Proceedings of the Fourth Workshop
on Teaching NLP, Sofia, Bulgaria,
August 2013

Semantic Technologies in IBM Watson$^{\text{TM}}$

Alfio Gliozzo
IBM Watson Research Center
Yorktown Heights, NY 10598
gliozzo@us.ibm.com

Or Biran
Columbia University
New York, NY 10027
orb@cs.columbia.edu

Siddharth Patwardhan
IBM Watson Research Center
Yorktown Heights, NY 10598
siddharth@us.ibm.com

Kathleen McKeown
Columbia University
New York, NY 10027
kathy@cs.columbia.edu

Abstract

This paper describes a seminar course designed by IBM and Columbia University on the topic of Semantic Technologies, in particular as used in IBM Watson$^{\text{TM}}$ — a large scale Question Answering system which famously won at Jeopardy!® against two human grand champions. It was first offered at Columbia University during the 2013 spring semester, and will be offered at other institutions starting in the fall semester. We describe the course's first successful run and its unique features: a class centered around a specific industrial technology; a large-scale class project which student teams can choose to participate in and which serves as the basis for an open source project that will continue to grow each time the course is offered; publishable papers, demos and start-up ideas; evidence that the course can be self-evaluating, which makes it potentially appropriate for an online setting; and a unique model where a large company trains instructors and contribute to create educational material at no charge to qualifying institutions.

1 Introduction

In 2007, IBM Research took on the grand challenge of building a computer system that can perform well enough on open-domain question answering to compete with champions at the game of Jeopardy! In 2011, the open-domain question answering system dubbed Watson beat the two highest ranked players in a two-game Jeopardy! match. To be successful at Jeopardy!, players must retain enormous amounts of information, must have strong language skills, must be able to understand precisely what is being asked, and must accurately determine the likelihood they know the right answer. Over a four year period, the team at IBM developed the Watson system that competed on Jeopardy! and the underlying DeepQA question answering technology (Ferrucci et al., 2010). Watson played many games of Jeopardy! against celebrated Jeopardy! champions and, in games televised in February 2011, won against the greatest players of all time, Ken Jennings and Brad Rutter.

DeepQA has applications well beyond Jeopardy!, however. DeepQA is a software architecture for analyzing natural language content in both questions and knowledge sources. DeepQA discovers and evaluates potential answers and gathers and scores evidence for those answers in both unstructured sources, such as natural language documents, and structured sources such as relational databases and knowledge bases. Figure 1 presents a high-level view of the DeepQA architecture. DeepQA utilizes a massively parallel, component-based pipeline architecture (Ferrucci, 2012) which uses an extensible set of structured and unstructured content sources as well as a broad range of pluggable search and scoring components that allow integration of many different analytic techniques. Machine Learning techniques are used to learn the weights for each scoring component in order to combine them into a single final score. Watson components include a large variety of state of the art solutions originating in the fields of Natural Language Processing (NLP), Machine Learning (ML), Information Retrieval (IR), Semantic Web and Cloud Computing. IBM is now aggressively investing in turning IBM Watson from a research prototype to an industry level highly adaptable system to be applied in dozens of business ap-

