Watson Discovery Advisor: Question-answering in an industrial setting

Charles Beller and **Graham Katz** and **Allen Ginsberg** and **Chris Phipps**
cebeller@us.ibm.com, egkatz@us.ibm.com, abginsbe@us.ibm.com, phippsc@us.ibm.com

Sean Bethard and **Paul Chase** and **Elinna Shek** and **Kristen Summers**
slbethar@us.ibm.com, pjchase@us.ibm.com, eshek@us.ibm.com, kmsummer@us.ibm.com

IBM Watson Group
Chantilly, VA, United States

Abstract

This work discusses a mix of challenges arising from Watson Discovery Advisor (WDA), an industrial strength descendant of the Watson Jeopardy! Question Answering system currently used in production in industry settings. Typical challenges include generation of appropriate training questions, adaptation to new industry domains, and iterative improvement of the system through manual error analyses.

1 Introduction

Question answering has been a focus for research in computational linguistics and natural language processing since early efforts in the 1960s (Simmons, 1970). Systems such as BASEBALL (Green Jr. et al., 1961) focused on specific, circumscribed domains, e.g. questions about baseball games. As computing power and knowledge resources developed, answering questions over a broad open domain became a focus of engineers and researchers in government, industry, academic settings (Voorhees, 1999; Wang, 2006). Various approaches to question answering have exploited logical forms and logic provers (Moldovan et al., 2003a), relationship matching over structured knowledge bases (Yao and Van Durme, 2014), and the wisdom of the masses via social media analysis (Bian et al., 2008) or crowd-sourcing (Boyd-Graber et al., 2012).

Open-Domain question answering received a wave of renewed attention when IBM's Watson system successfully competed with human champions in the Jeopardy! challenge. That system was developed into the Watson Discovery Advisor IBM product, which we describe below.

2 Watson Discovery Advisor

The Watson Discovery Advisor product (WDA) is a direct descendant of the Watson Jeopardy! system described in (Ferrucci et al., 2010; Ferrucci, 2012). The core processing pipelines are built on the Apache UIMA architecture (Ferrucci and Lally, 2004) and consist of a question-analysis phase (Lally et al., 2012; Kalyanpur et al., 2012), a document and passage retrieval phase, an answer generation phase (Chu-Carroll et al., 2012), an answer scoring phase, and an answer merging and ranking phase (Gondek et al., 2012).

There are several important differences between the Jeopardy! task and the question answering task performed by WDA. At the most superficial level, the Jeopardy! task provides an input consisting of a declarative clue, and expects an interrogative question, i.e. it's an Answer-Questioning task. In reality though, this tends to be a simple syntactic transformation. A more important difference is that Jeopardy! clues are always accompanied by *category* information. In some cases, this category level information is crucial, giving explicit parameters that restrict the range of answers to be considered.

Another difference has to do with the nature of the input questions. Jeopardy! includes a range of puzzle-type questions that required special logic (Prager et al., 2012), these kinds of trick questions do not typically have any place in an industrial question-answering setting. Even without these trick

Proceedings of 2016 NAACL Human-Computer Question Answering Workshop, pages 1–7,
San Diego, California, June 12-17, 2016.

questions, though, Jeopardy! questions do not look like the typical questions that users pose to WDA.

Consider an important element identified as part of question analysis, the *focus* of a question or clue. The focus is the element that, if replaced with a correct answer, will result in a true sentence. In Jeopardy! clues, the focus often contains considerably more information than a formulation of an equivalent question by a user. For example, where a Jeopardy! clue might read *This home of the Eiffel Tower is the capital of France*, a user might simply enter *What is the capital of France?*. Where the simple question can be answered only if a single fact is known, the Jeopardy! style clue provides two independent ways to answer, i.e. knowing the capital of France, or knowing the home of the Eiffel Tower. Such relative impoverishment of user questions as compared to Jeopardy! clues is typical.

