Identifying implicit relationships

J. Chu-Carroll
E. W. Brown
A. Lally
J. W. Murdock

Answering natural-language questions may often involve identifying hidden associations and implicit relationships. In some cases, an explicit question is asked by the user to discover some hidden concept related to a set of entities. Answering the explicit question and identifying the implicit entity both require the system to discover the semantically related but hidden concepts in the question. In this paper, we describe a spreading-activation approach to concept expansion, backed by three distinct knowledge resources for measuring semantic relatedness. We discuss how our spreading-activation approach is applied to address these questions, exemplified in Jeopardy!™ by questions in the "COMMON BONDS" category and by many Final Jeopardy! questions. We demonstrate the effectiveness of the approach by measuring its impact on IBM Watson™ performance on these questions.

Introduction

Answering natural-language questions may often involve identifying hidden associations and implicit relationships. In the most straightforward cases, people may be interested in knowing how CNN and HBO are related to each other (both are cable TV networks owned by Time Warner) or what Teddy Roosevelt and Barack Obama have in common (both are Presidents of the United States, Nobel Peace Prize recipients, and alumni of Columbia University). Questions that seek common links between entities are plentiful in Jeopardy!** and are frequently referred to by the category "COMMON BONDS." For example, *feet*, *eyebrows*, and *McDonald's* have arches in common, whereas *trout*, *loose change in your pocket*, and *compliments* are all things that you fish for.

Another type of question requires resolving an implicit reference to a hidden concept. For example, "How old was the youngest U.S. president when he took office?" requires first identifying Teddy Roosevelt as the youngest U.S. president, which then leads to the answer "42". Jeopardy! includes many such questions, particularly in Final Jeopardy! when 30 seconds is given to answer the question. Some examples include "The 1648 Peace of Westphalia ended a war that began on May 23 of this year" (requires first identifying that the war is the Thirty Years' War) and "In the 19th century, he created a new type of reference work, a dictionary named for the Greek word for 'treasury'" (requires first identifying that the dictionary is a thesaurus). In some questions, this implicit concept may not be clearly indicated by an indefinite noun phrase, as in "a war" and "a dictionary" in the earlier examples. For instance, consider "On hearing of the discovery of George Mallory's body, this explorer told reporters he still thinks he was first". The implicit entity in this query is "Mount Everest", which is strongly related to both George Mallory and the sought explorer. Once this implicit entity is identified, the question becomes identifying the first person to successfully climb Mount Everest, who is Edmund Hillary. We refer to these questions as *missing link* questions.

The unifying theme for answering common-bond questions and missing-link questions is the need to identify concepts that are closely related to those given in the question. In IBM Watson*, we developed a recursive spreading-activation algorithm, which identifies related concepts based on a collection of heterogeneous underlying data resources. Watson's spreading-activation process leverages both linked data extracted from a Web corpus, as well as lexical and syntactic resources derived from large text corpora to compute the degree of relatedness between concepts. For common-bond questions, spreading activation is applied to each entity given in the question, and the most relevant and prominent concept related to all entities is selected as the answer. For missing-link questions, the spreading-activation process is used to score the degree of relatedness between an identified missing link and a candidate answer.

Digital Object Identifier: 10.1147/JRD.2012.2188154

The remainder of this paper is organized as follows: In the next section, we discuss Watson's spreading-activation process and the resources it uses to determine the relatedness between concepts. We then describe how this process is applied in solving common-bond and missing-link questions and present experimental results to demonstrate its effectiveness. Finally, we discuss related work and conclude.

Spreading activation for concept expansion

Spreading activation refers to the idea that concepts in a semantic network may be activated through their connections with already active concepts based on a certain spreading strategy [1, 2]. This process allows us to identify concepts closely related to a given concept and to score the relatedness between two concepts. Traditionally, concepts are represented in a semantic network where concept nodes are related to one another via certain types of relations, such as *isa* and *part-of* [2, 3]. These semantic networks allow systems to relate *dogs* to *mammals* and *wheels* to *cars*. However, rather than relying on manually created semantic networks to represent relatedness, Watson uses naturally occurring texts and measures concept relatedness on the basis of frequencies that concepts co-occur with one another under specific conditions in these texts. Performing spreading activation over natural-language texts allows us to utilize knowledge inherent in significantly larger sources of information than can possibly be manually encoded in a semantic network. To make use of different types of information in naturally occurring texts and their associated metadata, we implemented spreading activation in Watson using three different underlying resources to measure relatedness: an *n*-gram corpus, the PRISMATIC knowledge base [4], and Wikipedia** links. The spreading-activation interface allows for the specification of fan size f and depth d, which causes the process to identify the f-most-related concepts to the current active concept and to recursively invoke the activation process on these f new concepts another $d - 1$ times. The rest of this section describes the knowledge resources, how they are used to measure concept relatedness, and the characteristics of relationships captured by each resource.

Using an n-gram corpus

Lexical collocation in naturally occurring text is a simple and natural measure for concept relatedness. For example, "JFK" is semantically strongly related to "airport" and "assassination", and these relationships are represented in the *n*-gram corpus by the high collocation frequency between the terms "JFK" and "airport", as well as "JFK" and "assassination". Conversely, the fact that "JFK" and "modem" have little to do with each other is reflected by the fact that they seldom, if at all, appear close to each other in text.