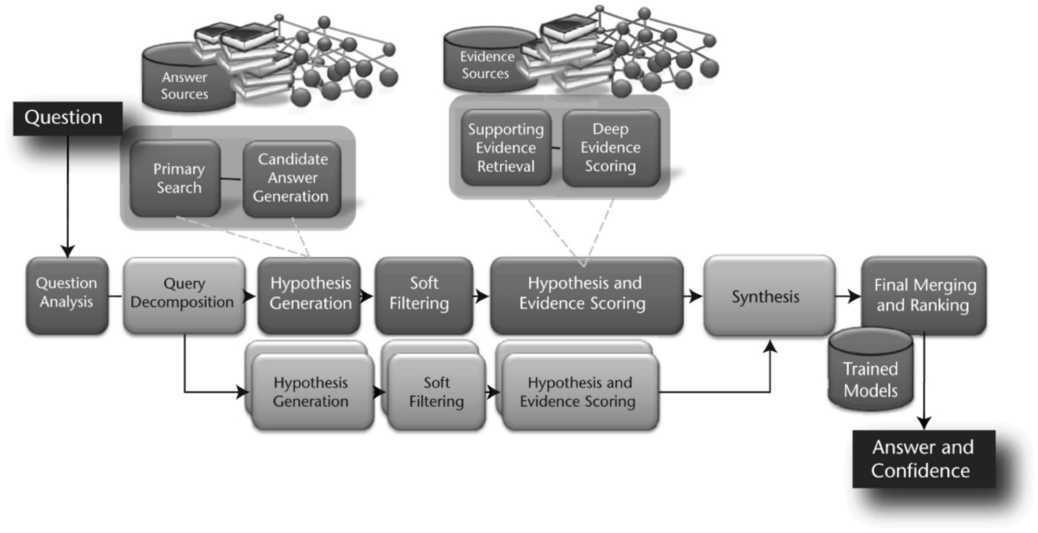


Figure 1: Overview of the DeepQA architecture

plications ranging from healthcare to finance (Ferrucci et al., 2012).

Finding that particular combination of skills in the entry-level job market is hard: in many cases students have some notion of Machine Learning but are not strong in Natural Language Processing; in other cases they have background in Knowledge Management and some of the basics of Semantic Web, but lack an understanding of statistical models and Machine Learning. In most cases semantic integration is not a topic of interest, and so understanding sophisticated platforms like Apache UIMA^{TM} (Ferrucci and Lally, 2004) is a challenge. Learning how to develop the large scale infrastructure and technology needed for IBM Watson prepares students for the real-world challenges of large-scale natural language projects that are common in industry settings and which students have little experience with before graduation.

Of course, IBM is interested in hiring entry-level students as a powerful way of scaling Watson. Therefore, it has resolved to start an educational program focused on these topics. Initially, tutorials were given at scientific conferences (NAACL, ISWC and WWW, among others), universities and summer schools. The great number of attendees (usually in the range of 50 to 150) and strongly positive feedback received from the students was a motivation to transform the didactic material collected so far into a full graduate-level course, which has been offered for the first time at Columbia University. The course (which is described in the rest of this paper) received very positive evaluations from the students and will be used as a template to be replicated by other partner universities in the following year. Our ultimate goal is to develop high quality didactic material for an educational curriculum that can be used by interested universities and professors all over the world.

2 Syllabus and Didactic Material

The syllabus[1] is divided equally between classes specifically on the Watson system, its architecture and technologies used within it, and classes on more general topics that are relevant to these technologies. In particular, background classes on Natural Language Processing; Distributional Semantics; the Semantic Web; Domain Adaptation and the UIMA framework are essential for understanding the Watson system and producing successful projects.

The course at Columbia included four lectures by distinguished guest speakers from IBM, which were advertised to the general Columbia community as open talks. Instead of exams, the course included two workshop-style presentation days: one at the mid term and another at the end of the

[1]The syllabus is accessible on line `http://www.columbia.edu/~ag3366`

course. During these workshops, all student teams gave presentations on their various projects. At the mid-term workshop, teams presented their project idea and timeline, as well as related work and the state-of-the-art of the field. At the final workshop, they presented their completed projects, final results and demos. This workshop was also made open to the Columbia community and in particular to faculty and affiliates interested in start-ups. The workshops will be discussed in further detail in the following sections. The syllabus is briefly detailed here.

- **Introduction: The Jeopardy! Challenge** The motivation behind Watson, the task and its challenges (Prager et al., 2012; Tesauro et al., 2012; Lewis, 2012).

- **The DeepQA Architecture** Chu-Carroll et al. (2012b), Ferrucci (2012), Chu-Carroll et al. (2012a), Lally et al. (2012).

- **Natural Language Processing Background** Pre-processing, tokenization, POS tagging, named entity recognition, syntactic parsing, semantic role labeling, word sense disambiguation, evaluation best practices and metrics.

