Explaining Watson: Polymath Style

Wlodek Zadrozny
UNC Charlotte
wzadrozn@uncc.edu

Valeria de Paiva
Nuance
valeria.depaiva@nuance.com

Lawrence S. Moss
Indiana University
lmoss@indiana.edu

Abstract

Our paper is actually two contributions in one. First, we argue that IBM's Jeopardy! playing machine needs a formal semantics. We present several arguments as we discuss the system. We also situate the work in the broader context of contemporary AI. Our second point is that the work in this area might well be done as a broad collaborative project. Hence our "Blue Sky" contribution is a proposal to organize a *polymath*-style effort aimed at developing formal tools for the study of state of the art question-answer systems, and other large scale NLP efforts whose architectures and algorithms lack a theoretical foundation.

Introduction

IBM Watson, the Jeopardy! playing machine, significantly advanced the state of the art in natural language processing and question answering (Ferrucci and others 2012). However, despite a comprehensive description of its algorithms and architecture in a dozen papers in a 2012 issue of IBM J. of Research and Development, a few dozen patent applications, and several newer papers by IBM researchers, there seem to be no abstract, scientific explanation of Watson's success. Furthermore, four years later, we are aware of no replication of Watson's results, i.e. a question answering system with the accuracy of a Jeopardy! champion.

Watson's ways of achieving (partial) language understanding are significantly different from the ones used in earlier projects. When trying to understand the meaning of a sentence, Watson doesn't adhere to the standard syntax-semantics division of labor. In fact, by using information retrieval techniques to generate candidate answers and find supporting evidence, one can argue that Watson almost completely abandons the standard view of semantics, at least when it comes to question answering.

On the other hand, Watson uses almost all the techniques developed by computational linguists (parsing, building of logical form, reference resolution, etc.). These techniques are used to create a collection of *scores* that account for certain aspects of syntax and semantics, but only in the context of the question and supporting evidence for a candidate answer. In addition Watson has a view of the background knowledge as organized around collection of concepts, as *title-oriented resources*, (Chu-Carroll et al. 2012), and not as a big flat knowledge base (which still seem to be the canonical view, e.g. (Russell and Norvig 2009) p.235).

We argue that AI needs a project to explain what Watson does in some form of abstract semantics. We believe a formal account should accelerate progress in natural language understanding, question answering and possibly other domains. Without it, the community can't achieve deeper understanding of what are Watson's advantages and limitations.

This paper is organized as follows: After this Introduction we discuss the need for an abstract theory of Watson. We focus on problems derived directly from the computational architecture of Watson. The follow-on section explores the opportunities of connecting Watson to prior work in semantics and knowledge representation. The final section explores organization of such research in a polymath fashion.

An abstract computational theory underlying Watson?

Technical details on Watson are included in the overview papers (Ferrucci and others 2010), patents (e.g. (Brown et al. 2012; Fan et al. 2012)), and patent applications (available at http://patft.uspto.gov/netahtml/PTO/index.html) as well as newer work arising from the project e.g. (Lally et al. 2014). As a result of these disclosures, we know *how* Watson works and some of its possible applications (e.g. (Gopinath 2013; Lally et al. 2014)). However, we do not know *why* Watson works. That is, given that Watson's architecture and approach differs from prior work, to what extent are these differences fundamental? What are the underlying mathematical and computational mechanisms?

We believe there should be a computational theory underlying Watson. A few examples to seed a discussion appear below, looking at what the Watson team considers their main innovations.

Heterogeneity of Natural language understanding: Watson evaluates multiple aspects of available information.