A typical and effective way to measure the frequency of word collocation is to use a large corpus of naturally occurring text. A more efficient approach, however, is to represent the corpus as a collection of *n*-grams that occur greater than a minimum frequency, along with the frequency of each *n*-gram. Stemming and stop-word[1] removal can further reduce the size of the *n*-gram corpus. This compact representation of collocation counts, analogous to the Web 1T 5-gram corpus [5], allows us to encode collocation counts extracted from a large corpus in a form that can be efficiently accessed. To support spreading activation in Watson, we built a 5-gram corpus with frequency counts from Watson's primary unstructured sources [6], which include Wikipedia and the Gigaword corpus [7]. All 5 grams were stemmed, and stop words were removed. In addition, 5 grams that occurred fewer than five times were eliminated to reduce the corpus size.

Our *n*-gram-based spreading-activation implementation uses Lucene [8] and includes application programming interfaces to support the retrieval of frequently collocated terms given a term and the computation of semantic similarity given two terms. For the former, given term t, the most frequent 5 grams that include t are retrieved from the corpus. All terms in the retrieved 5 grams are sorted by their total frequencies in those 5 grams, and the top f (indicated by the fan size) most frequent terms are returned. For the latter, the normalized Google** distance (NGD) semantic similarity metric [9] was used to compute the semantic distance between two given terms based on the underlying *n*-gram corpus.

Using the PRISMATIC knowledge base

A resource frequently used in Watson is PRISMATIC, i.e., a knowledge base of extracted frames and slots based on syntactic and semantic relationships [4]. PRISMATIC aggregates statistics across fully or partially instantiated frames. It covers a variety of frame types, some of which are syntactic and some of which are semantic; the component described here uses only syntactic frames. One type of syntactic frame is the SVO (subject–verb–object) frame; for instance, the sentence "Ford pardoned Nixon in 1974" results in an SVO tuple, i.e., (*Ford*, *pardon*, *Nixon*). PRISMATIC provides quick access to statistics over these tuples. In particular, one can abstract out any element of a tuple and ask for the count of tuples matching that abstraction, e.g., the SVO query (*Ford*, ?v, ?o), where ?v and ?o are unbound variables, will provide a count of all SVO tuples for which *Ford* is the subject.

We use PRISMATIC similarly to the way we use *n*-grams, by estimating the degree of relatedness between two concepts with the frequency of how often they co-occur.

[1]Stop words are common words in a language, such as prepositions and pronouns, which are often removed in search-based applications to improve efficiency.

The core difference is that the *n*-gram component counts related words that appear lexically near each other, whereas the syntactic component counts related words that are syntactically connected. For example, the SVO tuple (*Ford*, *pardon*, *Nixon*) is extracted from the following sentence: "Ford did not act hastily but did finally pardon Nixon in September." This SVO tuple would provide evidence that "Ford" and "Nixon" are semantically related, whereas the 5 grams extracted from this sentence would not support the relationship, because "Ford" and "Nixon" do not appear within a 5-word window.

In addition to the SVO frame type previously mentioned, two other syntactic frame types are used, i.e., SVPO (subject-verb-preposition-object) and NPO (noun-preposition-object). For example, text such as "Ford met with Nixon" and "Ford after Nixon" would relate "Ford" to "Nixon" via the SVPO and NPO frame types, respectively. The SVPO query (*Ford*, ?*v*, ?*p*, *Nixon*) provides a count of how often Ford and Nixon are related through that syntactic construct in Watson's text corpus, and that count is combined with similar counts for SVO and NPO to compute a total frequency of links between these two terms. The combined frequency is used to compute a relatedness score between "Ford" and "Nixon."

Using Wikipedia links

Our third knowledge resource supporting the spreading-activation process contrasts with the first two in using metadata encoded in Web documents, rather than the texts of the documents themselves. We analyzed Wikipedia documents and the targets of links within each document and noted that the target document titles typically represent concepts closely related to the source document title. As an example, consider the following text segment, which is the first paragraph of the Wikipedia article on IBM. In the text below, ⟨x⟩ represents links where the anchor text and the target document title are both "x", and ⟨x|y⟩ represent links where x is the anchor text and y is the title of the target document.

> International Business Machines (IBM) (NYSE: IBM) is an ⟨American | United States⟩ multinational ⟨technology⟩ and ⟨consulting⟩ firm headquartered in ⟨Armonk, New York⟩. IBM manufactures and sells computer ⟨hardware | Personal computer hardware⟩ and ⟨software | Computer software⟩ and it offers ⟨infrastructure⟩, ⟨hosting | Internet hosting service⟩ and ⟨consulting services | Consultant⟩ in areas ranging from ⟨mainframe computers | Mainframe computer⟩ to ⟨nanotechnology⟩.

This example shows that anchor texts are oftentimes the same as the target document titles in Wikipedia. In cases where they differ, we attempt to capture semantic relatedness using the target document titles for two reasons. First, anchor texts frequently co-occur with the source document title in the body of the text, and our other two knowledge resources based on document texts can likely capture that relationship. Second, the target document title represents the canonical form for all anchor texts pointing to that document. Using the canonical form representation gives us a higher likelihood that we will find a common related concept given two or more concepts. Using target document titles, we will find from the aforementioned example that "IBM" is related to concepts such as "computer software", "consultant", and "Internet hosting service", which are not present in the original document text.