- **Natural Language Processing in Watson** Murdock et al. (2012a), McCord et al. (2012).

- **Structured Knowledge in Watson** Murdock et al. (2012b), Kalyanpur et al. (2012), Fan et al. (2012).

- **Semantic Web** OWL, RDF, Semantic Web resources.

- **Domain Adaptation** Ferrucci et al. (2012).

- **UIMA** The UIMA framework, Annotators, Types, Descriptors, tools. Hands-on exercise with the class project architecture (Epstein et al., 2012).

- **Midterm Workshop** Presentations of each team's project idea and their research into related work and the state of the art.

- **Distributional Semantics** Miller et al. (2012), Gliozzo and Isabella (2005).

- **Machine Learning and Strategy in Watson**

- **What Watson Tells Us About Cognitive Computing**

- **Final Workshop** Presentations of each team's final project implementation, evaluation, demo and future plans.

3 Watson-like Architecture for Projects

The goal of the class projects was for the students to learn to design and develop language technology components in an environment very similar to IBM's Watson architecture. We provided the students with a plug-in framework for *semantic search*, into which they could integrate their project code. Student projects will be described in the following section. This section details the framework that was made available to the students in order to develop their projects.

Like the Watson system, the project framework for this class was built on top of Apache UIMA (Ferrucci and Lally, 2004)[2] — an open-source software architecture for building applications that handle unstructured information.

The Watson system makes extensive use of UIMA to enable interoperability and scale-out of a large question answering system. The architecture (viz., DeepQA) of Watson (Ferrucci, 2012) defines several high-level "stages" of analysis in the processing pipeline, such as *Question and Topic Analysis*, *Primary Search*, *Candidate Answer Generation*, etc. Segmentation of the system into high-level stages enabled a group of 25 researchers at IBM to independently work on different aspects of the system with little overhead for interoperability and system integration. Each stage of the pipeline clearly defined the inputs and outputs expected of components developed for that particular stage. The researchers needed only to adhere to these input/output requirements for their individual components to easily integrate them into the system. Furthermore, the high-level stages in Watson, enabled massive scale-out of the system through the use of the *asynchronous scaleout* capability of UIMA-AS.

Using the Watson architecture for inspiration, we developed a *semantic search* framework for the class projects. As shown in Figure 2, the framework consists of a UIMA pipeline that has several high-level stages (similar to those of the Watson system):

[2] http://uima.apache.org

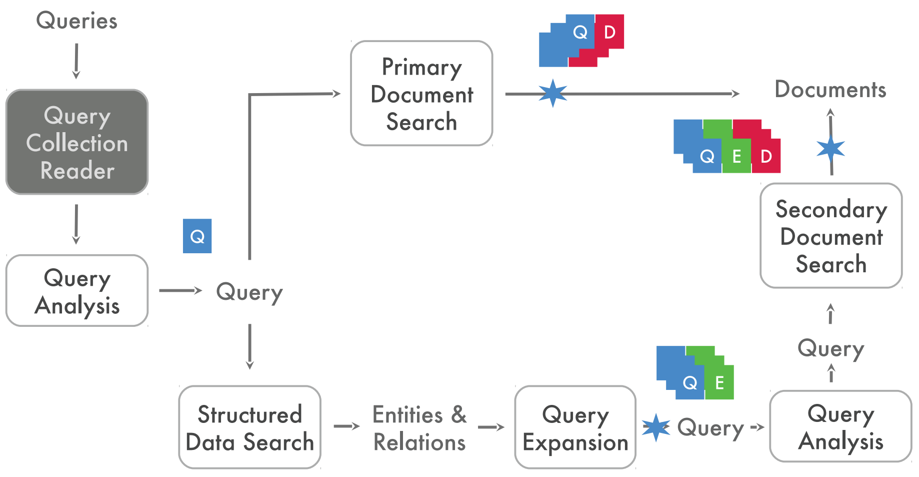


Figure 2: Overview of the class project framework

1. Query Analysis
2. Primary Document Search
3. Structured Data Search
4. Query Expansion
5. Expanded Query Analysis
6. Secondary Document Search

