FALCON: Boosting Knowledge for Answer Engines

Sanda Harabagiu, Dan Moldovan,
Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Răzvan Bunescu, Roxana Gîrju,
Vasile Rus and **Paul Morărescu**
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
{sanda,moldovan}@seas.smu.edu

Abstract

This paper presents the features of FALCON, an answer engine that integrates different forms of syntactic, semantic and pragmatic knowledge for the goal of achieving better performance. The answer engine handles question reformulations, finds the *expected answer type* from a large hierarchy that incorporates the WordNet semantic net and extracts answers after performing unifications on the semantic forms of the question and its candidate answers. To rule out erroneous answers, it provides a justification option, implemented as an abductive proof. In TREC-9, FALCON generated a score of 58% for short answers and 76% for long answers.

Introduction

The design of open-domain answer engines is guided by two thrusts. First, natural language processing (NLP) methods are used to derive the questions semantics, in order to identify the candidate answers in the text collections. These methods are integrated with specially crafted information retrieval (IR) techniques that return all *text paragraphs* of interest. Second, to be able to extract the correct answers, bag-of-words approaches are not always sufficient. They are replaced by surface-based NLP methods that are boosted with pragmatic knowledge that filters out incorrect answers.

In (Moldovan et al.1999),(Moldovan et al.2000) we have presented the surface-based NLP techniques that made possible *question processing*, *paragraph indexing* and the *answer processing* and their interaction with the paragraph search and the named entity recognizers. In (Harabagiu et al.2000) we report initial experiments that integrate knowledge engineering techniques with NLP methods for Question&Answering (Q&A). This additional knowledge allowed our answer engine to process a vast majority of open-domain questions, extending the initial cases covered by the TREC-8 questions. Furthermore, by generating semantic and logical forms of questions and answers we enabled a justification option based on abductive proofs.

The boosting methodology relies on several new sources of pragmatic knowledge. First, we considered that it is likely that an answer engine would be presented with reformulations of previously posed questions. Thus we devised an approach of recognizing *question reformulations* and *caching* their corresponding answers. Secondly, we designed a new paragraph retrieval mechanism that enables *keyword alternations*, such that paraphrases of question concepts and even some related concepts be included in the search for the textual answer. Finally, in order to extract and evaluate answer correctness, we replaced bags-of-words approaches with loose unifications of the semantic forms for questions and answers.

Instead of operating at word level, we have escalated our extraction methods to operate at the level of dependencies between words, thus better approximating the semantics of questions and answers. These dependencies also rely on name entity recognizers that incorporate significantly larger numbers of name categories than those currently employed in the Information Extraction (IE) technology. Without any loss of robustness and without downgrading the elegance of our answer engine, we enable the representation of questions and answers into semantic forms based on information brought forward by fast, wide-coverage probabilistic parsers. Furthermore, by translating the semantic forms into logical forms, we enable a justification option relying on minimal abductive knowledge. The proof mechanism is easily extensible for special domains or situations.

The Knowledge Features in FALCON

To find the answer of an open-domain natural language question in a large collection of texts we need to devise an efficient way of identifying text passages where the answer may lie, followed by a mechanism of extracting only correct answers, or alternatively, notify the absence of an answer in the collection. However, before initiating the search, it is very possible that the