To support the spreading-activation process, we extracted, from each Wikipedia source document, all target document titles for links in the source document. Since terms are typically only linked the first time it appears in a document, we do not model degrees of relatedness with this resource. Instead, given term *t*, we identify the Wikipedia document whose title best matches *t* and return all target document titles from links in that document.

Application to common-bond questions

Common-bond questions generally refer to questions that seek the hidden relationship among multiple entities. In Jeopardy!, they are frequently, although not uniformly, indicated by the category COMMON BONDS and have question texts that consist of a list of typically three elements:

(1) COMMON BONDS: Bobby, bowling, rolling. (Answer: "pins")
(2) COMMON BONDS: Your legs, your T's, the Rubicon. (Answer: "things you cross")
(3) CULINARY COMMON BONDS: Grinder, hero, submarine. (Answer: "sandwiches")
(4) COMMON BONDS: Shirts, TV remote controls, telephones (Answer: "things with buttons")

The aforementioned examples provide a sample of the different ways in which an answer may be related to concepts in the question. In (1), the answer, "pins", is a common head noun that may follow all three modifiers, whereas in (2), the answer, "cross", is a common verb that may precede all three entities. For (3), the answer, "sandwiches", is a supertype of all entities in the question, whereas in (4), the answer, "buttons", is a common attribute of all three given entities. Although it is possible to develop different algorithms for addressing different subtypes of common-bond questions, we focus on the commonality among these examples, namely, that the answers are all semantically closely related to the given entities. This observation of semantic relatedness enables us to adopt the spreading-activation mechanism previously outlined as a principal method for answering common-bond questions.

Spreading activation is used to answer common-bond questions in two ways: to identify concepts that are closely related to each given entity and to score each concept on the basis of their degrees of relatedness to all given entities. To comply with the DeepQA architecture [10], entity identification is implemented as a candidate generator (which produces candidate answers), and entity scoring is implemented as an answer scorer (which takes candidate answers and produces a numerical score for each answer).

Common-bond candidate generation

To maximize candidate recall, Watson's common-bond candidate generator identifies concepts that are closely related to each entity individually and considers the union of all such concepts as possible candidates. Our analysis of common-bond questions in Jeopardy! indicates that the answer is typically directly related to the given entities; therefore, we set the depth parameter d in our spreading-activation process to 1 and empirically determined the value of the fan size f to be 50, to balance the recall and the number of candidates produced.

The spreading-activation process is invoked on each entity given in the question. The related concepts found through this process are normalized on the basis of morphological variations, and each concept in the merged set is generated as a candidate answer. Since, for most questions, the common bond can be found in lexical proximity to the given entities, we used only the n-gram corpus to identify the most frequently collocated terms with each given entity and proposed them as candidates. We discuss the scoring of these candidates in the next section.

One characteristic of our n-gram-based common-bond candidate generation is that the candidate answers are limited to the singleton tokens that appear in the n-gram. For example, the correct answer for the clue "COMMON BONDS: Icebergs, torpedoes, loose lips" is "things that sink ships". Even allowing for just "sink ships" as the correct answer, we cannot generate that phrase as a candidate answer since the n-grams are constructed at the token level. Since Watson employs multiple candidate generation strategies, some of our general-purpose candidate generation techniques, which are attempted on all questions and are described in detail in [11], may be able to propose the correct candidate answer. For more complete coverage, however, we need to go beyond n-gram-based candidate generation and explore the use of other knowledge resources for concept expansion. We leave this as future work.

Common-bond answer scorer

Each candidate proposed by the common-bond candidate generator is scored on the basis of its semantic relatedness to each given entity using our spreading-activation process. Specifically, we compute an NGD similarity score using our n-gram corpus. Candidates that do not co-occur with a given entity in our corpus are given a low default background frequency for smoothing in the computation of its NGD score. The scores representing the candidate's semantic relatedness to the given entities are multiplied together to represent the overall goodness of the candidate as a common-bond answer. This score is posted as a feature for the candidate answer to be considered in the final merging and ranking process [12], in which features provided by the candidate generators and answer scorers are employed by a statistical model to rank the candidate answers and produce a confidence score for each one.

An alternative approach that we could have taken in the candidate generation and scoring process is to limit the proposed candidates to the intersection of the related concept sets, instead of taking the union of the sets. We opted for the latter approach because we have empirically found that for a substantial number of questions, the correct answer appears as frequently collocated concepts for two of the three entities. In those cases, while the concept is semantically related to the remaining entity, the link is not as prominent. Taking the union of the related concept sets allows us to take those candidates into consideration.

Application to missing-link questions

Missing-link questions are questions in which a missing entity is either explicitly or implicitly referred to and the identification of this missing entity facilitates answering the question. These questions are plentiful in the Jeopardy! domain, particularly in Final Jeopardy! where the questions are typically less direct and of which 20% may benefit from missing-link processing. Consider two examples:

(5) THE 17th CENTURY: The 1648 Peace of Westphalia ended a war that began on May 23 of this year. (Answer: "1618")
(6) EXPLORERS: On hearing of the discovery of George Mallory's body, this explorer told reporters he still thinks he was first. (Answer: "Edmund Hillary")

In the first example, the missing link is explicitly referred to by the indefinite noun phrase "a war", and the link can be resolved by answering the question "The 1648 Peace of Westphalia ended this war". The correct answer to that question instantiates the missing link as "Thirty Years' War", which can be then substituted into the original question to get "Thirty Years' War began on May 23 of this year".