The input to this system is provided by a *Query Collection Reader*, which reads a list of search queries from a text file. The Query Collection Reader is a UIMA "collection reader" that reads the text queries into memory data structures (UIMA CAS structures) — one for each text query. These UIMA CASes flow through the pipeline and are processed by the various processing stages. The processing stages are set up so that new components designed to perform the task of each processing stage can easily be added to the pipeline (or existing components easily modified). The expected inputs and outputs of components in each processing stage are clearly defined, which makes the task of the team building the component simpler: they no longer have to deal with managing data structures and are spared the overhead of converting from and into formats of data exchanged between various components. All of the overhead is handled by UIMA. Furthermore, some of the processing stages generate new CAS structures and the flow of all the UIMA CAS structures through this pipeline is controlled by a "Flow Controller" designed by us for this framework.

The framework was made available to each of the student teams, and their task was to build their project by extending this framework. Even though we built the framework to perform semantic search over a text corpus, many of the teams in this course had projects that went far beyond just semantic search. Our hope was that each team would be able to able independently develop interesting new components for the processing stages of the pipeline, and at the end of the course we would be able to merge the most interesting components to create a single useful application. In the following section, we describe the various projects undertaken by the student teams in the class, while Section 5 discusses the integration of components from student projects and the demo application that resulted from the integrated system.

4 Class Projects

Projects completed for this course fall into three types: scientific projects, where the aim is to produce a publishable paper; integrated projects, where the aim is to create a component that will be integrated into the class open-source project; and independent demo projects, where the aim is to produce an independent working demo/prototype. The following section describes the integrated projects briefly.

4.1 Selected Project Descriptions

As described in section 3, the integrated class project is a system with an architecture which, although greatly simplified, is reminiscent of Watson's. While originally intended to be simply a semantic search tool, some of the student teams created additional components which resulted in a full question answering system. Those projects as well as a few other related ones are described below.

Question Categorization: Using the DBPedia ontology (Bizer et al., 2009) as a semantic type system, this project classifies questions by their answer type. It can be seen as a simplified version of the question categorization system in Watson. The classification is based on a simple bag-of-words approach with a few additional features.

Answer Candidate Ranking: Given the answer type as well as additional features derived by the semantic search component, this project uses regression to rank the candidate answers which themselves come from semantic search.

Twitter Semantic Search: Search in Twitter is difficult due to the huge variations among tweets in lexical terms, spelling and style, and the limited length of the tweets. This project employs LSA (Landauer and Dumais, 1997) to cluster similar tweets and increase search accuracy.

Fine-Grained NER in the Open Domain: This project uses DBPedia's ontology as a type system for named entities of type *Person*. Given results from a standard NER system, it attempts to find the fine-grained classification of each *Person* entity by finding the most similar type. Similarity is computed using traditional distributional methods, using the context of the entity and the contexts of each type, collected from Wikipedia.

News Frame Induction: Working with a large corpus of news data collected by Columbia Newsblaster, this team used the Machine Linking API to tag entities with semantic types. From there, they distributionally collected "frames" prevalent in the news domain such as '[U.S President] meeting with [British Prime Minister]'.

Other projects took on problems such as Sense Induction, NER in the Biomedical domain, Semantic Role Labeling, Semantic Video Search, and a mobile app for Event Search.

5 System Integration and Demonstration

The UIMA-based architecture described in section 3 allows us to achieve a relatively easy integration of different class projects, independently developed by different teams, in a common architecture and expose their functionality with a combined class project demo. The demo is a collaboratively developed semantic search engine which is able to retrieve knowledge from structured data and visualize it for the user in a very concise way. The input is a query; it can be a natural language question or simply a set of keywords. The output is a set of entities and their relations, visualized as an entity graph. Figure 3 shows the results of the current status of our class project demo on the following Jeopardy! question.