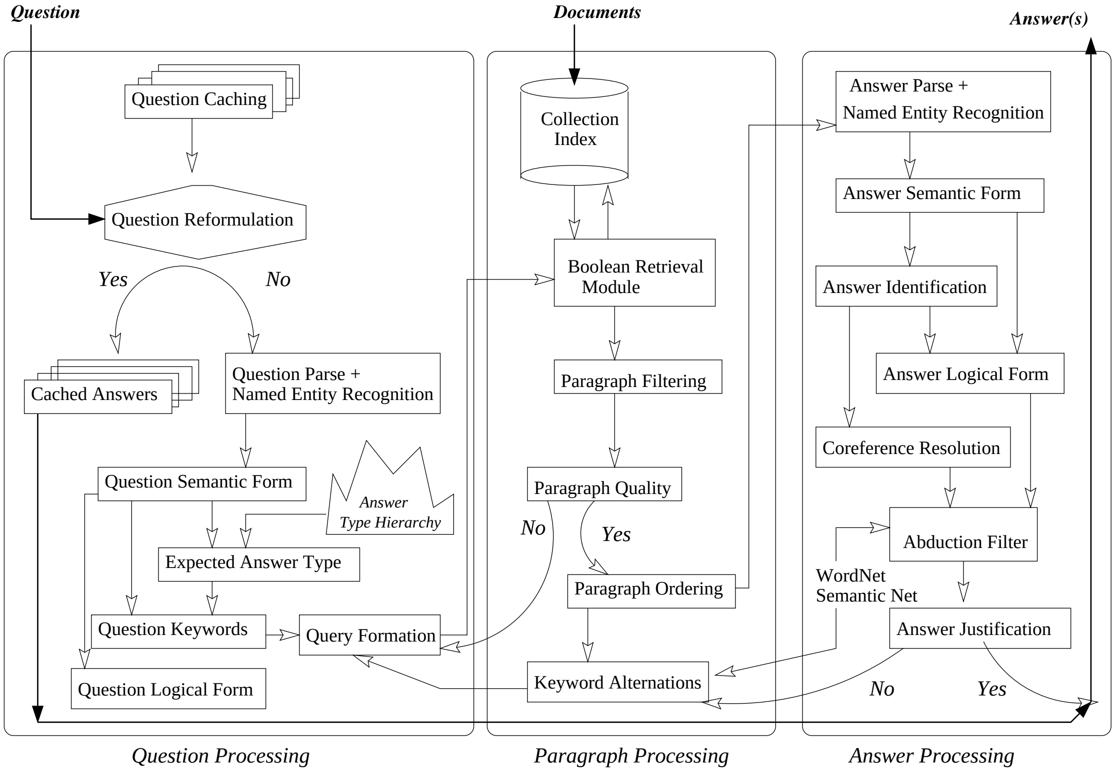


Figure 1: Question, Paragraph and Answer Processing in FALCON

same question or a very similar one has been posed to the system before, and thus those results can be used again. To find such *cached questions*, we measure the similarity to the previously processed questions and when a reformulation is identified, we consider all *question reformulations* and their corresponding answers.

The search for answers is based on the conjecture that the eventual answer is likely to be found in a text paragraph that (a) contains the most representative question concepts and (b) includes a textual concept of the same category as the expected answer. Since the current retrieval technology does not model semantic knowledge, we have to break down this search into a boolean retrieval, based on some question keywords and a filtering mechanism, that retains only those passages containing the expected answer type. Both the *question keywords* and the *expected answer type* are identified in the *question processing* module of FALCON, illustrated in Figure 1.

Finding the expected answer type of a natural language question by relying only on the semantics of the question stem (i.e. *What*, *How*) and some bag-of-words approaches is not always possible, since stems are associated with many different types of answers and shallow syntax (e.g. phrasal parsers) fails to find the most discriminating concept in the question. The syntactic dependencies between the question phrases help solve this ambiguity: the answer type is indicated by the question phrase most connected to other concepts. The question semantic form is used to search for this concept and map it into the answer taxonomy that contains WordNet subhierarchies, and thus covers a large majority of English words. The question semantic form is used as a knowledge source once more when the answer is eventually identified, to measure the answer plausibility.

Using large, publicly available resources such as WordNet (Miller 1995) makes possible the open-domain semantic processing of questions. Moreover, it extends the question processing centered around named entity recognition to deal with the semantic class of any concept lexicalized in English and encoded in WordNet. For example, FALCON identifies the expected answer type of the TREC-9 question Q257: *What do penguins eat?* to be *food*, since it is the most widely used concept in the glosses of the subhierarchy of the noun synset {*eating, feeding*}. Moreover, the mapping in the answer hierarchy does not presuppose word sense disambiguation, but a mere search along WordNet hierarchies to find a top of the answer tax-

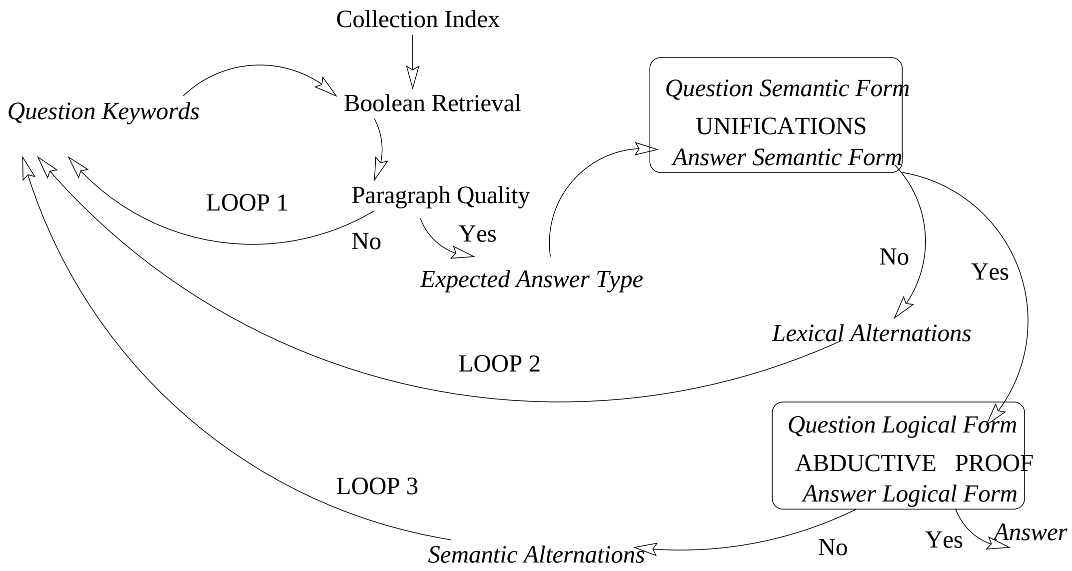


Figure 2: Refining the Answer Search by Boosting Knowledge into FALCON

onomy.

