Comparing the Contribution of Syntactic and Semantic Features in Closed versus Open Domain Question Answering

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

Abstract

In this paper we analyze the contribution of semantic, syntactic and word similarity of document features in closed and open domain question answering. Semantic similarity is computed as the similarity of the action in the candidate sentence to the action asked in the question, measured using WordNet::Similarity on main verbs. The syntactic similarity feature measures the unifiability of a candidate's parse tree with the question's parse tree. It uses syntactic restrictions as well as lexical measures to compute the unifiability of critical syntactic participants in the parse trees. Finally, the word similarity of the document containing a candidate sentence is computed as the cosine of the angle between the question keywords vector and the document vector. Since the semantic feature is more reliable on content verbs and syntactic similarity is suitable for questions with a subject-verb-object syntactic structure, we only consider questions with a main content verb in our analysis (non-copulative questions). This type comprise 70% *of our closed domain and* 33% *of our open domain test questions. The combination of these three features achieves an MRR of* 28% *in our closed domain and* 23% *in open domain.*

Our analysis shows that the syntactic feature has a significant contribution in both open and closed domains. However, the path-based lch *semantic similarity measure we used, only contributes in our closed domain probably because of less variation in the vocabulary and topic. Document IR score on the other hand, has more contribution in open domain, because query keywords are more discriminating in a large document set with a vast vocabulary range.*

1. Introduction

QA can be regarded as the next step beyond search engines that returns a ranked list of actual answers to a question. Given a collection of documents (such as the Web or a local collection), the system should be able to retrieve answers to questions posed in natural language.

Given a set of candidate sentences containing a number of question keywords, deciding which one actually answers the question is a challenging problem that a question answering system must solve.

Open domain question answering deals with questions of unrestricted topics, and can therefore only rely on general linguistic resources and world knowledge. On the other hand, these systems usually have much more data available from which they can extract the answer (typically the Web) and can therefore use redundancy of the candidate answers as an additional feature.

In this paper, we address the problem of finding the best candidate sentence to a question by measuring their similarity. We combine three features for this purpose: the semantic similarity of the main verbs, the syntactic overlap of counterpart subtrees, and the word similarity of the document containing the candidate sentence.

While most work in QA has been done on open domain, some systems are expert in only a specific domain (for example, touristic information [1] or the construction sector [16]). For our experiments in closed domain, we build a set of customer service question/answer pairs from Bell Canada's Web pages. For open domain, we used the question/answer sets provided by the NIST organization through the TREC QA conferences. We will show that our features are domain independent and achieve competitive performance in closed domain and open domain questions with a main content verb.

2. Previous Work

In open domain, some statistical systems such as Aranea [11], use lexical patterns built by reformulating the question words to filter sentences that seem to be irrelevant to the question and boost the ones that are more

similar to the answer sentence. The IBM statistical QA system [5] maximizes the word-by-word overlap between question words and answer words. Statistical systems return the most frequent phrase in the set that is of the expected answer type as the answer. Linguistic systems such as Sapere [6] however, take into consideration the syntactic relation of key words to make a more informed decision on the relevance of a candidate sentence to the question. Salvo et al. [2] construct and map concept graphs of the question and candidate sentences and obtain very good results in open domain.

Pure linguistic criteria for measuring the similarity of sentences, however, impose very strict syntactic constraints that result in high precision, but low recall. This problem has been observed by researchers in the field, such as Renxu Sun et al. [14]. Some statistical systems, such as in the work of Milen Kouylekov et al. [7] and Nyberg et al. [4], have tried to relax syntactic constraints; they learn to look for important syntactic links and only score these links. Such methods, however, are very lenient in considering the relative importance of primary roles (such as subject and object) over less important roles (such as a determiner or a modifier). The most interesting effort towards improving this syntactic measure is weighting the matching links (that have similar head, relation and tail) according to their Inverse Document Frequency (IDF) [5]; rare link types have more information content than frequent relation types. This effort has not significantly improved the recall problem; in the end, most parse tree based techniques perform poorly compared to syntactically blind statistical methods.