3 System Tuning

When WDA is deployed to a new customer, its capabilities are tuned for that customer's particular use case and setting. The first step in tuning is to identify domain appropriate data that will be ingested into the knowledge base. In the case of a financial information discovery application, for example, financial news documents and shareholder reports might be identified. Once appropriate data has been identified and acquired, the main task is to train WDA's answer ranking models. Like the original Jeopardy! system, WDA uses a supervised ranking model that requires a large set of question-answer pairs.

The original Jeopardy! system was trained on past Jeopardy! questions and answers collected from the J! Archive.[1] In industrial applications the QA sets are instead tailored to reflect customer domain interests. Development of these question sets is discussed in Section 3.1. In tandem with developing QA sets, WDA is typically also fed terminology, jargon, and differences in word usage from the domains of interest. This can include significant detailed domain-specific knowledge; e.g., names for corporate entities and financial instruments.

[1] *www.j-archive.com*

3.1 QA Training Data Development

For Watson Discovery Advisor, the bulk of training data is composed of sets of question-answer pairs (QA sets). These QA sets make up the labeled data used to train the system's ranking models. Each QA pair consists of a factoid question with a canonical answer and a set of variant forms for the answer. Variants might include nicknames, acronyms, or alternative spellings. The size of domain-specific QA sets ranges from several hundreds to many thousands. As with many supervised learning scenarios, more is typically better.

As WDA has matured through client engagements, QA sets have been developed for a wide variety of domains, providing a variety of hard-won lessons. The primary goal of Question Answer development is to exercise the feature space sufficiently in model training that WDA's answer ranking models recognize what makes a good factoid answer for a particular domain. The feature space is complex and includes feature scorers that are sensitive to a range of phenomena, from syntactic features of questions, relationships between questions and evidence passages, and orthographic details of answer candidates to connections between terms in evidence passages and internal ontologies or other knowledge resources. To best develop effective models, QA sets must consist of factoid questions with a variety of syntactic structures, based on specific domains and topics, and possessing unambiguous answers.

Ensuring this last condition is not a simple task. Most questions can have a range of possible answers, depending on what kind of information a person is looking for, and even identifying all variant names for a single entity is a challenge. For example, the question *Who won the World Cup?* seems like it should have a simple answer. However, there are several very different and equally plausible answers, depending on what the asker is looking for. Is the question asking about the FIFA world cup or an event from some other sport like cricket, rugby, or chess? Which year's world cup is being asked about? Is the question about the men's or the women's event? Is the question asking for the player who scored the winning goal, or the team?

In our experience, a good factoid question contains disambiguating information within it and has

a single, identifiable answer. There are some common mistakes people make when generating factoid questions. We list these below:

- ambiguous questions
- questions missing key references or context
- questions with confusing grammar or wording
- questions with incorrect or incomplete answers
- opinion questions
- non-English questions for an English system
- overly detailed questions
- question irrelevant to the desired domain
- trick questions
- complex question types

These break down into two broad types of (human) errors. At the top of the list are features that make a poor training question because they provide insufficient or insufficiently extractable information for a system to identify a single answer. Towards the bottom of the list are features that make a poor training question because they diverge too greatly from the typical inputs a user would provide. These classes are not mutually exclusive, features in the middle of the list are often combinations of the two.

Specifying answers is also difficult. Even unambiguous factoid questions can have multiple forms of a correct answer. For example, the simple factoid question *Which writer created the TV sitcom '30 Rock'?* has one unambiguous answer, but there are at least five common variants observed in documents: *Tina Fey*, *Elizabeth Stamatina Fey*, *Elizabeth Stamatina "Tina" Fey*, *Ms. Fey*, and *Fey*. Since it is not at all clear which of these should be taken to be the one objectively correct answer, and in general a variety of answers are accepted as correct by users and judges alike, variants must be listed to prevent WDA from learning to avoid generating answers that might well be taken to be correct.