The question keywords are structured into a query that is passed to a boolean retrieval engine, implemented by adding boolean operators to the SMART IR system (Buckley et al.1998). Additionally, we have post-processed the results of the IR engine, by retrieving only text paragraphs defined by the presence of the query keywords within a window of pre-defined size (e.g. 10 lines from the document). Overall, the quality of the paragraphs is measured by the number of returned paragraphs. It is well known that one of the disadvantages of boolean retrieval is that it returns either too many or too few documents. However, for question answering, this is an advantage. We defined the quality of the paragraph retrieval to be acceptable if the number of paragraphs returned is between a lower and an upper bound defined a-priori for each answer type. If the paragraph retrieval is within limits, the paragraphs are ordered and passed to be processed in the answer processing module. Otherwise, as detailed in (Moldovan et al.2000), if too many paragraphs were retrieved, new keywords would be added to the query. Alternatively, when too few paragraphs are returned, some keywords would be dropped and the retrieval reinitiated, generating the first loop represented in Figure 2. By boosting knowledge in the answer engine, we have generated a retrieval model that is based on the three loops represented in Figure 2.

In the answer processing module, each paragraph is parsed and transformed into a semantic form, that comprises its name categories as well. When unifications between the question and the answer semantic forms are not possible for any of the paragraphs, the answer cannot be found in any of the retrieved data, thus alternations of the question keywords, comprising synonyms and morphological derivations are considered and sent to the retrieval engine. The new paragraphs returned are evaluated, generating the second loop represented in Figure 2.

Finally, an answer is extracted only if a logical justification of its correctness can be provided. For this purpose, the semantic forms of questions and answers are translated into logical forms, comprising predicates whose arguments localize the dependencies. The logical proof is implemented as an abductive backchaining from the answer to the question. As the paragraph span few syntactic dependencies and a limited number of content words, the backchaining space is quite small. The number of coreference relations relevant to the paragraph is generally very small if not null, and the world knowledge encodes very few semantic relations from the WordNet semantic net. The abductive filter incurs very little additional processing. When no answer can be justified, related concepts are searched in WordNet to provide with some semantic alternations that guide new paragraph retrievals. For example, in the case of TREC-9 question *Q210: How many dogs pull a sled in the Iditarod?*, although unifications between the semantic forms of the question and multiple answers were possible, none of the paragraphs retrieved after searching for *Iditarod AND dog AND sled* could be justified, as they were not correctly answering the question. When keyword *sled* was replaced by keyword *harness*, a concept mined from WordNet, the correct answer could be found: *Race rules require mushers to arrive in Nome with at least five dogs in harness.* The semantic alternations allow for the third loop represented in Figure 2.

The complex search for an answer is guided by different knowledge sources that enable three different search loops. The advantage of boosting knowledge into FALCON is that it opens new search spaces at each new iteration. The validation of the answer is based on increasing levels of knowledge sophistication - from merely identifying paragraphs that contain question concepts to justifying answer correctness.

Question Reformulations

In TREC-9 243 questions were reformulations of 54 inquiries, thus asking for the same answer. The reformulation classes contained variable number of questions, ranging from two to eight questions. Two examples of reformulation classes are listed in Table 1. To classify questions in reformulation groups, we used the algorithm:

Reformulation_Classes(new_question, old_questions)
1. For each question from old_questions
2. Compute similarity(question,new_question)
3. Build a new similarity matrix $\mathcal{M}$ such that it is generated by adding to the matrix for the old_questions a new row and a new column representing the similarities computed at step 2.
4. Find the transitive closures for the set $\{old\_questions\} \cup \{new\_question\}$
5. Result: reformulation classes as transitive closures.

In Figure 3 we represent the similarity matrix for six questions that were successively posed to the answer engine. Since question reformulations are transitive relations, if at a step n questions Q_i and Q_j are found similar and Q_i already belongs to $\mathcal{R}$, a reformulation class previously discovered (i.e. a group of at least two similar questions), then question Q_j is also included in $\mathcal{R}$. Figure 3 illustrates the transitive closures for reformulations at each of the five steps from the succession of six questions. To be noted that at step 4 no new similarities were found , thus Q_5 is not found similar to Q_4 at this step. However, at step 5, since Q_6 is found similar to both Q_4 and Q_5, Q_4 resultssimilar to all the other quesitons but Q_3.

The algorithm that measures the similarity between two questions is:

Algorithm Similarity(Q, Q')
Input: a pair of question represented as two word strings:
Q: w_1 w_2 ... w_n
Q': w'_1 w'_2 ... w'_n ... w_m
1. Apply a part-of-speech tagger on both questions:
Tag(Q): w_1/tag_1 w_2/tag_2 ... w_n/tag_n
Tag(Q'): w'_1/tag'_1 w'_2/tag'_2 ... w_m/tag'_m
2. Set nr_matches=0
3. Identify quadruples $(w_i, tag_i, w'_j, tag'_j)$ such that if w_i and w'_j are content words then also Lexical_relation(w_i, w'_j) holds and moreover $tag_i \equiv tag'_j$
4. For each quadruple, increase nr_matches
5. Relax the Lexical_relation and goto step 3;
6. If (nr_matches/number of content words $\geq t$) then Q and Q' are similar then else Q and Q' are not similar

Q397:	*When was the Brandenburg Gate in Berlin built?*
Q814:	*When was Berlin's Brandenburg gate erected?*
Q-411:	*What tourist attractions are there in Reims?*
Q-711:	*What are the names of the tourist attractions in Reims?*
Q-712:	*What do most tourists visit in Reims?*
Q-713:	*What attracts tourists to Reims?*
Q-714:	*What are tourist attractions in Reims?*
Q-715:	*What could I see in Reims?*
Q-716:	*What is worth seeing in Reims?*
Q-717:	*What can one see in Reims?*

Table 1: Two classes of TREC-9 reformulations.