In closed domain, since redundancy of answers does not exist, statistical QA systems are not likely to perform as well as in open domain. Most current closed domain QA systems, such as Zhang et al. [16] and Benamara et al. [1], embed domain specific knowledge in the form of axioms or structured data in order to improve the QA accuracy. However, this prevents these QA systems to be portable to other domains.

3. Scoring Candidate Sentences

In this section, we present the features we use in scoring the candidate answers of a question: semantic, syntactic and word similarity of the containing document to the question.

We first parse the question and the candidate sentence and choose an appropriate subtree in the parse tree of the candidate sentence to be mapped on to the parse tree of the question. We compute the value of the syntactic and semantic features from this mapping.

Essentially, we believe that the best subtree in the candidate parse tree is the one that has a similar verb to the question's main verb as well as equivalent arguments that map on the arguments of the question's main verb. Performing this test on every combination of verbs will find the most appropriate subtree of the candidate sentence (the target subtree). However, for long sentences having multiple verbs (which are numerous in closed domain), computing multiple verb similarities is expensive in terms of processing time. To avoid going through this comprehensive test, we exploit another feature for locating the target subtree: a strong verb similarity should co-occur with an essential entity match (question head) in the target subtree. Therefore, we consider the question head as an anchor to locate relevant subtrees. We mark the one(s) that has a semantically similar main verb to the question as the target subtree(s), as explained in the following section.

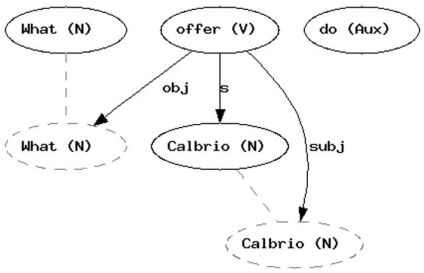

Figure 1. Parse tree of the question *"What does Calbrio offer?"*

3.1. Semantic Similarity of the Candidate

Since the main action specified in a non-copulative question is typically realized by a verb, our first step is to verify the semantic relatedness of the question's main verb to the verb in the target subtree.

WordNet::Similarity [13] provides six measures of similarity which use the information found in *hypernymy*, *hyponymy*, *holonymy* and *meronymy* relations between nouns, and *hypernymy*, *troponym* and *entailment* relations between verbs to quantify how much concept (verb) A is similar to concept (verb) B. Three of the six measures of similarity are based on the information content of the least common subsumer (LCS) of concepts A and B. Information content is a measure of the specificity of a concept, and the LCS of concepts A and B is the most specific concept that is an ancestor of both A and B. These measures include *res*, *lin*, and *jcn* [13].

The *lin* and *jcn* measures augment the information content of the LCS with the sum of the information content of concepts A and B themselves. The *lin* measure scales the information content of the LCS by this sum, while *jcn* takes the difference of this sum and the information content of the LCS. The other three similarity measures are based on path lengths between a pair of concepts: *lch* (Leacock and Chodorow), *wup* (Wu and Palmer), and *path*. The *lch* measure finds the shortest path between two concepts, and