> *This nation of 200 million has fought small independence movements like those in Aceh and East Timor.*

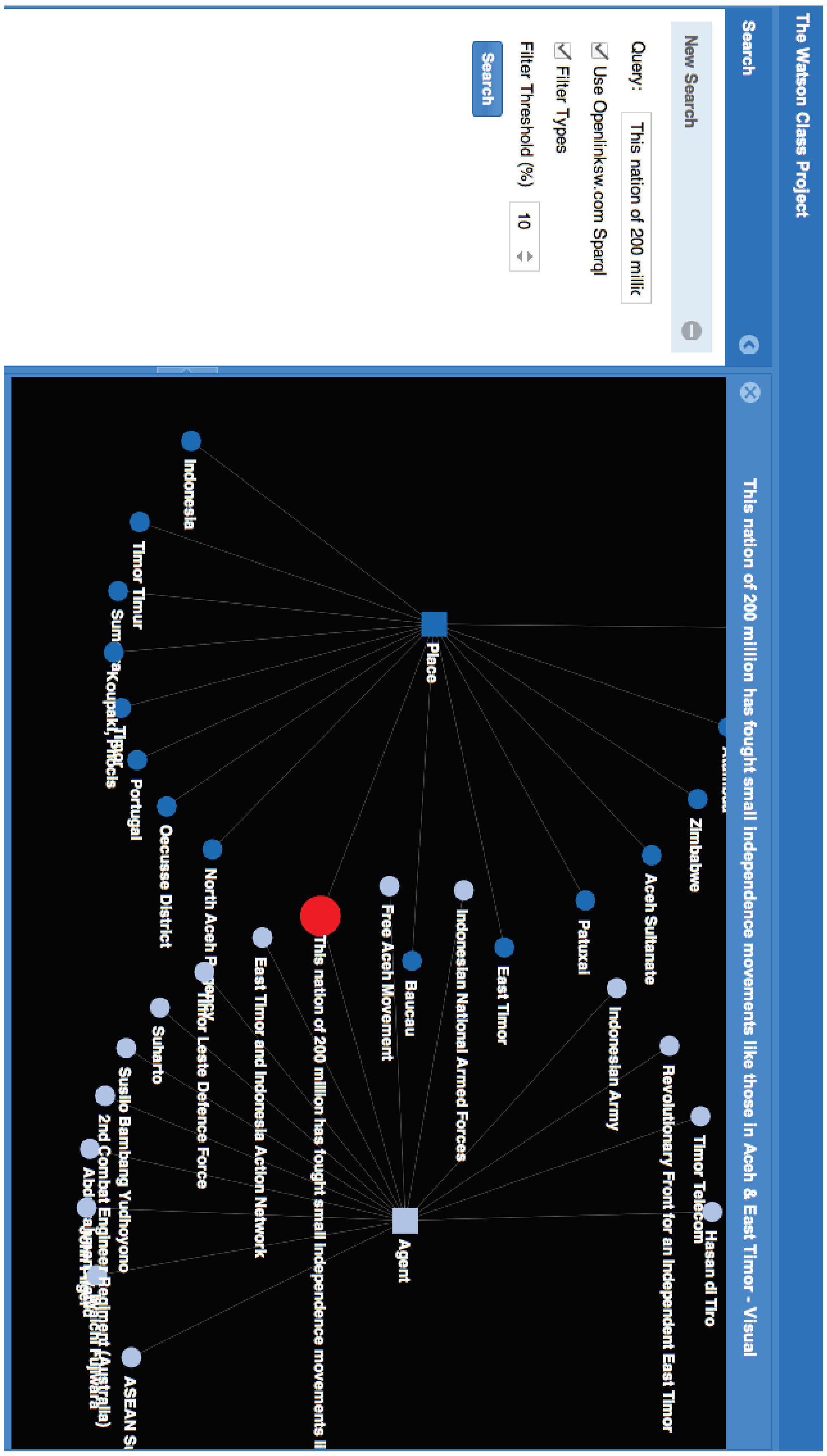
The output is a set of DBPedia entities related to the question, grouped by Type (provided by the DBPedia ontology). The correct answer, "*Indonesia*", is among the candidate entities of type Place. Note that only answers of type *Place* and *Agent* have been selected: this is due to the question categorization component, implemented by one of the student teams, that allows us to restrict the generated answer set to those answers having the right types.