	Q1	Q2	Q3	Q4	Q5	Q6	
Q1	0	1	0	1	0	0	
Q2	1	0	0	0	0	0	Step 1: {Q1, Q2}
Q3	0	0	0	0	0	0	Step 2: {Q1, Q2} {Q3}
Q4	1	0	0	0	0	1	Step 3: {Q1, Q2, Q4} {Q3}
Q5	0	0	0	0	0	1	Step 4: {Q1, Q2, Q4} {Q3}
Q6	0	0	0	1	1	0	Step 5: {Q1, Q2, Q4, Q5, Q6} {Q3}

Figure 3: Building reformulation classes with a similarity matrix.

The *Lexical_relation* between a pair of content words is initially considered to be a string identity. In later loops starting at step 3 one of the following three possibile relaxations of *Lexical_relation* are allowed: (a) common morphological root (e.g. *owner* and *owns*, from question *Q742: Who is the owner of CNN?* and question *Q417: Who owns CNN?* respectively); (b) WordNet synonyms (e.g. *gestation* and *pregnancy* from question *Q763: How long is human gestation?* and question *Q765: A normal human pregnancy lasts how many months?*, respectively) or (c) WordNet hypernyms (e.g. the verbs *erect* and *build* from question *Q814: When was Berlin's Brandenburg gate erected?*

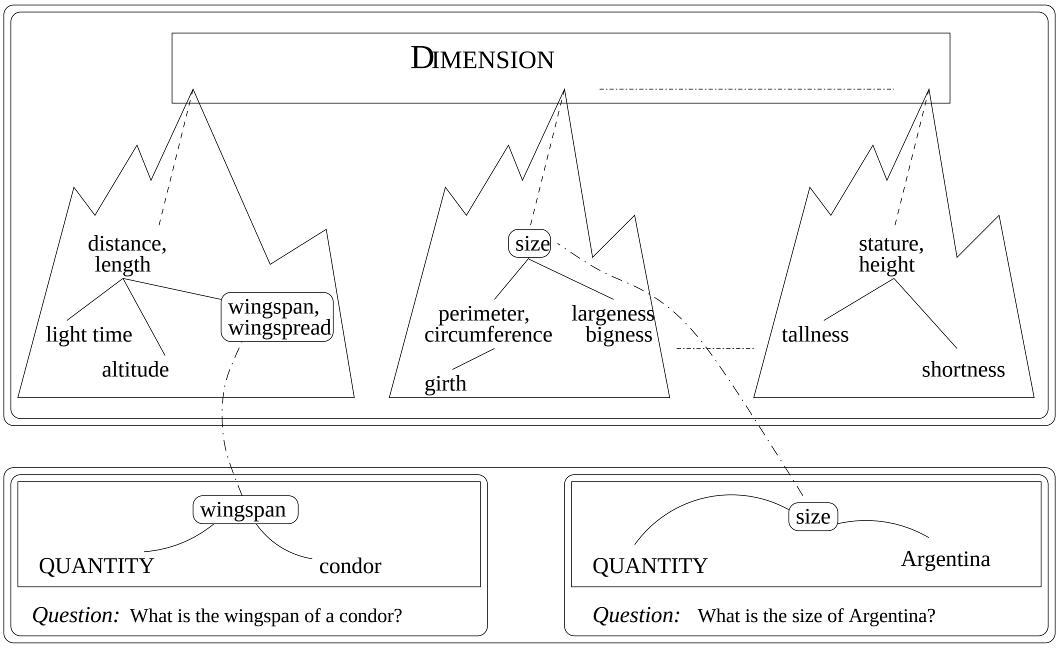


Figure 4: Incorporating WordNet hierarchies into the answer taxonomy

and question *Q397: When was the Brandenburg Gate in Berlin built?* respectively).

The expected answer type

One of the most important pieces of knowledge used in extracting the answer is the semantic category of the expected answer, i.e. the *expected answer type*. The answer semantic categories are mapped in the categories covered by a Name Entity Recognizer that scans paragraphs to identify the eventual answers. For example, the named entity recognizer used in FALCON covers 27 categories, listed in Table 2. For each question, when the question concept indicating the answer type is identified, it is mapped into an answer taxonomy.

date	*time*	*organization*	*town*
product	*price*	*country*	*money*
human	*disease*	*phone number*	*continent*
percent	*province*	*other location*	*plant*
mammal	*alphabet*	*airport code*	*game*
bird	*reptile*	*university*	*dog breed*
number	*quantity*	*attraction*	

Table 2: Named Entity Categories.

Currently, our answer taxonomy has 18 top categories, listed in Table 3. Many of answer tops are further categorized, as illustrated in Figure 5. Table 4 lists the 15 leaves of the current top hierarchies. The leaves of each top hierarchy as well as the stand-alone tops are connected to several word classes from the WordNet database. For example, Figure 4 illustrates the mapping of the *DIMENSION* leaf from the *NUMERICAL VALUE* top hierarchy in several WordNet classes, like *distance*, *size* or *height*. Figure 4 also shows how the mapping from the question to the answer hierarchy takes place. In a question like *Q335: What is the wingspan of a condor?*, the word *wingspan* is searched in the answer type taxonomy, and it is discovered in the *distance* subhierarchy, therefore the assigned category of the expected answer type becomes *DIMENSION* and the named entity recognizer will look for a *QUANTITY*. The selection of the word *wingspan* is enabled by knowledge derived from the semantic form of the question.