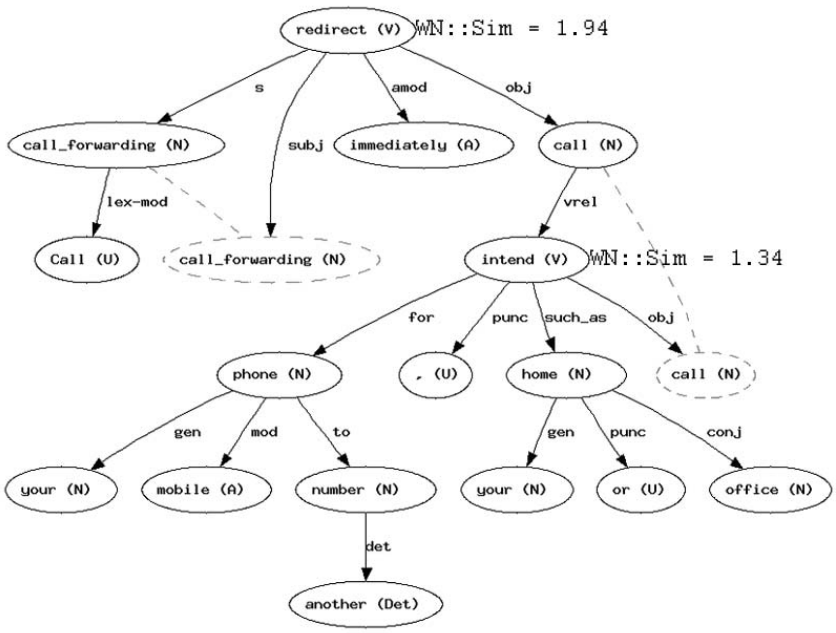

Figure 2. Finding a semantically similar verb in the sentence *"Call Forwarding immediately redirects calls intended for your mobile phone to another number, such as your home or office."* **to the question's main verb** *'work'***.**

scales that value by the maximum path length found in the *is-a* hierarchy in which they occur. The *wup* measure finds the depth of the LCS of the concepts, and then scales that by the sum of the depths of the individual concepts. The depth of a concept is simply its distance to the root node. The measure *path* is a baseline that is equal to the inverse of the shortest path between two concepts.

We chose the *lch* (Leacock and Chodorow) similarity measure from these six WordNet::Similarity measures for scoring the relatedness of two verbs. This measure quantifies the best if two verbs are synonyms in both our closed and open domain development sets.

As an example, the word *'Calbrio'* in the question *"What does Calbrio offer?"* (see Figure 1) is the only noun phrase in the question. It is found in the left subtree of the verb *'provide'* in the candidate sentence shown in Figure 3. Moving up from this anchor, skips the noun node *'system'* and marks the subtree at the verb node *'provide'* as the target subtree. 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. As another example, Figure 2 shows the parse tree of a candidate answer for the question *"How does Call Forwarding work?"*. The verb *'redirect'* has a similarity of 1.94 to the verb *'work'*, the question's main verb. This subtree (which happens to be the whole parse tree in this example) includes the question head (*"Call Forwarding"*); so, we can proceed with unification by checking whether their main verbs relate the same entities (subject and object in particular).

3.2. Syntactic Similarity of the Candidate

Having identified the relevant subtree of the candidate sentence (the target subtree), our syntactic feature catches approximate syntactic similarity of this subtree to the question. This feature is robust to minor syntactic differences and parsing errors; we consider a wrong modifier or propositional attachment as minor errors while a misplaced subject or object relation is considered a major parsing error, because entities in these two roles play the most important roles in conveying the meaning of the sentence. For this purpose, we strictly map subject, object and attaching subtrees in the target to their counterparts in the question. The syntactic similarity of two subtrees (one from the question and the other from the candidate sentence) is computed as:

$$UnificationScore(Q_i, T_i) = \beta \times WordOverlap + (1-\beta) \times LinkOverlap \quad (1)$$

In this formula, the number of matching *word stems* in the subtrees introduce a strong linguistic feature, while the number of matching *syntactic links* boosts the syntactic unification score. In this way, we contribute syntax while not being dependent on having a perfect parse tree match. In the following sections we describe how we compute these two features.

3.2.1 Weighted Bag-of-Words Overlap

Common words are a strong hint to identify equivalent phrases. For example, the noun phrase *"the American Legion"* in the sample question shown in Figure 4, should be

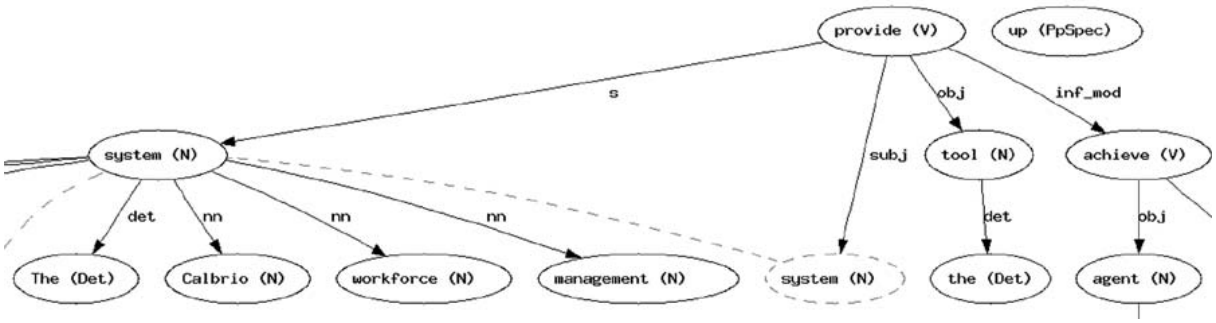

Figure 3. Parse tree of the sentence *"The Calbrio workforce management system, made up of fully integrated software modules, provides the tools to achieve optimal agent staffing ..."*