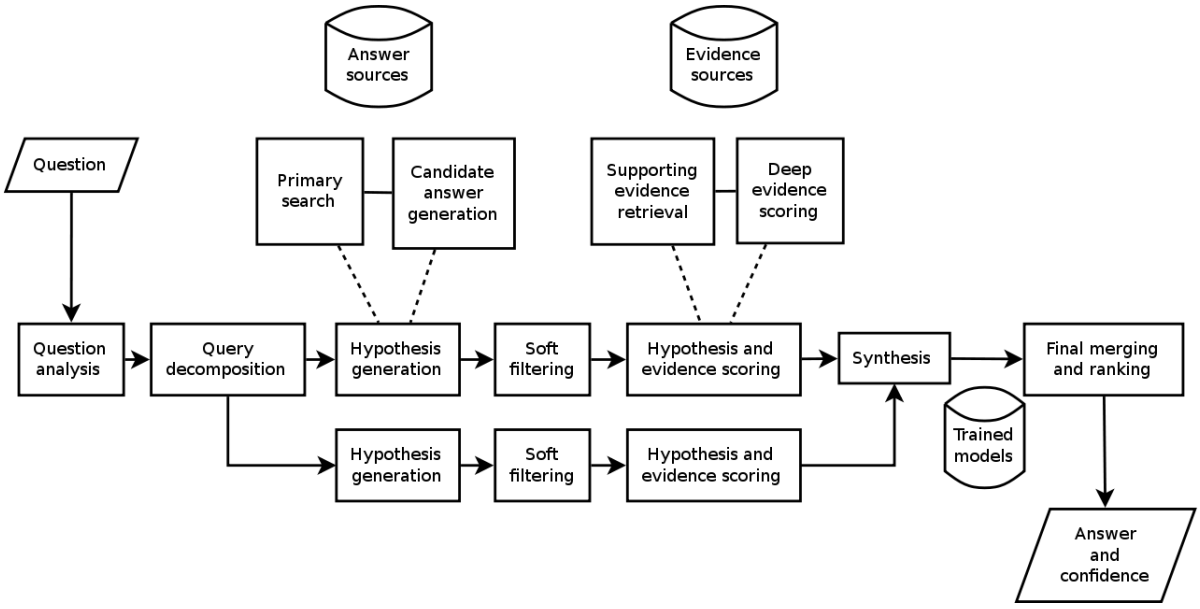

Figure 1: Watson architecture. A question is answered by generating candidate answers from document titles, finding supporting evidence, and evaluating the evidence using over 400 scorers. Source: Wikipedia

This is done by retrieving dozens of paragraphs of text with supporting evidence, using multiple programs to score the evidence, and synthesizing the final score (Fig.1). In other words, Watson uses heterogeneous sources of information, creates multiple representations, and evaluates using different kinds of scorers.

A recent paper (Schütze 2013) argues that semantics should be heterogeneous. It also contains the following passage about Watson: "*My impression* is that in these systems interaction is implemented in an ad-hoc and manually optimized way. For example, the IBM question answering system Watson (Ferrucci et al., 2010) has many heterogeneous modules that each contribute information useful for solving a Jeopardy task. The final design of the components that integrate this information into the Jeopardy! answer *seems to have been the result of careful and time-intensive fine tuning*. (italics ours).

As we see it, Watson's formal semantics have yet to be formulated. And proposing a semantics would help answer questions already asked by others.

Role of search in question answering As an initial step in Watson, **primary search** generates a few hundred candidate answers consisting of the **title** fields of matching documents (see Fig. 1). This works well for Jeopardy!-style questions, where many answers are entities with entries in Wikipedia or other encyclopedic sources. However we don't know how applicable this strategy is for question types other than those in Jeopardy! Can *any* text collection be organized (by a computational procedure) in such a way that candidate answers may be generated by search? Under what conditions can we say that a piece of text is *about* a given topic? Can Watson's title-oriented data organization be created dynamically? What if answers need to be synthesized from pieces of information coming from different documents?