In the second example, there is no explicitly mentioned missing link in the original question. However, George Mallory is best known for his attempts to climb Mount Everest, and this implicit concept helps us understand that the original question is seeking the person most widely credited as being the first to reach the top of Mount Everest.

For both types of missing-link questions, an important observation is that the missing link is strongly related to key concepts in the question. In the first example, "Thirty Years' War" is closely related to "Peace of Westphalia" and "May 23". In the second example, "Mount Everest" has a strong relationship with "George Mallory" and "discovery of Mallory's body". Our approach to answering these missing-link questions is to first hypothesize the missing links by scoring all candidate answers using a select subset of all features that are indicative of a candidate's semantic relatedness to key concepts in the question. We then invoke the system again by including these missing links in the search process, with the hope that the new search results will include some correct answers that we previously failed to generate as candidate answers. Finally, to favor candidates that are related to concepts in the question through the identified missing link, we developed new answer scorers based on the spreading-activation techniques described earlier to measure concept relatedness. The rest of this section describes the missing-link identification, candidate generation, and scoring processes. An alternative approach to addressing missing-link questions is based on syntactically decomposing the question and solving the subparts separately. That approach complements the spreading-activation approach discussed in this paper and is described in [13].

Missing-link identification

For an entity to be a good missing link, there are two necessary conditions: It must be highly related to concepts in the question, and it must be ruled out as a possible correct answer. We used existing Watson components to evaluate these conditions.

Several of Watson's components produce features that predominantly measure relatedness of a candidate answer with the question. These include the primary search rank and score [11], some of the textual evidence scores (not including the very precise logical form scorer) [14], and spatial and temporal association features [15]. We trained a machine-learning model with this subset of features as one phase in our multiphase answer merging and ranking framework [12]. We applied an empirically selected threshold on the score produced by this model to select a small number of entities that are most highly associated with the question.

Many of these "high-association" answers are actually the correct answer to the question. Those that are not correct, however, are frequently the missing links that we are looking for. The next step in finding missing links is to determine whether any high-association answers can be definitively ruled out as possible correct answers. The most common way that a candidate can be ruled out as the correct answer is that it is of the wrong answer type. In aforementioned example (5), "Thirty Years' War" appears as a high-association answer but is not of the right answer type "year" and is thus a prime candidate as a missing link. To identify these high-association answers of the wrong answer type, we use information from Watson's type coercion components [16] to select missing links that have a poor type match. Specifically, a candidate answer will be considered a missing link only if its combined type coercion score is below the mean for all candidate answers for that question. Note that one shortcoming of this identification process is that it is unable to handle questions that have useless lexical answer types,[2] such as "it" or "this", or those where the missing link and the correct answer are of the same type, as in the example below where the missing link "the Titanic" is a ship:

(7) FAMOUS SHIPS: In 1999, the wreck of this ship, known for its historic 1912 rescue effort, was discovered 120 miles off England. (Answer: "RMS Carpathia")

We leave an extension of our current approach that addresses this type of questions as future work.

Candidate generation using missing links

If the aforementioned identification process hypothesizes that one or more missing links exist for the question, we invoke part of the question-answering process again, taking the missing link(s) into consideration. This second iteration begins at the search and candidate generation phase and continues through the merging and ranking phase to produce a final ranked list of answers with their associated confidences. For additional information on the different phases in the DeepQA architecture, see [10].

In the second iteration, new search queries are produced by augmenting each existing query with a missing link. This allows the system to focus the search process on key concepts in the question with an additional bias toward the inferred missing link. Queries with and without the missing link are given to the search components, and candidates are extracted from the search results, as described in [11].

Missing-link answer scorer

Watson employs a number of textual evidence answer scorers that measure the goodness of match between the question and a passage supporting a candidate answer [14]. Although these answer scorers are effective for evaluating candidates whose passage context matches the question well, they are not as effective if the correct answer is related to concepts in the question through an implicit concept that is absent from the question. To properly evaluate these candidate answers obtained through missing-link processing, we can take one

[2]Lexical answer types are terms in the question that indicate what type of entity is being asked for. For more detail, see [17].

of the two approaches. First, we can formulate a new question that includes the identified missing link and use our existing answer scorers to align the passages with the augmented question. Second, we can develop new answer scorers specifically for the purpose of scoring semantic relatedness between a candidate answer and concepts in the question through the identified missing link. We opted for the second approach because some of our textual evidence scorers are very sensitive to the precise phrasings of the question and it is difficult to automatically generate grammatically and semantically correct augmented questions that are likely to align well with justifying passages. For instance, in example (6), it is fairly straightforward to identify "Mount Everest" as a missing link, given its strong association with George Mallory. However, to formulate a proper question that includes the missing link, Watson would have to automatically generate "On hearing of the discovery of George Mallory's body, this explorer told reporters he still thinks he was first to reach the top of Mount Everest". The generation of this augmented question is by no means trivial and, if not done properly, might diminish the chance of the successful scoring of candidate answers. On the basis of this observation, we believe that semantic-relatedness scorers through identified missing links are more likely to successfully capture the intended relationships expressed between the question and the correct answer.