The demo will be hosted for one year following the end of the course at `http://watsonclass.no-ip.biz`. Our goal is to incrementally improve this demo, leveraging any new projects developed in future versions of the course, and to build an open source software community involving students taking the course.

6 Evaluation

The course at Columbia drew a relatively large audience. A typical size for a seminar course on a special topic is estimated at 15-20 students, while ours drew 35. The vast majority were Master's students; there were also three PhD students and five undergraduates.

During the student workshops, students were asked to provide grades for each team's presentation and project. After the instructor independently gave his own grades, we looked at the correlation between the average grades given by the students and those give by the instructor. While


The Watson Class Project
Search
This nation of 200 million has fought small independence movements like those in Aceh & East Timor - Visual
Place
Agent
Indonesia
Timor Timur
Sum
Koupaki, Timor
Portugal
Oecusse District
North Aceh Regency
Zimbabwe
Aceh Sultanate
Patuxal
East Timor
Baucau
Free Aceh Movement
Indonesian National Armed Forces
Indonesian Army
Revolutionary Front for an Independent East Timor
Timor Telecom
Hasan di Tiro
This nation of 200 million has fought small independence movements li
East Timor and Indonesia Action Network
Timor Leste Defence Force
Suharto
Susilo Bambang Yudhoyono
2nd Combat Engineer Regiment (Australia)
ASEAN St

New Search

Query: This nation of 200 millic

☑ Use Openlinksw.com Sparql

☑ Filter Types

Filter Threshold (%) 10

Search

Figure 3: Screenshot of the project demo

Team	1	2	3	4	5	6	7	8	9	10	11
Instructor's grade	B+	B	C+	A-	B-	A+	B	B-	B+	A	B-
TA's grade	B+	B	B	A	B-	A	B-	B+	B+	A	C+
Class' average grade	B/B+	B+/A-	B/B+	A-	B/B+	A-	B+	A-/A	B+/A-	A-/A	B/B+

Table 1: Grades assigned to class projects

the students tended to be more "generous" (their average grade for each team was usually half a grade above the instructor's), the agreement was quite high. Table 1 shows the grades given by the instructor, the teaching assistant and the class average for the midterm workshop.

Feedback about the course from the students was very good. Columbia provides electonic course evaluations to the students which are completely optional. Participation in the evaluation for this course was just under 50% in the midterm evaluation and just over 50% in the final evaluation. The scores (all in the 0-5 range) given by the students in relevant categories were quite high: "Overall Quality" got an average score of 4.23, "Amount Learned" got 4, "Appropriateness of Workload" 4.33 and "Fairness of Grading Process" got 4.42.

The course resulted in multiple papers that are or will soon be under submission, as well as a few projects that may be developed into start-ups. Almost all student teams agreed to share their code in an open source project that is currently being set up, and which will include the current question answering and semantic search system as well as additional side projects.

7 Conclusion

We described a course on the topic of Semantic Technologies and the IBM Watson system, which features a diverse curriculum tied together by its relevance to an exciting, demonstrably successful real-world system. Through a combined architecture inspired by Watson itself, the students get the experience of developing an NLP-heavy component with specifications mandated by the larger architecture, which requires a combination of research and software engineering skills that is common in the industry.

An exciting result of this course is that the class project architecture and many of the student projects are to be maintained as an open source project which the students can, if they choose, continue to be involved with. The repository and community of this project can be expected to grow each time the class is offered. Even after one class, it already contains an impressive semantic search system.

Feedback for this course from the students was excellent, and many teams have achieved their personal goals as stated at the beginning of the semester, including paper submissions, operational web demos and mobile apps.