Document search with titles serving as answers can be contrasted with steps in QA before Watson: Question classification (who, what, when, how big,...); passage search to find passages with matching terms or relations; interpretation (typically using shallow or partial compositional semantics); matching relations ('unification') to extract an answer; and estimating the likelihood of the answer being correct. (See e.g. (Moldovan et al. 2003b))

Semantics of the deferred type evaluation Another Watson innovation is **deferred type evaluation** (Fan et al. 2012) (US Patent 8332394, 2012). Again, we ask whether it is fundamental or a piece of lucky heuristics? An anecdotal argument for its fundamental value can cobbled from a set of examples in the patent. The examples show the creativity of language precludes enumeration of all types of answers. Consider the question "which 19th century US presidents were assassinated?" To answer it, one can easily find a list of US Presidents (and use it as an ontology/set of types), but we are unlikely to find a list of 19th century (American) presidents. (For that matter, we are unlikely to find a list of presidents killed in a theater.) Therefore Watson's strategy is to answer a query by *waiting* until an "answer type" (i.e., a descriptor) is determined *and* a candidate answer is provided. The answer type is not required to belong to a predetermined ontology, but is viewed as lexical/grammatical item. Then, a search is conducted to look for evidence that the candidate answer has the required answer type. The match of candidate answer with the answer type is determined by a set of scorers using e.g. a parser, a semantic interpreter or a simple pattern matcher). We ask whether this procedure can be formalized and evaluated theoretically.

Semantics of scoring Thirdly, Watson uses **multiple scorers** to estimate degrees of match between a question and a candidate answer along a few hundred dimensions. Many of the scorers are evaluating quantities familiar from computational semantics (e.g. matching of predicate-argument structures). But some are not. Does this strategy of employing multiple scorers lead to a dialogue-like view of meaning? Such an alternative would generate candidate interpretations, then find supporting evidence, and finally evaluate the candidates. Meaning would reside in this process. But what further implications does such a semantics have? And returning to Watson, is this use of multiple scorers really a deep contribution, or is an engineering feat but nothing more?

Questions

We now go into more detail about the other broad kinds of questions which we hope to address.

Which kind of formal semantics? The Watson team did not pay much attention to formal natural language semantics. However, they incorporated most of the known computational tools. The question is why this combination of tools worked better than e.g. (Moldovan et al. 2003b), who showed that paying more attention to formal semantics can significantly improve QA results, that is "the overall performance of QA systems is directly related to the depth of NLP resources", and "the prover boosts the performance of the QA system on TREC questions by 30%" (Moldovan et al. 2003a). Similar claims about improvements provided by logical tools are found in (MacCartney and Manning 2009) At this point is not clear that Watson style of NLP can even be cast in a formal semantics.

How can large systems like Watson integrate statistical and symbolic methods? Along with many other researchers, we believe that both statistical methods and structural ones have their own place in computational and formal linguistics. The question of integrating these two approaches is of interest to many in the field. We know on the architectural level how Watson carries out this integration (from patents and the journal collection). However what are its foundational aspects? For example, what would be a natural formalism to describe it? How would it be related to other proposals in the field such as the BLOG language (Milch et al. 2007) combining probability with logic?

What are the appropriate foundations for automated deduction when reasoning for common sense? One avenue of investigation could be to adapt "automated deduction" from the more mathematical setting of interactive theorem provers such as Coq and HOL to the more AI-oriented setting of automatic question-answering. In fields like KR, databases (where the necessary assumptions for proofs need to be found) are too huge, but shallow. Similarly, 'proofs' only need a few inference steps, but finding the correct axioms/theories is the difficult part. Hence, for such fields, the appropriate notion of 'complexity' will be different from the one from traditional theoretical computer science. Complexity for question answering systems should somehow measure the size and kind of its collection of data and not simply the worst-case, or average-case, case complexity of inference.

We believe that a thorough adaptation of Automated Theorem Proving to KR scenarios would be interesting. In fact this adaptation has already been started in the form of the "track/division" called 'Large Theory Base' (LTB) of the CASC (CADE ATP System Competition). However this competition is restricted to theorem proving systems using the TPTP (Thousands of Problems for Theorem Provers) language. Thus this adaptation has been, so far, less ambitious, and frankly less "blue sky" than what we want to do. Given that our intent is more experimentation, less competition, we would like to try different provers with different sets of KR languages, aiming for compatibility of formalisms and hidden synergies between modules.