The goal of the semantic-relatedness scorers is to determine, for each candidate answer, the degree of relatedness between the candidate answer and some concept in the question through an identified missing link. Because the semantic relatedness between a given missing link and the concepts in the question is the same for all candidate answers, the scoring task can be simplified as computing the semantic relatedness between the missing link and each candidate answer.

For each candidate-answer and missing-link pair, we compute a semantic-relatedness score between them using our spreading-activation process previously described. Our analysis shows that the missing links in most Jeopardy! questions are directly related to a question concept and the correct answer. As a result, we set the depth parameter d in the spreading-activation process to 1 for semantic-relatedness scoring.

Three instantiations of the missing-link association scorer are implemented and integrated into Watson, each of which using a different underlying knowledge resource to measure semantic relatedness. Each scorer posts a feature on the candidate answer with a value that indicates the degree of relatedness between that candidate answer and the missing link. These features are taken into consideration in the final merging and ranking phase [12] to determine Watson's confidence in each candidate being the correct answer to the question. Note that in our subsequent ablation experiments, the scorer using our n-gram corpus did not have additional contributions beyond the other two scorers. As a result, the final configuration of Watson included only two missing-link scorers, i.e., one using Wikipedia links and the other using the PRISMATIC knowledge base.

Experimental evaluation

Common-bond evaluation

Experimental setup

To evaluate the impact of our common-bond candidate generation and scoring processes, we evaluated Watson's end-to-end performance on a set of 139 previously unseen common-bond questions. These questions are selected by extracting all questions that have the phrase "COMMON BONDS" in the category. By manual examination, all 139 questions are indeed common-bond questions. This relatively small test set reflects the general frequency of common-bond questions in Jeopardy!. Overall, common-bond questions are very infrequent, representing less than 0.2% of all Jeopardy! questions.

We compare end-to-end system performance for two versions of the system. The baseline system includes everything in Watson except the n-gram-based common-bond candidate-answer generator and answer scorer, which are previously described. The enhanced system adds the candidate-answer generator and answer scorer to the baseline. These components produce common-bond *rank* and *score* features that are included in the candidate-answer feature weighting models. The training set includes 102 common-bond questions among the 14,770 training questions. For both versions of the system, we compare candidate binary recall, defined as the percentage of questions for which the correct answer is found as a candidate answer; accuracy; and Precision@70, which is the system's precision when answering the top 70% of the questions of which it is most confident.

Results and discussion

The baseline system, without special common-bond processing components, achieves a binary recall of 69%, an overall accuracy of 48%, and Precision@70 of 62%. Adding the common-bond candidate-answer generator and answer scorer brings binary recall up to 73% (+4%), accuracy to 58% (+10%), and Precision@70 to 73% (+11%). These results are summarized in **Table 1**.

The contribution of the common-bond candidate-answer generator and answer scorer is primarily in the accuracy and the better confidence estimation for the candidate answers, whereas candidate binary recall is only modestly improved. Let us consider binary recall first. On our test set of 139 questions, the common-bond candidate generator produced at least one candidate for 113 (81%) questions. For 80 (81%) of the 113 questions, these common-bond

Table 1 Common-bond evaluation results.

	Binary recall	*Accuracy*	*Precision@70*
Baseline	69%	48%	62%
+Common bond	73%	58%	73%
Percentage change	4%	10%	11%

candidates contained the correct answer. When combined with Watson's other candidate generation strategies, the common-bond candidate-answer generator improves binary recall for just six questions, bringing the total number of questions in the test set for which the correct candidate answer is generated from 96 to 102. This suggests that the current common-bond candidate-answer generation technique has substantial overlap with existing methods and still leaves room for improvement. The six questions where the common-bonds candidate-answer generator helps are generally of the noun-phrase form, where either the head noun or the modifier are the common link.

The questions for which the common-bond candidate-answer generator fails to generate the correct candidate answer are generally of the form where the common bond is a more abstract concept and is not likely to co-occur with the given clue phrases in our *n*-gram resource. For example, the following questions are problematic:

COMMON BONDS: Cotton candy, a spider web, a top. (Answer: "things that are spun")
COMMON BONDS: Modem, Quasar, Gestapo. (Answer: "acronyms")
COMMON BONDS: A place setting, a baseball diamond, the earth's crust. (Answer: "plates")

In all of these examples, the common bond is a concept that is strongly related to each of the clue phrases but will not necessarily co-occur with the clue phrases such that they would appear in an *n*-gram. A more sophisticated technique that explores spreading activation that better models semantic relatedness at the conceptual level is required.

With respect to accuracy and Precision@70, adding the common-bond candidate-answer generator clearly improves both accuracy and precision, indicating that finding the common bond in proximity to the clue phrases, via *n*-gram spreading activation, is a useful technique. In particular, the *n*-gram technique is often able to identify the more generic term relevant to all of the clue phrases, whereas the baseline technique might not be able to overcome a high scoring candidate answer that is strongly related to just one of the clue phrases. For example, for the clue "COMMON BONDS: Spice, interrupted, Georgy", the baseline system prefers "Girl, Interrupted", whereas the system enhanced with the common-bond candidate-answer generator is able to prefer the correct answer, "girls", which is associated with all three clue phrases.