DATE	*TIME*	*ORGANIZATION*
REASON	*MANNER*	*NATIONALITY*
PRODUCT	*MONEY*	*LANGUAGE*
MAMMAL	*GAME*	*DOG BREED*
LOCATION	*REPTILE*	*NUMERICAL VALUE*
QUOTATION	*ALPHABET*	*PERCENTAGE*

Table 3: Top categories in the Answer Taxonomy.

It is to be noted that we have a many-to-many mapping between the named entity categories and the leaves of the answer type top hierarchies. Figure 6 illustrates some of the mappings implemented in FALCON. For example, the answer type *MONEY* is searched either as the *money* or as the *price* named

CITY	*COUNTRY*	*PERCENTAGE*
COUNT	*AMOUNT*	*TEMPERATURE*
SPEED	*RATE*	*DURATION*
DEGREE	*PROVINCE*	*DIMENSION*
UNIVERSITY	*CONTINENT*	*OTHER_LOCATION*

Table 4: Leaf nodes of the Top Answer hierarchies.

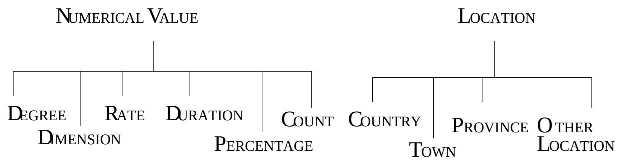


Figure 5: Two examples of top answer hierarchies.

entity category. In contrast, the named entity category *quantity* is used to recognize four types of answers: *SPEED*, *DURATION*, *DIMENSION* and *AMOUNT*.

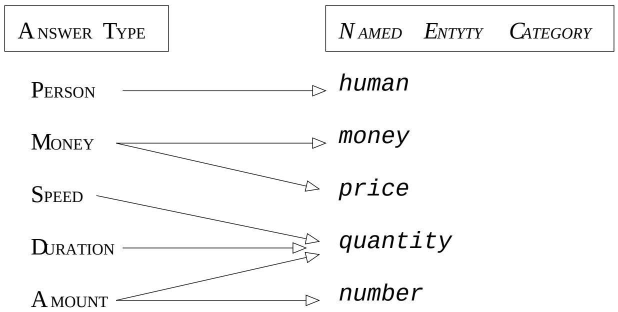


Figure 6: Mappings of answer types in named entity categories.

A special case of answer type is associated with questions that inquire about *definitions*. There are questions having a syntactic format that indicates that the question asks for the definition of a certain concept. Table 5 lists some of the TREC-9 questions that ask for definitions. Such questions are easily identified as they are matched by a set of patterns comprising:

(Q-P1): *What {is|are} <phrase_to_define>?*
(Q-P2): *What is the definition of <phrase_to_define>?*
(Q-P3): *Who {is|was|are|were} <person_name(s)>?*

The processing of questions asking for definitions does not use the expected answer type, but it is rather based on the recognition of the *<phrase_to_define>* and its matching one of the definition answer patterns. Some of the answer patterns are:

(A-P1):[*<phrase_to_define>* {*is*|*are*}]
(A-P2):[*<phrase_to_define>*, {*a*|*the*|*an*}]
(A-P3):[*<phrase_to_define>* –]

When a question does not match a definition pattern, the detection of its answer type is based on the question semantics.

Q228:	*What is platinum?*
Q239:	*Who is Barbara Jordan?*
Q358:	*What is a meerkat?*
Q710:	*What is the definition of hazmat?*

Table 5: Questions asking for definitions.

Semantic Knowledge

Finding the answer to a question by retrieving a text snippet from a large document collection cannot be done unless the question semantics is known. The semantics of the question can be approximated by deriving all dependencies between words, and thus creating a graph of anonymous relations spanning all question concepts. This information is more important than the local syntactic information produced by phrasal parsers. Therefore instead of producing only a phrasal parse for the question and answer, we make use of one of the new statistical parsers for large real-world text coverage (Collins 1996). The parse trees produced by such a parser can be easily translated into a semantic representation that (1) comprises all the phrase heads and (2) captures their inter-relationships by anonymous links. Figure 7 illustrates both the parse tree and the associated semantic representation of a TREC-9 question.

The actual transformation into semantic representation of a question or an answer is obtained as a by-product of the parse tree traversal. Initially, all leaves of the parse tree are classified as *skipnodes* or *non-skipnodes*. All nouns, non-auxiliary verbs, adjectives and adverbs are categorized as non-skipnodes. All the other leaves are skipnodes. Bottom-up traversal of the parse tree entails the propagation of leaf labels whenever the parent node has more than one non-skipnode child. A rule based on the syntactic category of the father selects one of the children to propagate its label at the next level in the tree. The winning node will then be considered linked to all the other former siblings that are non-skipnodes. The propagation continues until the parse tree root receives a label, and thus a semantic graph is created as a by-product. Part of the label propagation, we also consider that whenever all children of a non-terminal are skipnodes, the parent becomes a skipnode as well.