Recognizing that logical inference is necessary to build NL representations, as well as necessary to reason about information encoded in representations aleady in the system, and that the reasoning with representations already created can be much simpler, we would like to investigate tradeoffs between these two sets of logical inferences. The traditional mechanisms of over-simplification of parts of processing can be useful here. One should be able to reason easily with representations, if we concentrate the difficult issues of creating representations in some kind of blackbox that we can instantiate with different formalisms, be they description logics, first-order logics or higher-order logics (Nigam and de Paiva 2014). Perhaps instead we want to stay as close as possible to Natural Language, using one of the several versions of Natural Logic (cf., e.g.(Moss 2015)) recently at our disposal. Recent work on recognizing textual entailment (RTE) (Dagan et al. 2010; Sammons 2015) ought to be helpful as a way of testing formalisms and their inferential capabilities.

What does Big Data do for Watson? And what it does not do? We would like to know to what extent Watson's success is based on the large amount of training data: 200K available questions with answers, and about 50K of them used for training and testing. To what extent it is dependent on the highly parallel aspect of its scoring system? Parallelism was essential given the amount of background textual knowledge Watson relied on. But was less important when the background knowledge was much smaller. How do these factors interact with each other, with available background knowledge, and with deeper semantic processing? What are the underlying mechanisms of these interactions?

Prior Work in the Area

There seem to be no scholarly work on semantics of Watson. Our repeated searches on Google Scholar yield zero results addressing the problem. IBM Watson is mentioned in very few relevant contexts. For example, there are only 27 entries for "watson ibm jeopardy 'formal semantics'. For other searches the result is lower (Sept 26 2014, excluding citations). However, none of the papers presented a formal view of the complete Watson system. On the other hand, there are some papers to build on, including (Noy and McGuinness

2013; Iyyer et al. 2014).

Polymath Projects

We propose to organize a *polymath* project, an online massively cooperative blog (see (Wikipedia 2014; Gowers et al. 2014)). The name comes from mathematics, where Tim Gowers has led nine projects, including some where actual theorems have been proved. As the New Scientist notes in their editorial "Mathematics becomes more sociable" of 05 May 2011, "The first analysis of the Polymath project concludes that online collaborations set up in the right way can solve problems with unprecedented speed and efficiency." Of course, what we propose here is not the solution of concrete mathematical problems but rather the initiation of a research area. That is, the results would be a set of new concepts and ideas (all informal), and also formal definitions, test results, problems; in addition, we feel that the much of the work in this area will involve simulation and the borrowing of ideas and programs from other areas of AI, Cognitive Science, Computational Linguistics, and beyond.

We imagine we would moderate and seed the discussion, and contribute to solving the problems, but the problem(s) will be addressed by the whole community. The majority of insights could be provided by the community of researchers interested in the topic.

Experimenting in Watson Polymath

We imagine the Watson Polymath project to have at least two dimensions: The first would be discussion of new ideas, and maintenance of competing theories, e.g. in the form of a wiki. Second, because we are dealing with the topic of language understanding under a broad conception, there must be an experimental component. We already presented some starting points for the former, so let's discuss the resources for experimentation.

As mentioned above, IBM has been publishing a lot about Watson, but the basic idea behind Watson is that you should be able to plug-and-play modules, and perform experimental evaluations. Thus we imagine systematically evaluating selected combinations of open source NLP resources such as parsers and entity extractors. Lately, IBM has begun to allow access to Watson APIs (Sudarsan 2014) for selected groups (e.g. entrepreneurs, faculty and students) (Sudarsan 2014). In addition, it has been shown in classes at Columbia University (Gliozzo et al. 2013), and at Rensselear Polytechnic (RPI) (Hendler 2013; Hendler and Ells 2014), that selected portions of the Watson pipeline can be reconstructed. More recently, in Spring of 2014, almost complete simulation of the Watson architecture and several textual resources were created in the class on "Semantic technologies in IBM Watson" at UNC Charlotte (Zadrozny and Gallagher 2015). This class developed and/or used knowledge sources in TREC format (versions of Wikipedia, Wiktionary, Wikiquotes, etc.), Eclipse workspaces allowing experiments with various search engines and plug in NLP components. links to experiment evaluation software, and documentation.