Missing-link evaluation

Experimental setup

To evaluate the impact of our missing-link processing mechanism, we tested Watson's end-to-end performance on a set of 1,112 previously unseen Final Jeopardy! questions. The end-to-end performance is compared with a baseline where the configuration of Watson for answering regular Jeopardy! questions is used to answer these Final Jeopardy! questions, i.e., without missing-link processing.

We compare the performance of the two versions of the system along two dimensions. First, we evaluate the effectiveness of missing-link identification and missing-link candidate generation with candidate binary recall. The idea behind identifying missing links and including them in a second-round search process is to be able to produce candidate answers that are not sufficiently relevant to the question concepts alone to be found in the first place. Therefore, we expect candidate binary recall to increase with missing-link processing. Second, we evaluate the system's ability to use the missing-link answer scorers to promote these correct answers obtained through missing-link candidate generation to first place. This is measured by the system's end-to-end question-answering accuracy.

Results and discussion

Our evaluation results, shown in **Table 2**, demonstrate the performance of our missing-link processing mechanism using the metrics previously discussed. The left-hand side of the table shows performance metrics measured over all questions, and the right-hand side shows the same metrics measured over only those questions where a missing link was identified.

Of the 1,112 Final Jeopardy! questions in our test set, Watson identified a missing link for 259 of them. Just under 20% of these 259 questions were not missing-link questions, resulting in a missing-link question detection precision of greater than 80%. Roughly 60% of the identified missing-link questions had an explicit indefinite noun phrase in the question, whereas the other 40% had implicit missing links. A comparison of the performance figures for the missing-link subset and the full test set indicates that questions in the missing-link subset are more difficult than the other questions. It is more difficult to successfully produce the correct answer as a candidate, as evidenced by the 0.7% drop in binary recall. Furthermore, it is much more difficult to score the correct answers well, as shown by the 5.5% reduction in accuracy. In the rest of our analysis, we focus on results of the missing-link subset since that is the subset of questions where the system had any potential of making an improvement.

Table 2 Missing-link evaluation results.

	All questions (1,112)		*Missing-link subset (259)*	
	Binary recall	*Accuracy*	*Binary recall*	*Accuracy*
Baseline	74.82%	51.08%	74.1%	45.6%
+Missing link	75.63%	51.53%	76.5%	47.1%
Percentage change	0.81%	0.45%	2.4%	1.5%

Table 3 Select examples of successful missing-link questions.

Category and clue	*Previous top answer*	*Missing link*	*New top answer*
WORLD AUTHORS: Chapters in an 1831 work by this author include "Maitre Jacques Coppenole" & "A Tear For A Drop of Water."	The Hunchback of Notre Dame	The Hunchback of Notre Dame	Victor Hugo
CIVIL WAR-ERA FICTION: A northerner whose sympathies exiled him to the Confederacy, Bermuda & Canada inspired this 1863 tale.	Vallandigham	Vallandigham	The Man without a Country
1998 BESTSELLERS: 35 years after her death, she's the subject of a new collection of poems by her husband.	Mary Shelley	Hughes	Sylvia Plath
FILM CLASSICS: This 1951 classic stars the AFI's top picks for the greatest male & female film legends.	The Day the Earth Stood Still	Katharine Hepburn	The African Queen
2008: Though not elected to the position, a man from this state became the 1st blind governor & the 4th black governor in the U.S.	Arkansas	David Patterson	New York

The binary recall improvement measures the combined impact of missing-link identification and missing-link candidate generation, showing that Watson was able to extract the correct answer in its candidate list for an additional 2.4% of the questions in the missing-link subset. The accuracy metric shows an overall 1.5% absolute end-to-end question-answering improvement on these questions. To put these performance figures in context, we compare Watson's Final Jeopardy! performance with human performance. Statistics extracted from *J! Archive*, a website maintained by Jeopardy! fans that contains records of thousands of past Jeopardy! games, indicate that the performance on Final Jeopardy! questions by Jeopardy! contestants is roughly 48% in accuracy, further underscoring the difficulty of these questions. Our error analysis of Final Jeopardy! questions shows a wide variety of reasons for their failures, each of which requiring enhancements to the current Watson system to address. The missing-link processing mechanism discussed in this paper addressed a small portion of these questions and achieved a small but positive impact on the overall system performance.

Table 3 shows some examples where Watson used missing-link processing to identify the correct answers that it previously failed to identify. In the first two examples, the missing link, which is strongly related to the answer and is of the wrong answer type, was initially given as the top answer. Our missing-link processing mechanism identified the top answer as a missing link, causing the correct answer to be generated as a candidate and scored as the new top answer. The last three examples show cases where the initial top answer is incorrect but is of the correct answer type. Watson identified a candidate answer lower in the answer list that satisfies the missing-link criteria, i.e., high semantic relatedness and wrong type. Again, when the missing link is taken into consideration, Watson successfully produces the correct answer as a candidate and promotes it to the top position in the answer list.

Related work

The theory of spreading activation originated in cognitive psychology and was used to explain semantic processing, lexical and speech retrieval, etc. [1, 18, 19]. The theory has since been applied to information retrieval [2, 3, 20] and natural-language semantics [21–23]. In these efforts, the knowledge resources underlying the spreading-activation processes have used semantic networks such as WordNet** [23] or derived resources such as one from LDOCE (Longman Dictionary of Contemporary English) [22]. In contrast, instead of using existing structured semantic networks, Watson utilizes unstructured and semistructured knowledge resources derived from large text corpora to measure semantic relatedness in its spreading-activation process.