Developing good domain-specific QA sets is one of the major challenges to adapting the WDA system to a new domain. Once QA sets are developed, the system can be trained and evaluated. At this point further progress can be driven by error analysis. The goal is to identify what questions the system still fails on, and more importantly: Why?

3.2 Error Analysis

To tune the system and assess gaps in Watson's knowledge base, the system is evaluated on a held-out test set of QA pairs, which, after processing by the system, is subjected to detailed Error Analysis (Moldovan et al., 2003b; Tellex et al., 2003). Incorrectly answered questions are automatically assigned to several broad error classes, then manually categorized into more detailed sub-classes of error. The automatic deterministic classification of errors is done to bin the errors into the following classes:

- *No Search Hits:* Questions for which the generated search queries fail to retrieve any passage from the corpus that contains the correct answer to the question
- *Unextracted Answers:* Questions for which the retrieved passages contain correct answers, but none of these answers are generated as hypotheses
- *Imperfect Answers:* Questions for which one of the answers generated as a hypothesis partially matches the correct answers
- *Correct Answer 2-10:* Questions for which the correct answer is hypothesized, but ranked below the top answer
- *Correct Answer 11-100:* Questions for which the correct answer is hypothesized, but ranked well below the top answer

These initial bins are the starting point for the programmatic error analysis, as they tend to reflect the likely system component to which the error can be attributed. Broadly speaking the potential loci for error are the following system components:

- the *ingested corpora* might not contain a document that answers the question,
- the *search component* might fail to identify a responsive document passage,
- the *answer generators* might fail to hypothesize the correct or complete answer from a responsive passage,
- the *answer scorers* and *rankers* might fail to rank the correct answer highly.

Once the major error-contributing component is identified, attempts are made to articulate more

clearly the exact source of error. For example, answer generation-related errors might arise in a number of different situations, such as incorrect tokenization or problems with anaphora resolution and coreference.

Further Considerations Error analysis for a question-answering system which is deployed to customer application is somewhat more nuanced in its evaluation than standard academic metrics such as accuracy would reveal. While returning a correct answer is important, and returning more correct answers is ideal, there are other considerations. Among these are the topical domains of specialization: In our setting, it is clearly more important to answer correct questions that are of vital importance to a customers core informational needs then it is to correctly answer trivia questions. Of additional importance is that questions which are answered incorrectly are at least answered reasonably. For example, to the question *How many people have been on the moon?* an answer of *Ten* instead of the correct *Twelve* is understandable by a client. It is the kind of error that anyone might make. An answer of *Three pounds* is not. Improving both targeted accuracy (accuracy on a question set relevant to customer concerns) and reasonableness are important goals for industrial applications.

4 Watson on Quiz Bowl

In recent work on Question Answering, data from the Quiz Bowl competitions has played a central role (Boyd-Graber et al., 2015; Iyyer et al., 2014). While the goal of the Watson Discovery Advisor is to address the information discovery concerns of real-world customers, the authors thought there might be some interest in seeing how well WDA played Quiz Bowl. Unfortunately, issues with the answer key make a quantitative evaluation impossible. Despite this, a qualitative evaluation showed some interesting results. In the this section we discuss the issues raised by this attempt.

4.1 WDA plays (limited) Quiz Bowl

A number of History-domain test items from the Iyyer, et al., set were put to an untrained and untuned WDA instance. WDA comes "out of the box" trained on domain general questions not unlike those of Quiz Bowl. We configured the knowledge base to consist primarily of Wikipedia data. While WDA provided answers to the complete data set, automatic evaluation was impossible due to issues with the answer key. On inspection it became clear that many of the answers generated by WDA which did not match the answer key were acceptable variants of the correct answer.[2] For example, the question *For 10 points (FTP), name this antitrust act passed in 1914, which augmented the Sherman antitrust act* was listed as having the answer *Clayton Antitrust Act*. WDA's top ranked answer was *Clayton Act*, which is clearly correct.