Our long term goal is to replicate this course in multiple top universities around the world. While IBM does not have enough resources to always do this with its own researchers, it is instead going to provide the content material and the open source code generated so far to other universities, encouraging professors to teach the course themselves. Initially we will work on a pilot phase involving only a restricted number of professors and researchers that are already in collaboration with IBM Research, and eventually (if the positive feedback we have seen so far is repeated in the pilot phase) give access to the same content to a larger group.

References

C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. 2009. DBpedia—Crystallization Point for the Web of Data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, September.

J. Chu-Carroll, J. Fan, B. Boguraev, D. Carmel, D. Sheinwald, and C. Welty. 2012a. Finding Needles in the Haystack: Search and Candidate Generation. *IBM Journal of Research and Development*, 56(3.4):6:1–6:12.

J. Chu-Carroll, J. Fan, N. Schlaefer, and W. Zadrozny. 2012b. Textual Resource Acquisition and Engineering. *IBM Journal of Research and Development*, 56(3.4):4:1–4:11.

E. Epstein, M. Schor, B. Iyer, A. Lally, E. Brown, and J. Cwiklik. 2012. Making Watson Fast. *IBM Journal of Research and Development*, 56(3.4):15:1–15:12.

J. Fan, A. Kalyanpur, D. Gondek, and D. Ferrucci. 2012. Automatic Knowledge Extraction from Documents. *IBM Journal of Research and Development*, 56(3.4):5:1–5:10.

D. Ferrucci and A. Lally. 2004. UIMA: an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10(3-4):327–348.

D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. 2010. Building Watson: An Overview of the DeepQA project. *AI magazine*, 31(3):59–79.

D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. Mueller. 2012. Watson: Beyond Jeopardy. *Artificial Intelligence (in press)*.

D. Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4):1:1–1:15.

A. Gliozzo and T. Isabella. 2005. Semantic Domains in Computational Linguistics. Technical report.

A. Kalyanpur, B. Boguraev, S. Patwardhan, J. W. Murdock, A. Lally, C. Welty, J. Prager, B. Coppola, A. Fokoue-Nkoutche, L. Zhang, Y. Pan, and Z. Qiu. 2012. Structured Data and Inference in DeepQA. *IBM Journal of Research and Development*, 56(3.4):10:1–10:14.

A. Lally, J. Prager, M. McCord, B. Boguraev, S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll. 2012. Question Analysis: How Watson Reads a Clue. *IBM Journal of Research and Development*, 56(3.4):2:1–2:14.

T. Landauer and S. Dumais. 1997. A Solution to Plato's Problem: the Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. *Psychological Review*, 104(2):211–240.

B. Lewis. 2012. In the Game: The Interface between Watson and Jeopardy! *IBM Journal of Research and Development*, 56(3.4):17:1–17:6.

M. McCord, J. W. Murdock, and B. Boguraev. 2012. Deep Parsing in Watson. *IBM Journal of Research and Development*, 56(3.4):3:1–3:15.

T. Miller, C. Biemann, T. Zesch, and I. Gurevych. 2012. Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In *Proceedings of the International Conference on Computational Linguistics*, pages 1781–1796, Mumbai, India, December.

J. W. Murdock, J. Fan, A. Lally, H. Shima, and B. Boguraev. 2012a. Textual Evidence Gathering and Analysis. *IBM Journal of Research and Development*, 56(3.4):8:1–8:14.

J. W. Murdock, A. Kalyanpur, C. Welty, J. Fan, D. Ferrucci, D. Gondek, L. Zhang, and H. Kanayama. 2012b. Typing Candidate Answers Using Type Coercion. *IBM Journal of Research and Development*, 56(3.4):7:1–7:13.

J. Prager, E. Brown, and J. Chu-Carroll. 2012. Special Questions and Techniques. *IBM Journal of Research and Development*, 56(3.4):11:1–11:13.

G. Tesauro, D. Gondek, J. Lenchner, J. Fan, and J. Prager. 2012. Simulation, Learning, and Optimization Techniques in Watson's Game Strategies. *IBM Journal of Research and Development*, 56(3.4):16:1–16:11.