmy Lin. Selectively Using Relations to Improve Precision in Question Answering. In *Proceedings of the EACL 2003 Workshop on Natural Language Processing for Question Answering*, Hungary, 2003.

[Lin, 1993] Dekang Lin. Principle-based Parsing without Overgeneration. In *Proceedings of ACL-93*, pages 112–120, USA, 1993.

[Molla and Gardiner, 2004] Diego Molla and M. Gardiner. AnswerFinder - Question Answering by combining lexical, syntactic and semantic information. In *Proceedings of Australasian Language Technology Workshop (ALTW)*, pages 9–16, Sydney, 2004.

[Molla, 2003] Diego Molla. Towards semantic-based overlap measures for Question Answering. In *Proceedings of Australasian Language Technology Workshop (ALTW)*, Melbourne, 2003.

[Nyberg *et al.*, 2003] E. Nyberg, T. Mitamura, J. Callan, J. Carbonell, R. Frederking, K. Collins-Thompson, L. Hiyakumoto, Y. Huang, C. Huttenhower, S. Judy, J. Ko, A. Kupse, L. Lita, V. Pedro, D. Svoboda, and B. Van Durme. The JAVELIN Question Answering system at TREC 2003: A Multi-Strategy approach with dynamic planning. In *Proceedings of the Twelfth Text REtrieval Conference (TREC-12)*, USA, 2003.

[Pedersen *et al.*, 2004] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet::Similarity - measuring the relatedness of concepts. In *Proceedings of the Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04)*, USA, 2004.

[Raina *et al.*, 2005] Rajat Raina, Aria Haghighi, Christopher Cox, Jenny Finkel, Jeff Michels, Kristina Toutanova, Bill MacCartney, Marie-Catherine de Marneffe, Christopher D. Manning, and Andrew Y. Ng. Robust textual inference using diverse knowledge sources. In *PASCAL: Proceedings of the First Challenge Workshop on Recognizing Textual Entailment*, pages 57–60, UK, 2005.

[Voorhees and Tice, 1999] Ellen Voorhees and Dawn Tice. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, 1999.