Additional Possible Targets for New Theories

As mentioned above we believe there are opportunities for interesting research on ideas and techniques inspired by Watson. In particular, we mentioned a "heterogeneous" semantics of NL and related to it semantics of scoring; topicality of knowledge (i.e. title- or topic-oriented organization of knowledge), and deferred type evaluation. There are obvious connections between these problems and the domains of natural language processing and knowledge representation.

But there additional areas that can be targeted for formalization, or even formulation of semi-formal hypotheses of what is going on. (We imagine there would be also experimental work in these targeted areas). Thus, for example, we would like to know more about the trade-off between adding semantics to candidate answers search vs. deeper semantics of supporting passage retrieval vs. deeper semantics of scoring. This has intuitive connections to the work on "faceted search" (Tunkelang 2009) and "semantic search" (Hendler 2010), since many scorers could be in principle used to create facets. And yet would probably be different, because the Watson team showed there isn't one ontology (or set of topics) covering all types of Jeopardy! questions ,

The Watson team used almost all known NLP techniques. We can ask whether there are any types of natural language understanding (NLU) components are NOT useful for Watson-like tasks or architectures? Or, in other words, what are the trade-offs between different classes of components. For example can adding additional types of faceted search decrease the need for scorers (keeping the accuracy constant).

We are not aware of a theory of multi-stage scoring used in Watson(Agarwal et al. 2012), nor of the mathematics of composing Machine Learning (ML) systems, with very large numbers of scorers of different types (e.g. a group scoring quality of sources, and another group providing variants of predicate argument structures). How different can or should the ML systems be? How does one decide on the optimal number of scorers? This might be connected to work on structure learning and ensemble learning (e.g. (Cortes, Kuznetsov, and Mohri 2014; Mendes-Moreira et al. 2012)), but we are not aware of an explicit theory.

Conclusions

We argued there is need for an abstract explanation of the Watson success. In addition, we propose to work on this problem in a polymath-style project, that is, as an open collaborative project. An argument can be made for this approach both based on the success of the original Polymath projects, and from general trends towards open science and open research (e.g. (Nielsen 2012)). Given Watson's importance and publicity, there's a good chance that such a project would be successful; we expect both quality scientific papers and new algorithms. With the partial opening of the Watson APIs by IBM, we can even hope to influence the development of applications in not so distant future.

References

Agarwal, A.; Raghavan, H.; Subbian, K.; Melville, P.; Lawrence, R. D.; Gondek, D. C.; and Fan, J. 2012. Learning to rank for robust question answering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, 833–842. New York, NY, USA: ACM.

Brown, E.; Ferrucci, D.; Lally, A.; and Zadrozny, W. W. 2012. System and method for providing answers to questions. US Patent 8,275,803.

Chu-Carroll, J.; Fan, J.; Schlaefer, N.; and Zadrozny, W. 2012. Textual resource acquisition and engineering. *IBM Journal of Research and Development* 56(3.4):4:1–4:11.

Cortes, C.; Kuznetsov, V.; and Mohri, M. 2014. Ensemble methods for structured prediction. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 1134–1142.

Dagan, I.; Dolan, B.; Magnini, B.; and Roth, D. 2010. Recognizing textual entailment: Rational, evaluation and approaches–erratum. *Natural Language Engineering* 16(01):105–105.

Fan, J.; Ferrucci, D.; Gondek, D. C.; and Zadrozny, W. W. 2012. System and method for providing questions and answers with deferred type evaluation. US Patent 8,332,394.

Ferrucci, D., et al. 2010. Building Watson: An overview of the DeepQA project. *AI Magazine* 31(3).

Ferrucci, D. A., et al. 2012. Introduction to "This is Watson". *IBM J. RES. and DEV* 56(3).

Gliozzo, A.; Biran, O.; Patwardhan, S.; and McKeown, K. 2013. Semantic technologies in IBM WatsonTM. *ACL 2013, http://www.aclweb.org/anthology/W13-3413* 85.