Although previous work on question answering has focused for the most part on factoid questions, there have been attempts at addressing some more complex questions analogous to the missing-link questions addressed in this paper. Most question-answering systems that address complex questions do so by performing syntactic and/or semantic decomposition of the original question so that new but simpler questions may be formulated and answered. The answers to these new questions can be then composed to form the answers to the original question [24, 25]. Instead of developing strategies for general factoid decomposition, some previous approaches specifically focus on the identification of certain expressions within a question and center decomposition around those expressions, such as temporal expressions [26, 27] and meronymy [26]. A significant difference between these systems and Watson's missing-link processing mechanism is that because of their reliance on syntactic and semantic decomposition, those systems can handle only the class of questions that we classified as *explicit* missing links, i.e., when the missing entity is explicitly referred to in the question, although not named. They cannot infer implicit entities such as "Mount Everest" in the George Mallory/Edmund Hillary example discussed earlier in this paper.

Conclusion

In this paper, we have described a spreading-activation approach for concept expansion and for measuring semantic relatedness. We have developed three knowledge resources for supporting the spreading-activation process: 1) the *n*-gram corpus, which captures semantic relatedness based on lexical collocation, 2) the PRISMATIC knowledge base, which measures relatedness of concepts based on syntactic collocation, and 3) Wikipedia links, which uses metadata mined from Wikipedia link structures to indicate semantic relatedness.) The spreading-activation process has been applied to identify missing semantic relations between concepts. More concretely, we have shown how this technique can be adopted in an end-to-end question-answering system to more effectively address two types of Jeopardy! questions, i.e., common-bond questions, which seek a common element among multiple given entities, and Final Jeopardy! questions, for which identifying a missing element alluded to in the question can facilitate the process of finding the correct answer. Our experimental results on blind data show that the techniques that we have developed for identifying missing associations improved common-bond questions by 10% in accuracy and 11% in Precision@70, and improved the subset of Final Jeopardy! questions for which a missing link was identified by 2.4% in candidate recall and 1.5% in accuracy.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Jeopardy Productions, Inc., Wikimedia Foundation, Google, Inc., or Trustees of Princeton University in the United States, other countries, or both.

References

1. A. M. Collins and E. F. Loftus, "A spreading-activation theory of semantic processing," *Psychol. Rev.*, vol. 82, no. 6, pp. 407–428, Nov. 1975.
2. G. Salton and C. Buckley, "On the use of spreading activation methods in automatic information retrieval," in *Proc. 11th ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1988, pp. 147–160.
3. P. Cohen and R. Kjeldsen, "Information retrieval by constrained spreading activation on semantic networks," *Inf. Process. Manage.*, vol. 23, no. 4, pp. 255–268, Jul. 1987.
4. J. Fan, A. Kalyanpur, D. C. Gondek, and D. A. Ferrucci, "Automatic knowledge extraction from documents," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 5, pp. 5:1–5:10, May/Jul. 2012.
5. T. Brants and A. Franz, *Web 1T 5-gram Version 1*, 2006, LDC 2006T13. [Online]. Available: http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html#!/2006/08/all-our-n-gram-are-belong-to-you.html
6. J. Chu-Carroll, J. Fan, N. Schlaefer, and W. Zadrozny, "Textual resource acquisition and engineering," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 4, pp. 4:1–4:11, May/Jul. 2012.
7. D. Graff, J. Kong, K. Chen, and K. Maeda, *English Gigaword Third Edition*, 2007, LDC 2007T07. [Online]. Available: http://www.mendeley.com/research/english-gigaword-third-edition/
8. Apache Lucene. [Online]. Available: http://lucene.apache.org
9. R. Cilibrasi and P. Vitanyi, "Automatic meaning discovery using Google," in *In Kolmogorov Complexity and Applications*, 2006. [Online]. Available: http://homepages.cwi.nl/~paulv/papers/amdug.pdf
10. D. A. Ferrucci, "Introduction to 'This is Watson,'" *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 1, pp. 1:1–1:15, May/Jul. 2012.
11. J. Chu-Carroll, J. Fan, B. K. Boguraev, D. Carmel, D. Sheinwald, and C. Welty, "Finding needles in the haystack: Search and candidate generation," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 6, pp. 6:1–6:12, May/Jul. 2012.
12. D. C. Gondek, A. Lally, A. Kalyanpur, J. W. Murdock, P. Duboue, L. Zhang, Y. Pan, Z. M. Qiu, and C. Welty, "A framework for merging and ranking of answers in DeepQA," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 14, pp. 14:1–14:12, May/Jul. 2012.
13. A. Kalyanpur, S. Patwardhan, B. K. Boguraev, A. Lally, and J. Chu-Carroll, "Fact-based question decomposition in DeepQA," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 13, pp. 13:1–13:11, May/Jul. 2012.
14. J. W. Murdock, J. Fan, A. Lally, H. Shima, and B. K. Boguraev, "Textual evidence gathering and analysis," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 8, pp. 8:1–8:14, May/Jul. 2012.
15. A. Kalyanpur, B. K. Boguraev, S. Patwardhan, J. W. Murdock, A. Lally, C. Welty, J. M. Prager, B. Coppola, A. Fokoue-Nkoutche, L. Zhang, Y. Pan, and Z. M. Qiu, "Structured data and inference in DeepQA," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 10, pp. 10:1–10:14, May/Jul. 2012.
16. J. W. Murdock, A. Kalyanpur, C. Welty, J. Fan, D. A. Ferrucci, D. C. Gondek, L. Zhang, and H. Kanayama, "Typing candidate answers using type coercion," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 7, pp. 7:1–7:13, May/Jul. 2012.
17. A. Lally, J. M. Prager, M. C. McCord, B. K. Boguraev, S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll, "Question analysis: How Watson reads a clue," *IBM J. Res. & Dev.*, vol. 56, no. 3/4, Paper 2, pp. 2:1–2:14, May/Jul. 2012.