Figure 8 represents the label propagation for the parse tree of the question represented in Figure 7. The labels of *astronaut*, *walk* and *space* are propagated to the next level. This entails that *walk* is linked

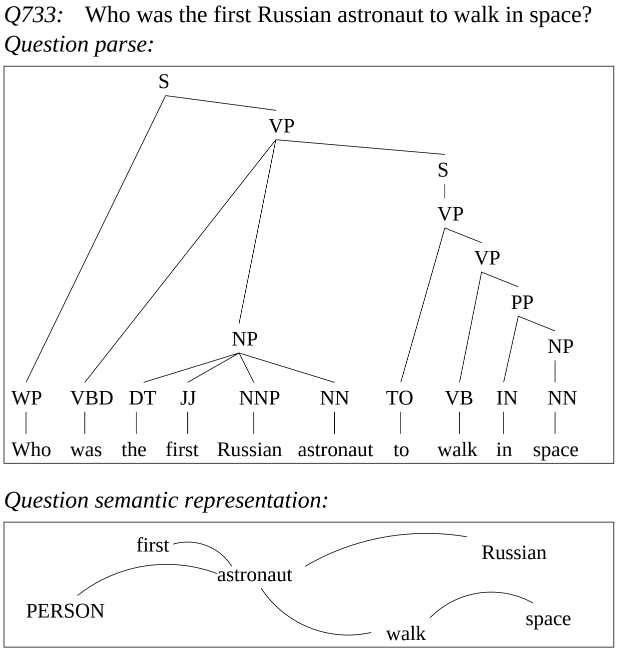


Figure 7: Building semantic forms

to *astronaut*, *walk* to *space* and *astronaut* and *PERSON*, the answer type. The label propagation rules are identical to the rules for mapping from trees to dependency structures used by Michael Collins (Collins 1996). These rules identify the head-child, and propagate its label up in the tree.

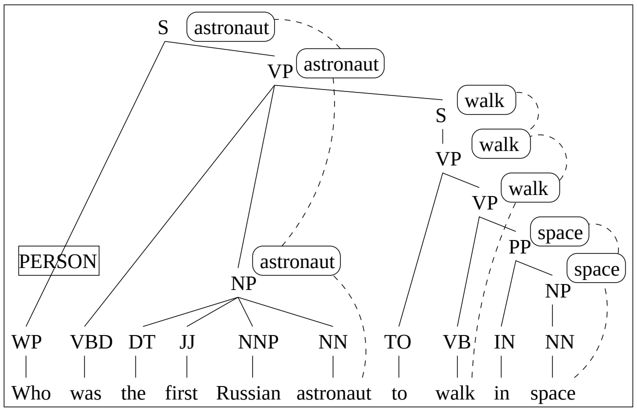


Figure 8: Parse tree traversal

Current probabilistic parsers have very good performance, are fast and robust, and can be easily operated on short texts, as represented by questions and short paragraphs were the answer may lie. The resulting trees do not impose great complexity for their traversal, therefore obtaining semantic forms of the format implemented in FALCON does not impose great processing constraints.

There are three immediate advantages of processing semantic forms:

1. Semantic forms facilitate the detection of the answer type. The node that has the largest connectivity in this representation is mapped in the answer hierarchy, producing the expected answer type;

2. Semantic forms indicate what keywords should be considered, and furthermore how their alternations should be searched. All nouns that are immediately related to the concept that determined the answer type are considered among the retrieval keywords. In addition, we consider their adjectival and adverbial adjuncts.

3. Semantic forms enable abductions based on relations between words rather than on bag-of-words approaches.

Keywords and Alternations

The paragraphs containing the answers are retrieved based on the keywords that are passed to the boolean search engine. To reduce the number of loops illustrated in Figure 2 and thus to enhance the performance of FALCON, we allow some alternations of the keywords to be passes as well. Such alternations can be classified according to the linguistic knowledge they are based upon:

1. *Morphological Alternations.* Based on the specificity of the question keyword that has determined the expected answer type we enable all the morphological derivations that are accessible from WordNet. For example, in the case of question *Q209: Who invented the paper clip?* we allow all the morphological alternations of the verb *invented*. For this question, the verb was mapped into its nominalization *inventor*, which is in the subhierarchies of the answer type *PERSON*. Therefore, we passed to the retrieval engine the query:

   QUERY(Q209):[*paper AND clip AND (invented OR inventor OR invent)*]

   To estimate the specificity of a concept in the answer hierarchy we followed some ideas set forward in (Pasca 2000). First a distinction between the *hypernym* relations encoded in WordNet is made. Some hypernym relations that model ISA relations and whereas other model INSTANCEOF relations. Their distinction is made by the presence of proper names in the conceptual synsets of the subsumed words.

Additionally, whenever a synset element contains the lexeme of one of its hypernyms, it is considered to be an instance of that hypernym. The specificity is measured by the number of nodes that are connected by IsA types of hypernym relations in the answer hierarchies.

2. *Lexical Alternations.* WordNet encodes a wealth of semantic information that is easily mined. Seven types of semantic relations span concepts, enabling the retrieval of synonyms and another semantically related terms. Such alternations improves the recall of the answer paragraphs. For example, in the case of question *Q221: Who killed Martin Luther King?*, by considering the synonym of *killer*, the noun *assassin*, the system retrieved paragraphs with the correct answer. Similarly, for the question *Q206: How far is the moon?*, since the adverb *far* is encoded in WordNet as being an attribute of *distance*, by adding this noun to the retrieval keywords, a correct answer is found.