To provide an appropriate quantitative assessment of WDA's capabilities, we would need to manually identify all cases of incorrectly answered questions for which returned answer was an acceptable variant of the correct answer. Unfortunately this information is unavailable in the open data set. We do note some interesting differences: WDA returns surnames rather than full names for many historical figures (*Truman* for *Harry S Truman*). WDA prefers to avoid repeating nominals from the question in the answer, returning *Waitangi* for *Treaty of Waitangi*, in response to the query: *. . . name this* **treaty** *that formally put New Zealand under British rule and sparked the Maori Wars*. In addition there are numerous derivational, inflectional or orthographic variants that judges would likely deem acceptable, such as *St. Petersburg* for *Saint Petersburg* and *Vandal* for *Vandals*, and many others. Further, WDA finds cases of true alternates such as *Bonus Expeditionary Force* for *Bonus Army*. Finally, there are cases where the answer key is simply incorrect: For the clue *Ruling from 1556 to 1605, he also conquered Afghanistan and Baluchistan. FTP name this 16th century emperor, the greatest of the Mughals*, the answer key provides the incorrect *Shah Jahan*, while WDA finds the correct answer, *Emperor Akbar I*.

These issues make an overall quantitative evaluation of WDA on the Quiz Bowl data a challenge. There are, however, a number of cases in which WDA gets a clearly wrong answer. In the next section we discuss an example that illustrates one of the

[2] Approaches that treat answers as labels in a question classification task (e.g. Iyyer, et al. 2014) avoid this issue. We return to QA as classification in section 4.3

difficulties WDA had with Quiz Bowl data.

4.2 Analyzing a Quiz Bowl Example

For the question below the answer is the *War of 1812*.

> During this war, Andrew Jackson defeated the Creek at the Battle of Horseshoe Bend. Tecumseh died during the Battle of the Thames in this war. The White House was burned by the British Army during this war. Francis Scott Key composed the "Star Spangled Banner" during this conflict. Impressment of U.S. sailors was a major cause of this war. FTP, what 19th-century war between the U.S. and Britain was named for the year it began.

In our experiment with WDA *War of 1812* occurs at rank 7. The answers at rank 1 through 6 include the candidate answers *Baltimore*, *McHenry*, *Siege of Fort Erie*, etc. All of the candidates come from documents that are relevant to the question and most of the passages selected for candidate generation are highly responsive. What is clear, however, is that the (incorrect) candidates are not the "right type of thing" to be an answer to this question, i.e., they are not wars. One reason these non-war candidates are ranked above the correct answer, in this case, has to do with the identification of the question's *Lexical Answer Type* (LAT), the word or phrase in the question that indicates the type of thing that an answer to the question would be an instance of. WDA's ranking models promote answers which are appropriate to the LAT above those that are not. Since WDA fails to find the LAT for this question (*war*), the correct answer is not appropriately ranked.

One general complication with running WDA on Quiz Bowl questions is that they involve multiple sentences. This complicates the LAT identification task. The version of Watson designed for the Jeopardy! challenge assumed that the question clues would be a single sentence or a structured input of a single sentence clue and a short category label. Both the Watson Jeopardy! system and WDA are optimized for that use case. To deal with the Quiz Bowl questions, the question analysis mechanisms, namely the LAT identification models, would need to be retrained to handle multi-sentence input. We are confident that if this were done, questions like the above would be highly likely to be answered correctly by WDA.

To support this contention, we rephrased the initial multi-sentence clue into the single sentence question: *In which 19th-century war between the U.S. and Britain did Andrew Jackson defeat the Creek at the Battle of Horseshoe Bend?* Under this formulation WDA identifies the correct LAT (*war*) and gets the correct answer in 1st place with 52% confidence. The 2nd (and incorrect) answer, *Civil War* (confidence 23%) is notably also of the correct answer type. Answers 3-10 are not wars, but the system's confidence associated with those answers drops sharply from 7% to 1%.