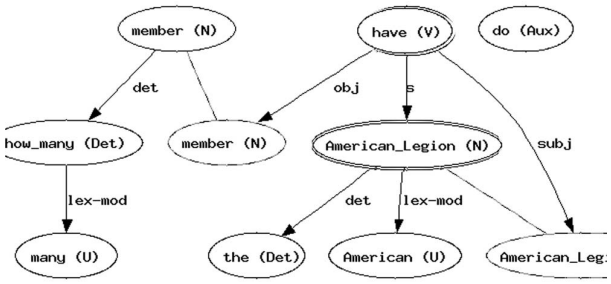

Figure 4. Parse tree of the question *"How many members does the American Legion have?"*

Category	Part of Speech	Weight
Proper Noun	NNP	3.0
Common Noun	NN, NNS	1.0
Verb	VB, VBD, VBN, VBZ, VBG	0.75
Adjective	JJ, JJS	0.5
Adverb	RB, RBS	0.25

Table 1. Part of Speech weights used in ranking the keywords.

unified with *"the spokesman for the American Legion"* in the answer sentence depicted in Figure 5. When weighting the question keywords, we give more importance to proper nouns, nouns and verbs, and consider a lower score for adjectives and adverbs in order to emphasize on distinctive words with a significant and unique meaning.

The values we determined experimentally with our closed domain training set for each category are shown in Table 1. For our last example, a score of 6.25 is returned by matching the words ('the'=0.25, 'American'=3.0 and 'Legion'=3.0).

3.2.2 Weighted Syntactic Links Overlap

As equation 1 shows, in addition to computing the lexical similarity of two subtrees, we measure the syntactic similarity of two phrases to boost their unification score if the matching words have similar relations to one another.

Expecting strict syntactic similarity results in low recall when facing syntactic paraphrases: *"president of Russia, Jeltsin"* and *"the Russian president, Jeltsin"* for example are two semantically equivalent phrases, with different syntactic relations between perfectly matching words. Therefore, as long as there is word overlap, we unify the two subtrees. Note that syntactic relations inside arguments are not critical like the subject and object syntactic relations, and will only boost the unification score.

By analyzing a few unification cases, we realized that matching different types of links should have a variable contribution to the final unification score. Compare a modifier (*'mod'*) link matching in the candidate *"wireless network"* as opposed to a determiner (*'det'*) link in the candidate *"a network"* matching with the phrase *"... a wireless network ..."* in the question. The first case shows a stronger similarity since it narrows down the meaning of the noun (*'network'*). To account for this, we weight links differently: i.e., a lexical modifier link (shown as "lex-mod" in the Minipar [10] parser) has the highest weight of 1.0 because it connects two proper nouns, while a determiner has the lowest score. Table 2 shows the classes of equivalent links we selected and the values we obtained experimentally for each class. These values can also be learned given a tagged set of equivalent, but syntactically different phrases, such as an appropriately selected subset of the *"Equivalent sentence pairs with minor differences in content"* from the Microsoft Research Paraphrase corpus [3]. In Section 4, we perform a sensitivity analysis for these valuses and show that our scor-

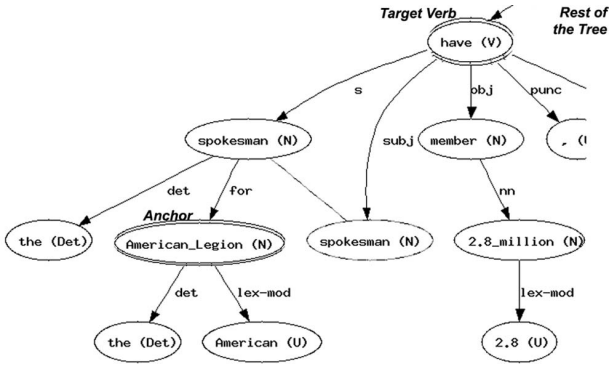

Figure 5. Parse tree of the sentence *"...said Phil Budahn, spokesman for the American Legion, which has 2.8 million members."*