3. *Semantic Alternations.* Mining from WordNet semantic knowledge that is not always localized in the conceptual synset allows for semantic alternations. An example was used in the case of question *Q258: Where do lobsters like to leave?*. Since in WordNet the genus of the definition of the verb *prefer* is *liking better*, the query becomes:

QUERY(Q58):[(lobster OR lobsters) AND (like OR prefer)]

In this way the likelihood of retrieving the correct answer is greatly enhanced.

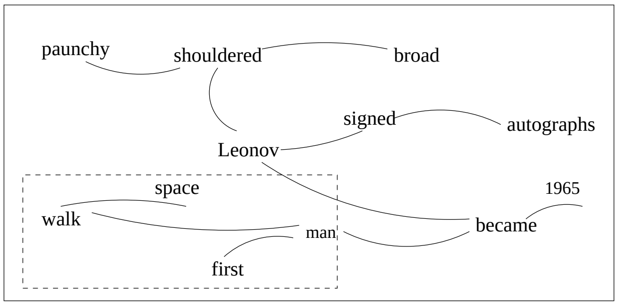
Abductive Knowledge

The semantic forms of questions and answers can be unified and thus enable a matching between the conceptual relations expressed in the question and the relations derived from the answer. Heuristics that are based on these relations are more reliable than those based on bag-of-words approaches. However, they do not always account for the correctness of the answer. For example, in Figure 9 we illustrate the answer to question Q733, described in Figure 7. The dashed rectangle indicates the result of the unification, but none of the concepts have the expected answer type, matched by the named entity Leonov. To better assess the correctness of the answer type we need to transforms both the question and the answer into logical forms.

The logical formulae in which questions or answers are translated are inspired by the notation proposed in

Answer: The broad-shouldered but paunchy Leonov, who in 1965 became the first man to walk in space, signed autographs.

Answer semantic representation:



Answer logic form:

paunchy(y) ^ shouldered(e1 y x) ^ broad(x) ^ Leonov(x) ^ first(z) ^
^ man(z) ^ space(t) ^ walk(e2 t z) ^ became(e3 z u x) ^
^ 1965(u) autographs(v) ^ signed(e4 v x) ^ HUMAN(x) ^ DATE(u)

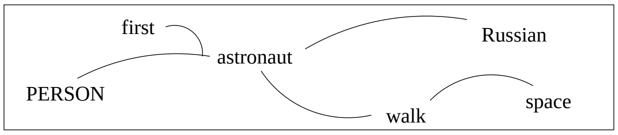
Figure 9: Semantic unifications and logical forms of an answer

(Hobbs 1986-1) and implemented in Tacitus (Hobbs 1986-2).

Based on the Davidsonian treatment of action sentences, in which events are treated as individuals, every question and every answer are transformed in a first-order predicate formula for which (1) verbs are mapped in predicates *verb(e,x,y,z,...)* with the convention that variable e represents the eventuality of that action or event to take place, whereas the other arguments (e.g. x, y, z, ...) represent the predicate arguments of the verb; (2) nouns are mapped into their lexicalized predicates; and, (3) modifiers have the same argument as the predicate they modify. For example, the question illustrated in Figure 7 is mapped in the logical form transformation (LFT) represented in Figure 10. We use the same procedure to build both question logical formulae (QLF) and answer logical formulae (ALF).

Q733: Who was the first Russian astronaut to walk in space?

Question semantic representation:



Question logic form:

first(x) ^ astronaut(x) ^ Russian(x) ^ space(z) ^ walk(y z x) ^
^ HUMAN(x)

Figure 10: Logic form transformations

The logical transformations are also used in the most elaborated form of filtering of our system: the justification option. A Q/A system that provides with the option of justifying the answer has the advantage that erroneous answers can be ruled out systematically. In our quest of enhancing the precision of a Q/A system by incorporating additional knowledge, we found this option very helpful. However, the generation of justifications for open-domain textual Q/A systems poses some challenges. First, we needed to develop a very efficient prover, operating on logical form transformations. Our proofs are backchaining from the questions through a mixture of axioms. We use three forms of axioms: (1) axioms derived from the facts stated in the textual answer; (2) axioms representing world knowledge; and (3) axioms determined by coreference resolution in the answer text. The LFT of the answers represent the first category of axioms. New axioms are added, modeling the coreference information and some general world knowledge, accessible from WordNet.

Feature	*Definition Question*
Q358	What is a meerkat?
Answer (short)	Score: 284.40 *The meerkat, a type of mongoose, thrives in ...*
Feature	*Keyword Alternations*
Q205	What is the population of Bahamas?
Answer (long)	Score: 142.56 *Mr Ingraham 's charges of 'impropriety' are unlikely to excite the 245,000 people of the Bahamas*
Feature	*Abductions*
Q258	Where do lobsters like to live?
Answer	Score: 145.58 *The water is cooler, and lobsters prefer that*

Table 6: Examples of FALCON answers.

Performance evaluation

Table 7 summarizes the scores provided by NIST for our system.