4.3 Quiz Bowl as Classification

One of the factors making practical question answering so challenging is the sparseness of the data. The vast majority of answers required of a working system are unique and — insofar as the training data consists of the QA sets used to tune the ranking (and other) models — unseen in training. For example, in a set of 5045 simulated-user questions used by IBM Watson to train the WDA system, more than 65% of the answers are unique (they were answers to just one question in the data set). The distribution of answer frequencies follows the familiar Zipfian distribution, with many low frequency items and very few high frequency items (Zipf, 1949) — the ratio of questions to (unique) answers is about 1.8:1. While it appears that Quiz Bowl questions in general have a similar ratio of questions to answers, the Quiz Bowl dataset used as the basis for the experiments by Iyyer et al. (2014) does not. In that data set questions with low-frequency answers are removed. Overall for the the 20407 questions there are only 2370 different answers—a question to answer ratio of nearly 9 to 1. This quantitative difference underscores the qualitative differences between the open-domain question answering task that WDA typically addresses and the more coarse-grained question-labeling tasks described in (Iyyer et al., 2014), which in some ways is more of a classification task.

To explore text classification as a method of "question answering" we applied the IBM Bluemix deep-learning based Natural Language Classifier (*www.bluemix.net*) (Ma et al., 2015; Kurata et al.,

	Iyyer et al.	**IBM Bluemix**
History	82.3%	**84.6%**
Literature	78.7%	**90.7%**

Table 1: Classifier accuracy on history and literature questions.

2016) to the data set.[3] We trained two different short-text classifiers with the NLC, one for History-themed questions and one for Literature-themed questions. Each classifier was trained in a supervised manner to label the full-text of the clue (as input) with the answer string (as label). The History training set had 2708 items and 629 labels, the Literature training set had 4064 items and 812 labels. We applied the History-trained classifier to a held-out 455-item test set of History-themed questions and the Literature-trained classifier to a held-out 596-item test set of Literature-themed questions.[4] Table 1 reports the accuracy achieved by these classifiers along with the best results reported in (Iyyer et al., 2014) on the same data sets.

Such classification methods are, of course, limited in the ways that they can address the true question answering task — the task of answering a question with an answer not seen in the training data. As it happens, there were 26 items in the History test data set whose answer did not appear in the training set. The NLC classifier, as expected, failed to label these items correctly, and we suspect that none of the other classifier-based methods did either. We ran this small set of data through WDA, identifying acceptable answer variants by hand. WDA answered 20 of these 26 questions correctly, achieving an accuracy of 76% on this (admittedly small) data set.

5 Discussion

Our test run of WDA on a portion of the Quiz Bowl question set raises some issues with regard to the differences of doing question-answering for real world clients and domains versus domains like Jeopardy! and Quiz Bowl. As discussed in Section 2, one of the key differences is the amount of information, essentially clues or hints, that a question provides. Experience has shown that the average question length in real-world engagements is very short compared to that of Quiz Bowl and Jeopardy! type questions. This has a direct effect on search-query generation for passage retrieval and all subsequent processing. Upon inspection, a small sample of queries generated from Quiz Bowl questions had an average of 69.2 simple query terms.[5] A comparable sample of queries generated from WDA training questions had an average of 8.6 simple query terms, about 1/8th as many.[6] For this reason, search results returned in a real-world system will tend to cover a broader set of topics than in the more constrained search engendered by the Quiz Bowl setting. The richness of the Quiz Bowl clues makes finding relevant documents a much easier task than it is in the world of industrial question answering. One potentially valuable avenue for research involves robustly expanding sparse queries to generate better search results.