Category	Minipar Relation	Weight
Lexical modifier	lex-mod	1.0
Adjective/Nominal modif	mod, pnmod, pcomp-n, nn	0.5
(pre)Determiner	(pre)det	0.25
Possessives	gen	0.25

Table 2. Weights of different syntactic links used in scoring the similarity of two phrases.

ing method is not very sensitive to them, as long as the order of the classes of syntactic relations is preserved.

For the previous example (Figure 5), the value of the $LinkOverlap$ feature will therefore be $1.0 + 0.25 = 1.25$ (for the lexical modifier link in *"American Legion"* and the determiner link in *"the Legion"*).

3.2.3 Combining the Syntactic Sub-Features

The parameter β in equation 1 shows the relative importance of the two parts in computing the syntactic similarity feature: $\beta = \frac{1}{3}$ (our configuration) considers the link overlap feature to be twice as important as the bag-of-words feature. Note that the $LinkOverlap$ feature subsumes $WordOverlap$ because the head and tail words need to match in order to have a link match. The purpose of this feature is to reflect the syntactic similarity of matching words. Also 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. This combination results in a total score of $\frac{1}{3} \times 6.25 + (1 - \frac{1}{3}) \times 1.25 = 2.89$ for the matching of these two phrases (subject subtrees) in the previous example.

3.3. Word Similarity of the Document

Most QA systems use an off-the-shelf IR system to find relevant documents from the document collection. The IR engine typically scores the candidate documents it returns based on their lexical similarity to the query (question keywords). Research has shown that this score is useful for question answering [12]. For information retrieval in both open and closed domain, we use the Lucene IR engine[1]. It is based on the vector space model and the score of a document is computed as the cosine of the angle between the question keywords vector and that document's vector. A cosine value of zero means that the question and document vector are orthogonal and have no match (i.e. the question keywords do not exist in the document being considered). We use this useful information as a feature in our candidate scoring.

3.4. Combining the Similarity Features

To score a candidate sentence, we combine the syntactic and semantic similarity scores in addition to the IR score of the document that the candidate is selected from:

$$\begin{aligned} & w_1 \times \Sigma_{i:Subtree} UnificationScore(Q_i, T_i) \\ & + w_2 \times WN :: Similarity(Verb_Q, Verb_T) \\ & + w_3 \times Score_{IR}(Candidate) \end{aligned} \quad (2)$$

where, (w_1, w_2, w_3) represent the weights of each feature. They are set to (1, 1, $\frac{1}{3}$) in our best setting. Experimenting with different weights can show the importance of each feature. We postulate that the semantic similarity of the question (q) and the candidate's (T) main verb should have more importance; however, in order to receive a good similarity measure, verb senses must be specified; this is very difficult to achieve with current state of the art word sense disambiguation algorithms [8]. Even varying the weights w_i from 0 to 1 does not find an optimal value for w_2. We fix minimum thresholds (obtained experimentally) for these features in order to guarantee minimum semantic and syntactic similarity for the candidates that get a score: $WN :: Similarity(Verb_Q, Verb_T) > T_{semantic} = 1.8$ and $\Sigma_{i:Subtree} UnificationScore(Q_i, T_i) > T_{syntactic} = 0.8$.

[1] Apache Software Foundation, Lucene 1.4.3 API `http://lucene.apache.org/java/docs/api/`

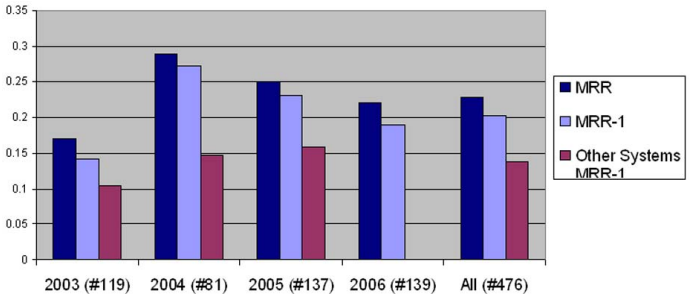

Figure 6. Comparison of the performance of our scoring method with other TREC QA participants.