	NIST score *lenient*	NIST score *strict*
Short answer	59.9%	58.0%
Long answer	77.8%	76.0%

Table 7: Accuracy performance

Another important performance parameter is the contribution of each knowledge feature to the general performance of FALCON. From the total of 692 questions, keyword alternations were used for 89 questions and the justification option was needed to rule out erroneous answers for 121 questions. Table 6 illustrates examples of questions that could be answered by using different combinations of knowledge features.

Lessons learned

There are many forms of ambiguities that can be resolved when considering novel usages of large, open-domain linguistic resources such as WordNet. With FALCON we have been able to detect the expected answer type for 79% of the TREC-9 questions. To accomplish this, we have incorporated WordNet noun and verb hierarchies into our answer taxonomy. Furthermore, we have been able to mine WordNet for several forms of lexico-semantic knowledge that was used to generate keyword alternations. Additionally, we have built axioms for the abductive prover by combining the WordNet relational semantics with the data processed from the glosses of WordNet concepts. The proofs were quite short and did not generate any significant overhead on FALCON. Currently, most of the time is spent on retrieving the paragraphs.

In TREC-9, the test questions were not only more numerous than in TREC-8, but also with a higher degree of difficulty. Three dimensions define the increased difficulty. First, the TREC-9 test set comprised a larger variety of question classes and given the larger size, in each class there were more degrees of complexity. Table 8 shows the many-to-many correspondence between question classes and answer types. Third, given the origin of the test set, TREC-9 questions reflected real-world needs of current users, thus contained reformulations, definitions, comparatives and a significant number of questions dealing with superlative attributes of named entities or concepts.

References

Chris Buckley, Mandar Mitra, Janet Walz and Claire Cardie. SMART Hight Precision: TREC 7. In the *Proceedings of the Text Retrieval Conference TREC-7*, 1998.

Michael Collins. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34st Annual Meeting of the Association for Computational Linguistics, ACL-96*, pages 184–191, 1996.

Sanda Harabagiu and Dan Moldovan. Knowledge Processing on Extended WordNet. In *WordNet: An Electronic Lexical Database and Some of its Applications*, editor Fellbaum, C., MIT Press, Cambridge, MA, 1998.

Sanda Harabagiu and Steven Maiorano. Finding answers in large collections of texts: paragraph indexing + abductive inference. *Working Notes of the Fall*

Question Class	Examples	Answer type
Concept completion	*Q481: Who shot Billy the Kid?* *Q432: What state produces the best lobster to eat?* *Q449: What does Nicholas Cage do for a living?*	*PERSON* *STATE* *PROFESSION*
Definition	*Q228: What is platinum?* *Q584: What does NASA stand for?* *Q424: What do you call a group of geese?*	*DEFINITION* *ACRONYM* *GROUP*
Quantitative	*Q276: How much money does the Sultan of Brunei have?* *Q288: How fast can a Corvette go?* *Q210: How many dogs pull a sled in the Iditarod?*	*MONEY* *SPEED* *NUMBER*
Superlative	*Q636: Italy is the largest producer of what?* *Q294: Who is the richest person in the world?* *Q380: What language is mostly spoken in Brazil?*	*PRODUCT* *PERSON* *LANGUAGE*
Exemplification	*Q577: Can you give me the name of a clock maker in London, England?* *Q460: Name a Gaelic language.* *Q567: Name a female figure skater.*	*PERSON* *LANGUAGE* *PERSON*
Enablement/ Explanation	*Q616: What is the purpose of a car bra?* *Q315: Why can't ostriches fly?*	*PURPOSE* *REASON*

Table 8: Some TREC-9 Question classes and their corresponding answer types

AAAI Symposium on Question Answering, November 1999.

Sanda Harabagiu, Marius Paşca and Steven Maiorano. Experiments with Open-Domain Textual Question Answering. In the *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 292–298, 2000.

Jerry R. Hobbs. Discourse and Inference. Unpublished manuscript, 1986.

Jerry R. Hobbs. Overview of the TACITUS Project. In *Computational Linguistics*, 12:(3), 1986.

Jerry Hobbs, Mark Stickel, Doug Appelt, and Paul Martin. Interpretation as abduction. *Artificial Intelligence*, 63, pages 69–142, 1993.

G.A. Miller. WordNet: A Lexical Database. *Communication of the ACM*, vol 38: No11, pages 39–41, November 1995.

Dan Moldovan and Rada Mihalcea. A WordNet-based Interface to Internet Search Engines. In *Proceedings of the FLAIRS-98*, pages 275–279, 1998.

Dan Moldovan, Sanda Harabagiu, Marius Paşca, Rada Mihalcea, Richard Goodrum, Roxana Gîrju and Vasile Rus. LASSO - A tool for Surfing the Answer Net. *Proceedings of the Text Retrieval Conference (TREC-8)*, 1999.

Dan Moldovan, Sanda Harabagiu, Marius Paşca, Rada Mihalcea, Richard Goodrum, Roxana Gîrju and Vasile Rus. The Structure and Performance of an Open-Domain Question Answering System. *Proceedings of the 38th Annual Meeting of the Association for Comoutational Linguistics (ACL-2000)*, pages 563–570, 2000.

Marius A. Paşca. Open-Domain Factual Answer Extraction. *PhD Qualifying Thesis*, Department of Computer Science & Engr. Southern Methodist University, 2000.