# Simple and Effective Question Processing using Regular Expressions and WordNet

Chris Whidden

August 2005

**Abstract**

Natural Language Processing is a computationally expensive and complex area of Computer Science. Question Answering in particular is known for having time and resources thrown at it, to achieve a system that is suitable only for a limited domain. Is this really necessary? Simplicity and efficiency are just as important as accuracy. A fast system is easier to test, debug and experiment with; it is also more attractive to an end user. This report details attempts to improve upon the question processing component of the Dalhousie University Jellyfish question answering system.

To achieve speed as well as efficiency, our system utilizes a part-of-speech tagger instead of a full natural language parser. First, Regular expressions distinguish the syntactic structure, then entities and relations in a question are found and processed, and finally the question is assigned a category with the help of WordNet. Our approach is based on the syntactic structure of a question and is independent of any particular word groupings.

Some methods for dealing with reversed questions, questions about an attribute of an entity, and question about an entity type are discussed, as well as ideas for future research.

The results of the system are promising. While requiring less than ten seconds to process a set of 351 questions, an accuracy of 88% is achieved.

A simple and effective system was created for the processing of natural language questions. It is an improvement upon our earlier system that maintains the speed and is not overly complex. While it may not be as powerful as those used by top question answering systems, it works well and is suitable for further research.

# Contents

# 1 Introduction

Question Answering is a field of Natural Language Processing that attempts to answer questions stated in a natural language. Stimulated by the Text REtreival Conference (TREC) QA track, Question Answering systems are constantly improving [18]. TREC question answering is about giving a complete and concise answer to a question in a natural language. It also currently involves a 'target' that the question is related to.

Open-Domain Question Answering aims to extract concise, complete answers from large document collections to questions written in a natural language, such as English. A key phase of answering a question is understanding it — Question Processing. Question Processing is about converting a natural language question into an unambiguous form that a computer is capable of understanding. Properly understanding a question is vital to question answering and an interesting topic to Artificial Intelligence as a whole.

The Dalhousie University NLP group has a Question Answering system named Jellyfish. This is our second year of participating in the TREC QA track, and it has brought many improvements [6].

Our question processing system has been completely reworked for this year. It's goal is to improve upon the earlier system while keeping the same general idea. The DalTREC Question Answering system is built for speed and simplicity so the Question Processing system must be fast, easy to use and update, and still correctly process the varied ways of asking a question.

The main idea of the system involves finding the syntactic structure of a question. Questions with the same syntactic structure can be treated the same and their differences handled by later processing. These similar questions are processed with the same regular expression and their syntactic structure determines their properties.

Another addition to the system is further processing of the entities and relations. Adjectives and extra information are separated from the main part of entities to allow simpler searching and provide additional data.

An essential concept in Question Processing is the question category [1, 4]. This describes the kind of question and often the expected answer type. Some examples are *CITY*, *PERSON*, and *TRANSLATE*. The previous method of specific regular expressions had a category for each expression, but in a more general system another method is necessary. The system uses WordNet, a lexical reference system that is currently very popular in Question Answering [8].

Some suggestions gleaned from working on this system are adding more semantic information, better processing of entities and relations and finding uses of such, and dynamically creating categories based on the expected answer type.

This system has been much improved over the previous one. It provides a way to process difficult questions, and is extensible. A greatly increased amount of information is now available about a question, but is not currently used by Jellyfish. Uses of this information need to be researched and put into effect.

# 2 Related Work

## 2.1 Other Systems

Question Answering has been developing a lot of interest as the required technology and processing power becomes feasible. The Text REtrieval Conference (TREC) has been a major impetus in Question Answering research with their QA track.

Many modern question answering systems have similarities in their basic structure, especially in their question processing component. Because of this, Question Processing is not often given much focus in the literature. The following three systems use typical question processing techniques.

The University of Wales Bangor question answering system, QITEKAT [2], uses a part-of-speech tagger and regular expressions, similar to our system. It, however, automatically generates those expressions. Although this is potentially better than manually creating the expressions, it takes much more time, effort, and people to develop such a generating system.

The Fudan University question answering system, FDUQA [19], uses linkparser in place of a tagger and regular expressions. They also use WordNet to generate the question category. Using linkparser is a novel idea but we rejected it for the same reason that our system does not use a chunker: regular expressions provide a higher degree of control and error checking. Linkparser and chunkers are simply not designed for questions.

The University of Singapore question answering system [3] uses the MiniPar parser and the syntactic relations it provides. This is a good method, but in the interests of speed and simplicity, we chose to use tagging not full parsing.

## 2.2 Previous Question Processing

Previously, the DalTREC Question Answering system, Jellyfish, used a simple, but effective, method to process and understand questions: regular expressions of the most common question formats [17]. An expression such as */∧When $did (.*) ($REverb) $/* will correctly match and process many simple questions, and is quick and easy to create.

Some patterns are very general, such as the above. That generality can cause incorrect matches. Questions such as *"When did Bob stop the bandit attack?"* will be improperly matched. On the other hand, some questions are very specific, such as */∧How (cold|warm|hot) is(.∗)/*. It is easy to see that synonyms of cold and hot will not be matched as they should.

To overcome these issues, more and better patterns must be created. Although it is initially simple to create a wide variety of useful patterns, the task quickly grows more complex. