State of the art on Prolog use for Natural Language Processing

Dinis Marques Firmino - P13240786

November 3, 2017

Abstract

Our brain has evolved complex mechanisms to be able to reason with the ever so complex environment around us. One of these mechanisms gives humans the ability to reason with language, and the process of giving computers the ability to share the same level of language reasoning has proven to be one of toughest challenges in computing. Natural language processing is the field that aims to tackle these challenges and is the main focus of this state of the art report. More specifically, we'll look at recent uses of Prolog in approaches to problems in the various domains of NLP.

Keywords: Prolog, Natural language processing, Q/A systems, Knowledge representation, Understanding

1 Introduction

Natural language processing (NLP for short) is a crossover between the fields of computer science, AI and computational linguistics, and concerns itself with giving computers the ability to reason with language. This field explores a vast sector of research and faces some of the toughest challenges in computing due to the inherent difficulty in creating systems that exhibit performance on par with the human brain when reasoning with language. This is no easy feat because of the ambiguity, fuzziness, and the complex syntactic and semantic relationships associated with human language. Traditional methods of binary logic are not sufficient for dealing with such complex problems.

Similarly to other machine learning fields, advancing the field of NLP is crucial because of its vast applications in machine translation, conversation agents, sentiment analysis and natural interfaces between humans and information stored within computer systems. All these applications could have a direct impact and generate tremendous value in the sectors of medicine, education and business.

Prolog developed in 1972 [1] was an attempt at tackling the problem of NLP. It is a declarative logic programming language and functions by using recursive mechanisms to assert truths within a knowledge base, provided that it was given

facts and rules that describe the domain of interest. It differs from traditional procedural languages by allowing the programmer to focus on the objects in the program and the relationships that connect them, instead of the general procedure of the program.

Due to the vastness of the NLP field, this paper is split into various sections each covering a specific topic within the field. Each section aims to briefly explain the main topic from an NLP perspective and provide some examples of recent work. The sections are as follows:

Section 2 focuses on knowledge representation, and its importance in the field of NLP and many others. A total of 3 papers have been reviewed all of which contain some form of Prolog use in their approaches.

Section 3 explores the vast topic of question answering (Q/A) systems, explaining briefly what the field is, and more thoroughly some instances of Prolog use in helping the resolution of problems proposed in this topic such as those in DeepQA [5], the framework that powers IBM's Watson.

Section 4 is on natural language understanding or NLU for short. This section aims to cover the very core of most NLP applications and how Prolog has recently helped contribute value to the field.

Lastly, we wrap up the report by providing a solid conclusion while referring to the content in the report and how it helped shed light on Prolog's state of the art within the various sectors of NLP, and offering a viewpoint on the its future.

2 Knowledge Representation

The bulk of our civilizations information is stored on the web, in books and academic journals written in natural language, mostly in an unstructured form. To deal with this lack of structure, approaches using NLP have been at the forefront of successes in fields such as semantic web [2]. The importance of representing this knowledge in a structured way is vital in further advancing the field of computational intelligence and many others. This is because data is at the core of techniques used to resolve problems in many fields.

Below are a number of recent papers that were reviewed which feature systems that use Prolog in some form, and contribute novel approaches to the sector of knowledge representation within NLP.

The first paper [2] compares two methods for the automatic creation of an ontology, one purely Prolog based, and the other a cloud approach using Watson through the Bluemix cloud API. The aim of the research is to find alternatives to the manual creation of ontologies. This is important because ontologies take a significant amount of time to create and require specialized engineers to establish relationships between information to create a structure.

Focusing on the Prolog approach, the basis of it is around the execution of grammar and logic analysis of the text using Prolog rules, and WordNet as the lexical database for the vocabulary. With the aid of Thea [**?**], the result of these analyses is used to construct a knowledge base that represents OWL2 [**?**]

axioms as Prolog facts. Using Thea, these facts are translated to an ontology in the official OWL2 format. The Pure Prolog approach produced acceptable results in representing a correct ontology shown in figure 1 of [2], but lacked in vocabulary size and performance when contrasted with the Watson services. This was due to the constraint of WordNet, and design decisions which refrained from the full use of Prolog's recursive nature. However, the Prolog approach is more flexible due to not being constrained to an API.

The second paper [4], uses a similar approach to the research above by making use of ontologies, but are applied to a different application domain of NLP. This research instead looks at the use of Prolog for an automatic knowledge acquisition procedure for multi-agent conversation systems, argumentative ones more specifically. These systems need to be able to produce sensible structured arguments and counter-arguments, and as such, the quality of the knowledge base determines the performance of the system. Moreover, as with any knowledge representation task, manually retrieving and organising knowledge is a tedious and difficult problem. The authors propose using two ontologies, a domain ontology and argument ontology. The domain ontology describes the related concepts of the argument domain and the argument ontology consists of the constructed arguments. Prolog fits into this equation as the middle man. It translates the domain ontology into a set of structured arguments. It does this by using its recursive features and rules to backtrack to all the possible arguments within the argument conclusion (provided by the domain ontology). The result is a tree structure of arguments which conform to the argument ontology structure, thus being easily converted to the OWL format.

While Prolog accounted for a small portion of the proposed system, it vastly simplified the knowledge representation task due to its in-built inference mechanisms. Some limitations were found with regards to the inability of Prolog translating certain axioms from the domain ontology into clauses. Albeit, the implementation validated the research by successfully increasing the practicality of argumentative agents and providing a novel approach to the knowledge representation problem.

Lastly, the third paper [3] proposes a model for reasoning with words by obtaining their semantic relationships from WordNet and incorporating this into the inference mechanisms of Bousi-Prolog. The aim is to make the process of reasoning with words automatic by representing the knowledge of the lexical relationships (synonyms, antonyms, hypernyn, hyponym) of such words in hindsight, avoiding the need to define them explicitly. The focus of approach is on the meaning of individual words, and not the meaning of a sentence as a whole. By introducing this model, there is an increased ability to deal with syntactic vagueness of text expressed in natural language, resembling the way our brain interprets language. However, the model does not take full context into consideration. This could be seen as a shortcoming of the research, but the main focus seemed to be on producing a system with limited scope initially and providing plenty of room for future extensions such as embedding context.

3 Question Answering

Question answering or Q/A systems is a vast topic within NLP that deals with systems which can answer questions posed by humans. Typically, these systems perform a series of analyses on the input question and fetch an appropriate response from either structured data in a knowledge base or from unstructured sources using natural language techniques. Depending on their aim, they can operate within open or closed domains. This process can be eased by the use of ontologies of related information, such as those seen in [2] [4].

In recent years, one of the biggest breakthroughs in the field of NLP was when IBM's Watson beat two former world champions at *Jeopardy!*, a question answering game covering a broad domain of knowledge. The questions in the game require complex human level reasoning for correct interpretation and to respond with the correct answer. To face this challenge, the DeepQA [5] framework was designed by IBM researchers. The framework consists of many components that use a series NLP techniques to generate a series of possible answers, evaluate these hypothesis using information gathered from large unstructured natural text sources and choose the answer which is most likely correct [7].

While DeepQA consists of a pipeline of components utilizing different rule-based and statiscal techniques which lay outside the scope of this report, the rule-based portion of the question analysis module in DeepQA is built using Prolog [6]. Question analysis in Watson is concerned with finding 4 crucial elements in a question, these are: the focus, lexical answer types (), question classification and QSection [6]. While Prolog isn't used to determine all of these elements, it is good at establishing relationships between features in the text, which in turn serve as very fast way of asserting truths through the use of backtracking. To demonstrate this, the example used in [6] uses the question *"He worked as a bank clerk in Yukon before he published 'Songs of Sourdough'"* as a means of explaining how Prolog tackles this. Initially, the sentence is parsed to identify key words such as the verb "publish", subject "he" and objects "Songs of Sourdough'". Using Prolog, rules can be created to establish relationships between the elements of the question (connection between "he" the author and "Songs of Sourdough" the song/composition) to produce facts which could be later consulted when determining the focus of the question. In this case the focus is finding out the name of the author. By establishing this fact and determining the class of the question, Prolog and its backtracking mechanisms can get to work if by chance Watson comes across a piece of text that refers to *"Songs of Sourdough"* and a particular entity/subject in its vast domain knowledge relating to the question class. Watson could then easily assert that the entity it found is indeed the author of *"Songs of Sourdough"*.

Prolog is extremely suited for the question analysis tasks in Watson, as it is simple, very flexible (favouring future expansions to fit other domains), and is well suited for representing large numbers of rules that indicate patterns/relationships between the text in a question. The other notable advantage to the use of Prolog was its speed as it satisfies the constraint proposed by the

Jeopardy! challenge of having to analyse a question and produce a response in minimal time. Watson is able to analyse questions in a fraction of a second due to the Prolog implementation [6] allowing it to compete against the very best humans.

Another, albeit smaller scale example of a question answering system that harnesses the power of Prolog's expressiveness is RASM (Reading Answering System Model) [8]. Using Definite clause grammar (DCG), rules are expressed to represent the semantics of Vietnamese news titles in order to answer several types of questions. The system succeeded in successfully answering 6 types of questions, however, the questions were somewhat related to the news titles indicating that the system is aimed to operate in a closed domain.

RASM could be seen as a small segment of one of the components seen in the Watson system, i.e parser. This doesn't contradict the purpose of the research because RASM is dealing with a much smaller knowledge base of only a few newspaper article titles and tested against a few closed domain questions. Whereas Watson uses a vast collection of ontologies and is aimed at answering a large array of open domain questions which feature a much larger degree of vagueness and ambiguity, thus requiring a much more complex architecture consisting of a mixture of rule-based and statistical techniques. While is isn't fair to contrast RASM with Watson due to clear differences in research funding, it is evident that both approaches favour Prolog use for dealing with natural language processing tasks.

4 Natural Language Understanding

Natural language understanding or NLU for short, is the most difficult field in NLP which deals with the machine capabilities of reading, processing and understanding of meaning in natural language text input, synonymous with reading comprehension. In humans, mastering this skill requires extensive practice through the development of many other interrelated traits such as inferencing, vocabulary size and reading strategy [**?**]. As such, this field could be seen as the precursor required for the work of all the domains explored above, thus generating substantial academic and commercial interest.

This section focuses on reviewing systems that have the sole aim of understanding language. While it is true that the systems in the above sections have to perform the techniques mentioned in this section, the domains contain different problems which were the main focus of those sections.

The first paper under review is [9] which proposes a grammar checking system to help english students learn Persian grammar by using a feedback mechanism to alert students to mistakes rather than discarding the text entirely. This differs from common parsers in which the goal is to pre-process the text which of many things, aims to strip unnecessary or incorrect data. Essentially the system's goal is to understand the meaning of the correct Persian sentence intended by the student to be able to pick out where the flaws are. To do this, the authors built a test for the students containing a set of English sentences to be

translated to Persian. With this, the aimed to generate structures representing the syntactic and semantic meaning of both the English and Persian sentences. Then using SICStus Prolog, rules describing the relationships between the two languages were created in order to make a comparison between the generated structures and conclude if there are any anomalies with the meaning of the translated sentence. The results are then reported back to the student in the form of informative errors.

The conclusion summarises that the system performed reasonably well and with some more work it could have a positive impact on education, however, the dynamic nature of natural language requires systems to have a very large number of rules to model all possible occurrences of utterance. As expressed by the authors, the rules required an enormous amount of time and effort even though the system could be seen as conforming to a closed-domain of translations, suggesting that scaling the system to a more open-domain would be impossible or otherwise infeasible to fully represent using hand-crafted rules.

Another example of grammar and language structure analysis is demonstrated in [10]. The authors present a novel approach to syntactic analysis of the structure of Arabic sentences using the Government and Binding theory (GB) which is an attempt at creating universal grammatical rules that apply to all languages. However, Arabic like many languages has unique language structure features such as flexible word order and high inflection [10] requiring the authors to build a system that could handle variations of sentence orders. The system was implemented using SICStus Prolog and tested on 500 sentences producing reasonably accurate syntax structures demonstrated using various figures.

The proposed system shows much resemblance to the approach seen in [9]. Both papers propose grammar parsers targeted at a limited domain of knowledge, and differ only in their approach to the analysis of the structure of the target languages. Also both leave a lot of room for further research and system expansion.

There are similarities between systems focused purely on the understanding of text such as the ones examining in this section, and systems examined in the question answering, such as Watson. All language processing systems need to have some form of parsing at its core to break down the grammatical structure of language, the difference between these two systems is their goal. While Watson applies a series of parsing techniques to find key elements related to its goal of correctly generating the right answer to a question, these parsers are mainly aimed at understanding language structure as means of providing less complex software functionality such as grammar checking. The combination of DeepQA's techniques also allows Watson to somewhat mask the problem of scaling the system to a more open-domain of knowledge, something which is not seen in the above systems.

5 Conclusion

In this report, a series of recent papers were reviewed which provided some indications of the state of the art of Prolog, particularly in the field of natural language processing. The papers reviewed showed that there is still significant use of Prolog in solving many problems within the sectors of NLP. As reported in the above sections, the approaches that used Prolog have been demonstrated to be extremely successful, regardless of using Pure Prolog or a conjunction of techniques in the said approach. The majority of the approaches concluded that using Prolog, made the systems more flexible, allowed for the natural expression of rules [6] [3] [8], greatly facilitated the process of structuring knowledge [4] [2] and language understanding [9, 10, **?**].

With that being said, one of the main obstacles in researching the state of the art of Prolog, was that most of the research in NLP was carried out well over 10 years ago. It seems that as the years have progressed, pure Prolog approaches to NLP problems have become less common. While this isn't inherently a problem with Prolog, it suggests that it might not be adequate for solving all the problems proposed by natural language, but more realistically, it is merely another tool which is extremely suited for solving particular problems as part of the bigger picture. Watson and DeepQA [5] [6] being a very good example of this.

Another reason for this decay in use is due to the rise of other techniques within fields such as neural networks. Recently, with the advent of Deep learning and access to huge amounts of data, statistical methods are gaining traction and have achieved state of art results in many AI fields including NLP [11]. More specifically, Deep Recurrent Neural Networks such as LSTMs (Long short-term memory) and GRU (Gated Recurring Unit) have been shown to be extremely effective at processing natural language and at tasks such as sentiment analysis and conversation agents [11]. These methods unlike Prolog, don't rely on hand-crafted rules, provided that they have access to enough data to cover the domain of interest. Given data, they could possibly deal with the vagueness of language better by eliminating the process of manual rule creation, and in turn prioritizing the pre-processing of textual datasets. However, their black-box nature could also prove to be problem with regards to debugging and understanding how the systems are reaching certain conclusions which isn't a problem with Prolog due to its expressive and flexible nature.

To conclude, in Appendix A is a table showing the key papers reviewed and a comparison of their publishing dates, the way in which they used Prolog in their approach (Pure, Extended...), application domain, and the NLP sector in which the research contributes to. It summaries most of the recent research using Prolog as part of their approaches to problems and shows that while Prolog isn't the main player in the field anymore, it still contributes substantial value to research. It could be seen that Prolog's extensibility and the various adaptations it has undergone, is the reason it has persevered so long as being one of the go to AI programming languages, and surely opens up the possibility for further use in the future, albeit fragmented.

References

[1] Colmerauer, A. and Roussel, P. (1993). *The birth of Prolog.* ACM SIGPLAN Notices, 28(3), pp.37-52

[2] Martino, B.D., Esposito, A., DAngelo, S., Marrazzo, A., and Capasso, A. (2016). *Automatic Production of an Ontology with NLP: Comparison between a Prolog Based Approach and a Cloud Approach Based on Bluemix Watson Service.* In Proceedings - 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2016, (Institute of Electrical and Electronics Engineers Inc.), pp. 537542.

[3] Rubio-Manzano, C., and Julin-Iranzo, P. (2014). *Reasoning with Words: A First Approximation.* In IEEE International Conference on Fuzzy Systems, (Institute of Electrical and Electronics Engineers Inc.), pp. 569574.

[4] Liu, B., Yao, L., Liu, F. (2017). *Ontology-based argument acquisition for argumentative agent.* In 2017 3rd International Conference on Information Management, ICIM 2017, (Institute of Electrical and Electronics Engineers Inc.), pp. 399-406.

[5] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. a., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., et al. (2010). *Building Watson: An Overview of the DeepQA Project.* AI Magazine 31, 5979.

[6] Lally, A., Prager, J.M., McCord, M.C., Boguraev, B.K., Patwardhan, S., Fan, J., Fodor, P., and Chu-Carroll, J. (2012). *Question analysis: How Watson reads a clue.* IBM Journal of Research and Development 56, 2:1-2:14.

[7] Lally, A., and Fodor, P. (2011). *Natural language processing with Prolog in the IBM Watson system.* The Association for Logic Programming (ALP) 14.

[8] Pham, S.T., and Nguyen, D.T. (2014). *Implementation method of answering engine for vietnamese questions in reading answering system model (RASM).* In Proceedings - Asia Modelling Symposium 2014: 8th Asia International Conference on Mathematical Modelling and Computer Simulation, AMS 2014, (Institute of Electrical and Electronics Engineers Inc.), pp. 175180.

[9] Mirzaeian, V.R., Kohzadi, H., and Azizmohammadi, F. (2016). *Learning Persian grammar with the aid of an intelligent feedback generator.* Engineering Applications of Artificial Intelligence 49, 167175.

[10] Moubaiddin, A., Tuffaha, A., Hammo, B., and Obeid, N. (2013). *Investigating the syntactic structure of Arabic sentences.* In 2013 1st International Conference on Communications, Signal Processing and Their Applications, ICCSPA 2013, p.

[11] Yin, W., Kann, K., Yu, M., and Schtze, H. (2017). *Comparative Study of CNN and RNN for Natural Language Processing.*

A
Key Papers Comparison Table

Paper	Date	System Name	Prolog Use	NLP Sector
Automatic Production of an Ontology with NLP: Comparison between a Prolog Based Approach and a Cloud Approach Based on Bluemix Watson Service	2017	No Name	Pure Prolog w/ OWL helper libraries	Knowledge Representation
Reasoning with Words: A First Approximation	2014	No Name	Bousi-Prolog	Knowledge Representation
Ontology-based argument acquisition for argumentative agent	2017	Argument Acquisition Based on Ontologies (AABO)	Pure Prolog w/ OWL helper libraries	Knowledge Representation
Question analysis: How Watson reads a clue.	2012	DeepQA	Prolog approach in conjunction with many other components	Question Answering
Implementation method of answering engine for vietnamese questions in reading answering system model (RASM).	2014	RASM	Pure Prolog	Question Answering
Investigating the syntactic structure of Arabic sentences.	2013	No Name	SICStus Prolog	Understanding
Learning Persian grammar with the aid of an intelligent feedback generator.	2016	No Name	SICStus Prolog	Understanding

2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems

Automatic production of an ontology with NLP: Comparison between a Prolog based approach and a Cloud approach based on Bluemix Watson service

Beniamino Di Martino*, Antonio Esposito† Salvatore D'Angelo‡, Alessandro Marrazzo§ and Angelo Capasso¶
Second University of Naples
Aversa, Italy
Email: beniamino.dimartino@unina2.it *, antonio.esposito@unina2.it †, salvatore.dangelo4@studenti.unina2.it ‡,
alessandro.marrazzo@studenti.unina2.it §, angelo.capasso@studenti.unina2.it¶

***Abstract*—Nowadays, most of the information available on the web is in Natural Language. Extracting such knowledge from Natural Language text is an essential work and a very remarkable research topic in the Semantic Web field. The logic programming language Prolog, based on the definite-clause formalism, is a useful tool for implementing a Natural Language Processing (NLP) systems. However, web-based services for NLP have also been developed recently, and they represent an important alternative to be considered. In this paper we present the comparison between two different approaches in NLP, for the automatic creation of an OWL ontology supporting the semantic annotation of text. The first one is a pure Prolog approach, based on grammar and logic analysis rules. The second one is based on Watson Relationship Extraction service of IBM Cloud platform Bluemix. We evaluate the two approaches in terms of performance, the quality of NLP result, OWL completeness and richness.**

I. Introduction

In the recent years, there has been an increasing discussion on Natural Language processing techniques. Here for Natural Language Processing (NLP) we intend the automatic understanding of natural language, driven by software and machine based tools, enabling computers to derive meaning from human or natural language input. To ensure that a machine is able to understand natural language, several techniques have been presented over the years. One of these techniques is represented by logic programming implemented with Prolog. Such a language allows to build a system in which basic knowledge, based on facts and rules describing the domain of interest, is asserted in a logic program. The programmer no longer has to describe the process, but she focuses its attention on the objects and the relationships that connect them. Recently, Cloud Computing has rapidly increased its diffusion in academic communities and business companies due to its characteristics and qualities. The cloud platform on which we focus in this paper, to create our Web application, is IBM Bluemix. This platform allows users to create applications directly on the Web and offers many services that can be included in them. For natural language processing IBM provides the Watson ecosystem with the purpose to have computers that can interact with humans. The goal of paper is to compare two different approaches: a local approach based on Prolog rules for the grammar analysis of text and another one based on Cloud, the latter implemented via the IBM Watson Service. Such methodologies are presented, evaluated and qualitatively compared.

The remainder of this paper is structured as follows: section II provides an insight of existing NLP techniques and tools; section III describes the two approaches we have exploited for our NLP analysis; section IV provides more detailed information on the Prolog based approach; in section V we instead describe the Cloud approach which exploits the IBM Bluemix services; section VI provides a comparison among the proposed approaches; finally, section VII closes the paper with some comments on current work and future developments.

II. State of the Art

Many studies and approaches have been proposed during the recent years for creating, populating and evaluating ontologies [1]. Information Extraction methods by the use of Natural Languages Processing techniques (NLP) have been already proposed [2], [3]. In [4] the authors provide an overview of ontology learning techniques, referring in particular to the (semi-)automatic construction of an ontology. According to the type of input used by the learning process, the proposed technique defines three different kinds of input: **structured data**(database schemes), **semi-structured data** and **unstructured data** (natural language text documents, like the majority of the HTML based web-pages). In literature there are many implementations of semi-automatic or automatic techniques for the creation of ontologies. The work presented in [5] defines and implements a technique to automatically derive ontologies from analysed text, which is based on hierarchical clustering of document corpora. **Text2Onto**, a framework for ontology learning from textual resources; this framework represents the learned knowledge at a meta-level in the form of instantiated modeling primitives within a so called Probabilistic Ontology Model (POM) [6]. **SHELDON** combines NLP machine learning and Semantic Web to provide semantic meaning to a given sentence [7]. The **OntoRich** framework is a support tool for semi-automatic ontology enrichment and evaluation. The WordNet vocabualry is used to extract

978-1-5090-0987-9/16 $31.00 © 2016 IEEE
DOI 10.1109/CISIS.2016.98

Question analysis: How Watson reads a clue

A. Lally
J. M. Prager
M. C. McCord
B. K. Boguraev
S. Patwardhan
J. Fan
P. Fodor
J. Chu-Carroll

The first stage of processing in the IBM Watson™ system is to perform a detailed analysis of the question in order to determine what it is asking for and how best to approach answering it. Question analysis uses Watson's parsing and semantic analysis capabilities: a deep Slot Grammar parser, a named entity recognizer, a co-reference resolution component, and a relation extraction component. We apply numerous detection rules and classifiers using features from this analysis to detect critical elements of the question, including: 1) the part of the question that is a reference to the answer (the focus); 2) terms in the question that indicate what type of entity is being asked for (lexical answer types); 3) a classification of the question into one or more of several broad types; and 4) elements of the question that play particular roles that may require special handling, for example, nested subquestions that must be separately answered. We describe how these elements are detected and evaluate the impact of accurate detection on our end-to-end question-answering system accuracy.

Introduction

The question-answering process in IBM Watson*, like that of most other question-answering systems, begins with a question analysis phase that attempts to determine what the question is asking for and how best to approach answering it. Broadly speaking, question analysis receives as input the unstructured text question and identifies syntactic and semantic elements of the question, which are encoded as structured information that is later used by the other components of Watson. Nearly all of Watson's components depend in some way on the information produced by question analysis.

Question analysis is built on a foundation of general-purpose parsing and semantic analysis components. Although these components are largely domain-independent, some tuning to the special locutions of Jeopardy!** questions has been done, which we describe in this paper.

Based on this foundation, we apply numerous detection rules and classifiers to identify several critical elements of the question. There are a variety of such elements, each of which is vital to different parts of Watson's processing. The most important elements are the *focus, lexical answer types (LATs), Question Classification*, and *Question Sections (QSections)*. The definitions of these terms refer to the following example Jeopardy! question:

> POETS & POETRY: He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.

The *focus* is the part of the question that is a reference to the answer. In the example above, the focus is "he". (In the case where multiple words refer to the answer, it is generally sufficient to detect one focus and then apply general-purpose co-reference resolution to find the other references.) The focus is used, for example, by algorithms that attempt to align the question with a potential supporting passage [1]; for proper alignment, the answer in the passage should align with the focus in the question.

LATs are terms in the question that indicate what type of entity is being asked for. The headword of the focus is generally a LAT, but questions often contain additional LATs, and in the Jeopardy! domain, categories are an additional source of LATs. In the example, LATs are "he", "clerk", and "poet". LATs are used by Watson's type coercion components [2] to determine whether a candidate answer is an instance of the answer types.

Digital Object Identifier: 10.1147/JRD.2012.2184637

0018-8646/12/$5.00 © 2012 IBM