4. Evaluation

The evaluation was performed on two question sets: the Bell customer service questions and TREC QA questions. To create our closed domain collection, we asked 15 students to formulate questions and answers given a corporate document collection of Bell Canada. The document collection consists of 340 Web pages, internal documents and a few technical manuals (750KB of text). The questions produced vary in style, length and complexity; most are long and complex 'what' and 'how' type questions. We randomly chose 100 questions for development and kept the remaining 120 for testing.

The performance of our question answering system is reported by the standard Mean Reciprocal Ranking (MRR) [15]. The MRR is equal to the inverse of the position of the first correct answer in the result:

$$MRR = \frac{1}{Rank(first\ correct\ answer)} \qquad (3)$$

In closed domain our method achieves an overall MRR of 28% for 120 test questions [9]. For open domain, Figure 6 shows an MRR of 23% for the best combination of our three features. The "Other Systems" column in this figure shows how other TREC participants performed on the questions with a main content verb (476 non-copulative questions in the last four TRECs): the average MRR-1[2] of all submissions for these questions is 13.7% compared to an MRR-1 of 20.3% (2.7% less than the MRR) for our candidate scoring method.

In the rest of this section, we analyze the usefulness of each scoring feature as well as the sensitivity of the optimum weights we experimentally assigned for question answering in our closed domain and the TREC questions.

Table 3 shows the MRR achieved in our closed and open domain by adding or removing each feature. As the table shows, using only the syntactic similarity measure achieves an MRR of 16% above the baseline in closed domain and 10% in open domain. This is significant in both domains. This feature helps more in our closed domain possibly because of the annotators bias towards composing questions in a similar syntactic structure as the answer sentence.

In closed domain, adding semantic similarity information to choose the target subtree (changing w_2 from 0 to 1) increases the MRR by approximately 2%. In closed domain, similar nouns appear with multiple verbs (ex. "...*provided* in POP3", "...*introduce* POP3", "customers who *use* POP3", etc.) and even an average performing similarity measure can reject a considerable number of irrelevant sentences. Additionally, the fact that variation of verbs is limited to a single topic helps in getting a more reliable semantic similarity measure from WordNet::Similarity in closed domain. Finally, adding the document score feature (changing w_3 from 0 to its optimal value of $\frac{1}{3}$) slightly increases the MRR by 2%. We expected more contribution from this feature; however, query words are not as distinctive as in open domain, since they appear in many documents and usually more than once (especially frequent nouns and proper nouns in our corpus, such as 'Bell', 'Sympatico', 'wireless', 'cell', 'phone', etc.). Therefore, keyword statistics are not very helpful.

In open domain, initially the syntactic measure results in an MRR of 10% above the baseline (for non-copulative questions). It is interesting to note that so many non-copulative questions have the same verb and strong syntactic similarity to their candidate answer. By using semantic similarity to choose the target subtree, we hardly gain any improvement in MRR: the *lch* measure is unreliable and introduces as much noise in the candidate set. We suspect that this is not a problem in closed domain, because the variation of verbs is less and limited to the Telecommunications topic, compared to open domain, in which verbs are about almost any topic with even different senses in different contexts. Finding semantic similarity of two general verbs is harder than when they are from a closed set and related to a given topic.

Adding the document score to sentence scores, increases the MRR by 3%. Compared to closed domain, information retrieval has more contribution in open domain (3% versus 2%). It again confirms that statistical word frequency information is more helpful when having a large document collection.

In order to analyze the sensitivity of the syntactic features, we changed the *part of speech* and *parse link* weights by 50% from the values we used in Tables 1 and 2. As long as varying these values preserves the order of categories, the MRR decreases by only 4%. However, changing the order that we linguistically justified, will drastically lower the final results.

[2]MRR-1 is a variation of the MRR measure introduced in 2003 that considers only the top answer for scoring: if this answer is correct, system receives a score of 1 and 0 otherwise.