Gopinath, R. 2013. Watson goes global: How IBM Research India is advancing cognitive computing. *http://asmarterplanet.com/blog/2013/10/watson-goes-global-how-ibm-research-india-is-advancing-cognitive-computing.html*.

Gowers, T.; Kalai, G.; Nielsen, M.; and Tao, T. 2014. The polymath blog. `http://polymathprojects.org/`.

Hendler, J., and Ells, S. 2014. Retrieved September 03 2014 from http://www.slideshare.net/jahendler/why-watson-won-a-cognitive-perspective?next_slideshow=1.

Hendler, J. 2010. Web 3. 0: The dawn of semantic search. *Computer* 43(1):77–80.

Hendler, J. 2013. Watson at RPI. Retrieved September 03 2014 from http://www.slideshare.net/jahendler/watson-summer-review82013final.

Iyyer, M.; Boyd-Graber, J.; Claudino, L.; Socher, R.; and III, H. D. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of EMNLP 2014*.

Lally, A.; Bagchi, S.; Barborak, M. A.; Buchanan, D. W.; Chu-Carroll, J.; Ferrucci, D. A.; Glass, M. R.; Kalyanpur, A.; Mueller, E. T.; Murdock, J. W.; Patwardhan, S.; Prager, J. M.; and Welty, . C. A. 2014. Watsonpaths: Scenario-based question answering and inference over unstructured information.

MacCartney, B., and Manning, C. D. 2009. An extended model of natural logic. In *Proceedings of the eighth international conference on computational semantics*, 140–156. Association for Computational Linguistics.

Mendes-Moreira, J. a.; Soares, C.; Jorge, A. M.; and Sousa, J. F. D. 2012. Ensemble approaches for regression: A survey. *ACM Comput. Surv.* 45(1):10:1–10:40.

Milch, B.; Marthi, B.; Russell, S.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2007. 1 blog: Probabilistic models with unknown objects. *Statistical relational learning* 373.

Moldovan, D.; Clark, C.; Harabagiu, S.; and Maiorano, S. 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*, NAACL '03, 87–93. Stroudsburg, PA, USA: Association for Computational Linguistics.

Moldovan, D.; Paşca, M.; Harabagiu, S.; and Surdeanu, M. 2003b. Performance issues and error analysis in an open-domain question answering system. *ACM Trans. Inf. Syst.* 21(2):133–154.

Moss, L. S. 2015. Natural logic. In *Handbook of Contemporary Semantic Theory, Second Edition*. Wiley. 761–802.

Nielsen, M. 2012. *Reinventing Discovery: the New Era of Networked Science*. Princeton University Press.

Nigam, V., and de Paiva, V. 2014. Towards a rewriting framework for textual entailment. In *Pre-proceedings of the 9th Workshop on Logical and Semantic Frameworks, with Applications, Brasilia, Brazil, Sept 2014*.

Noy, N., and McGuinness, D., eds. 2013. *Research Challenges and Opportunities in Knowledge Representation, Final Report on the 2013 NSF Workshop on Research Challenges and Opportunities in Knowledge Representation*. NSF.

Russell, S., and Norvig, P. 2009. Artificial intelligence: A modern approach. *Prentice Hall*.

Sammons, M. 2015. Recognizing textual entailment. In *Handbook of Contemporary Semantic Theory, Second Edition*. Wiley. 711–757.

Schütze, H. 2013. Two maps of Manhattan. In King, T. H., and de Paiva, V., eds., *From Quirky Case to Representing Space: Papers in Honor of Annie Zaenen*. Center for Study Language and Information, Stanford.

Sudarsan, S. 2014. Creating cognitive applications powered by IBM Watson: Getting started with the API.

Tunkelang, D. 2009. *Faceted Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers.

Wikipedia. 2014. Polymath project. `http://en.wikipedia.org/wiki/Polymath_Project`.

Zadrozny, W., and Gallagher, S.and Shalaby, W. 2015. Simulating IBM Watson in the Classroom. *to appear in: Proc. ACM SIGCSE 2015* 6pp.