18. J. Neely, "Semantic priming and retrieval from lexical memory: Roles of inhibitionless spreading activation and limited-capacity attention," *J. Exp. Psychol., Gen.*, vol. 106, no. 3, pp. 226–254, 1977.
19. G. Dell, "A spreading-activation theory of retrieval in sentence production," *Psychol. Rev.*, vol. 93, no. 3, pp. 283–321, Jul. 1986.
20. F. Crestani, "Application of spreading activation techniques in information retrieval," *Artif. Intell. Rev.*, vol. 11, no. 6, pp. 453–482, Dec. 1997.
21. G. Hirst, "Resolving lexical ambiguity computationally with spreading activation and polaroid words," in *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics*, G. Cottrell and M. Tanenhaus, Eds. San Mateo, CA: Morgan Kaufmann, 1988.
22. H. Kozima and T. Furugori, "Similarity between words computed by spreading activation on an English dictionary," in *Proc. 6th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 1993, pp. 232–239.
23. G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos, "Word sense disambiguation with spreading activation networks generated from Thesauri," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 1725–1730.
24. B. Katz, G. Borchardt, and S. Felshin, "Syntactic and semantic decomposition strategies for question answering from multiple resources," in *Proc. AAAI Workshop Inference Textual Question Answering*, 2005, pp. 35–41. [Online]. Available: http://groups.csail.mit.edu/infolab/publications/Katz-etal-AAAI05W5.pdf
25. A. Hickl, P. Wang, J. Lehmann, and S. Harabagiu, "FERRET: Interactive question-answering for real-world environments," in *Proc. COLING/ACL Interactive Present. Sessions*, 2006, pp. 25–28. [Online]. Available: http://acl.ldc.upenn.edu/P/P06/P06-4007.pdf
26. S. Hartrumpf, "Semantic decomposition for question answering," in *Proc. 18th Eur. Conf. Artif. Intell.*, 2008, pp. 313–317.
27. E. Saquete, J. Vicedo, P. Martinez-Barco, R. Munoz, and H. Llorens, "Enhancing QA systems with complex temporal question processing capabilities," *J. Artif. Intell. Res.*, vol. 35, no. 1, pp. 755–811, May 2009.

Received July 18, 2011; accepted for publication December 12, 2011

Jennifer Chu-Carroll *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (jencc@us.ibm.com).* Dr. Chu-Carroll is a Research Staff Member in the Semantic Analysis and Integration Department at the IBM T. J. Watson Research Center. She received a Ph.D. degree in computer science from the University of Delaware in 1996. Prior to joining IBM in 2001, she spent 5 years as a Member of Technical Staff at Lucent Technologies Bell Laboratories. Dr. Chu-Carroll's research interests are in the area of natural-language processing, more specifically in question-answering and dialogue systems. Dr. Chu-Carroll serves on numerous technical committees, including as program committee co-chair of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2006 and as general chair of NAACL HLT 2012.

Eric W. Brown *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (ewb@us.ibm.com).* Dr. Brown is a Research Staff Member in the Semantic Analysis and Integration Department at the IBM T. J. Watson Research Center. He received a B.S. degree in computer science from the University of Vermont in 1989 and M.S. and Ph.D. degrees in computer science from the University of Massachusetts, Amherst, in 1992 and 1996, respectively. He subsequently joined IBM, where he has worked on information retrieval, text analysis, and question answering. Since 2007, he has been a technical lead on the Watson project. He is a member of the Association for Computing Machinery and Sigma Xi.

Adam Lally *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (alally@us.ibm.com).* Mr. Lally is a Senior Technical Staff Member at the IBM T. J. Watson Research Center. He received a B.S. degree in computer science from Rensselaer Polytechnic Institute in 1998 and an M.S. degree in computer science from Columbia University in 2006. As a member of the IBM DeepQA Algorithms Team, he helped develop the Watson system architecture that gave the machine its speed. He also worked on the natural-language processing algorithms that enable Watson to understand questions and categories and gather and assess evidence in natural language. Before working on Watson, he was the lead software engineer for the Unstructured Information Management Architecture project, an open-source platform for creating, integrating, and deploying unstructured information management solutions.

J. William Murdock *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (murdockj@us.ibm.com).* Dr. Murdock is a member of the IBM DeepQA Research Team in the T. J. Watson Research Center. In 2001, he received a Ph.D. degree in computer science from Georgia Tech, where he was a member of Ashok Goel's Design and Intelligence Laboratory. He worked as a post-doctorate with David Aha at the U.S. Naval Research Laboratory. His research interests include natural-language semantics, analogical reasoning, knowledge-based planning, machine learning, and self-aware artificial intelligence.