		Closed Domain (#120)		Open Domain (#476)	
Feature	Weight	MRR	Contribution	MRR	Contribution
Baseline (random selection)		8%		10%	
Syntactic Unification Score	$w_1 = 1$	24%	+16%	20%	+10%
Semantic WN::Similarity Score	$w_2 = 1$	26%	+2%	20%	+0%
Lucene IR Score	$w_3 = \frac{1}{3}$	28%	+2%	23%	+3%

Table 3. Contribution of individual features in the final QA performance (MRR).

5. Conclusion and Future Work

In this paper we analyzed the contribution of three features in ranking candidate sentences in our Telecommunications closed domain as well as in the TREC open domain. Since the syntactic and semantic features rely on the semantic similarity of the question's main verb with the verb in the target subtree in the candidate, it does not perform as well on copulative questions. Additionally, questions with a 'to be' main verb have a relatively free syntactic structure that makes the modeling of mapping rules (between a copulative question and candidate sentences) in order to compute the syntactic similarity feature very difficult. The accuracy levels we achieved in both domains for answering questions with a main content verb, though, show the usefulness of syntactic mapping and semantic information.

We showed that the syntactic feature has significant contribution in both open and closed domains; possibly because of the annotators bias towards composing questions in a similar syntactic structure as the answer sentence, this feature helps more in our closed domain. Verb similarity contributes more in closed domain probably because there is less variation in the vocabulary and topic. Document similarity score on the other hand, contributes more in open domain because query keywords are more discriminating.

The scoring method we described in this paper was developped for answering non-copulative questions. We believe that different types of questions should be answered using different strategies. Ideally, one could use our scoring method on non-copulative questions and use a different strategy for copulative questions, hence improving the overall performance.

References

[1] F. Benamara. Cooperative Question Answering in Restricted Domains: the WEBCOOP Experiment. In *ACL Workshop: Question Answering in Restricted Domains*, Spain, 2004.

[2] R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. An inference model for semantic entailment in natural language. In *AAAI05*, Illinois, USA, 2005.

[3] B. Dolan, C. Brockett, and C. Quirk. Microsoft research paraphrase corpus, 2005.

[4] B. V. Durme, Y. Huang, A. Kupsc, and E. 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.

[5] A. Ittycheriah, M. Frans, and S. Roukos. IBM's Statistical Question Answering System. In *Proceedings of TREC-10 Conference*, pages 258–264, Gaithersburg, Maryland, 2001.

[6] B. Katz and J. 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.

[7] M. Kouylekov and H. Tanev. Document filtering and ranking using syntax and statistics for open domain question answering. In *Proceedings of ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP*, pages 21–30, Nancy, France, 2004.

[8] A. K. Lamjiri, O. E. Demerdash, and L. Kosseim. Simple features for statistical word sense disambiguation. In *Proceedings of the 3rd International ACL Senseval Workshop*, pages 37–44, Barcelona, Spain, 2004.

[9] A. K. Lamjiri, L. Kosseim, and T. Radhakrishnan. A hybrid unification method for question answering in closed domains. In *Proceedings of the 3rd International IJCAI KRAQ'07 Workshop*, pages 36–42, Hyderabad, India, 2007.

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

[11] J. Lin and B. Katz. Question Answering from the Web Using Knowledge Annotation and Knowledge Mining Techniques. In *Proceedings of CIKM'03*, pages 116 – 123, Louisiana, USA, 2003. ACM.

[12] C. Monz. *From Document Retrieval to Question Answering*. PhD thesis, University of Amsterdam, Netherlands, 2003.

[13] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity - measuring the relatedness of concepts. In *Proceedings of the NAACL-04*, Boston, USA, 2004.

[14] R. Sun, J. Jiang, Y. F. Tan, H. Cui, T.-S. Chua, and M.-Y. Kan. Using Syntactic and Semantic Relation Analysis in Question Answering. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC-13)*, Gaithersburg, MD, 2004.

[15] E. Voorhees and D. Tice. The TREC-8 Question Answering Track Evaluation. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 83–106, Gaithersburg, MD, 1999.

[16] Z. Zhang, L. D. Sylva, C. Davidson, G. Lizarralde, and J.-Y. Nie. Domain-Specific QA for the Construction Sector. In *Information Retrieval for Question Answering (IR4QA) Workshop in 27th ACM-SIGIR*, pages 64–70, UK, 2004.