6 Conclusion

Using an open domain question answering system in a production setting offers a number of challenges distinct from those encountered in research and game-playing settings. We have discussed techniques and strategies for adapting and improving a Watson Discovery Advisor question answering system to improve performance in any particular production environment. Additionally we provided an outline of our Error Analysis process on cases in which WDA was applied to Quiz Bowl data and demonstrated the effectiveness of the IBM Natural Language Classifier on this data. Furthermore, we were able to show how effectively WDA performed on cases in which classifier-based methods failed.

Acknowledgments

The authors would like to thank Will Dubyak and Dick Darden for their invaluable technical guidance and mentorship, and we thank Brian Keith for his leadership and vision. We also thank Bowen Zhou for the robust Natural Language Classifier service that we used in the Quiz Bowl experiments.

[3] For this method, the issues with the answer key raised above are, with the exception of the actual errors, irrelevant, since the answer key is normalized.

[4] For line-length reasons a small number of test and training items had to be excluded from this experiment.

[5] In this context, simple query terms encompass both atomic terms and alternations (logical OR).

[6] N=5 for both samples. Max: 83 min: 52 for Quiz Bowl, max: 12, min: 7 for WDA

References

Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. 2008. Finding the right facts in the crowd: factoid question answering over social media. In *Proceedings of the 17th international conference on World Wide Web*, pages 467–476. ACM.

Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. 2012. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1290–1301. Association for Computational Linguistics.

Jordan Boyd-Graber, Mohit Iyyer, He He, and Hal Daumé III. 2015. Interactive incremental question answering. In *Neural Information Processing Systems*.

Jennifer Chu-Carroll, James Fan, BK Boguraev, David Carmel, Dafna Sheinwald, and Chris Welty. 2012. Finding needles in the haystack: Search and candidate generation. *IBM Journal of Research and Development*, 56(3.4):6–1.

David Ferrucci and Adam Lally. 2004. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

David A Ferrucci. 2012. Introduction to this is watson. *IBM Journal of Research and Development*, 56(3.4):1–15.

DC Gondek, Adam Lally, Aditya Kalyanpur, J William Murdock, Pablo Ariel Duboué, Lei Zhang, Yue Pan, ZM Qiu, and Chris Welty. 2012. A framework for merging and ranking of answers in deepqa. *IBM Journal of Research and Development*, 56(3.4):14–1.

Bert F Green Jr., Alice K Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, pages 219–224. ACM.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Empirical Methods in Natural Language Processing*.

Aditya Kalyanpur, Siddharth Patwardhan, BK Boguraev, Adam Lally, and Jennifer Chu-Carroll. 2012. Fact-based question decomposition in DeepQA. *IBM Journal of Research and Development*, 56(3.4):13–1.

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In *Proceedings of NAACL/HLT 2016*.

Adam Lally, John M Prager, Michael C McCord, BK Boguraev, Siddharth Patwardhan, James Fan, Paul Fodor, and Jennifer Chu-Carroll. 2012. Question analysis: How watson reads a clue. *IBM Journal of Research and Development*, 56(3.4):2–1.

Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. 2015. Dependency-based convolutional neural networks for sentence embedding. *Volume 2: Short Papers*, page 174.

Dan Moldovan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. 2003a. Cogex: A logic prover for question answering. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 87–93. Association for Computational Linguistics.

Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. 2003b. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21(2):133–154.

John M Prager, Eric W Brown, and Jennifer Chu-Carroll. 2012. Special questions and techniques. *IBM Journal of Research and Development*, 56(3.4):11–1.

Robert F Simmons. 1970. Natural language question-answering systems: 1969. *Communications of the ACM*, 13(1):15–30.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM.

Ellen M Voorhees. 1999. The trec-8 question answering track report. In *Proceedings of the 8th Text REtrieval Conference (TREC-8)*, pages 77–82.

Mengqiu Wang. 2006. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics*, 1(1).

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *ACL (1)*, pages 956–966. Citeseer.

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort*. Addison-Wesley Press.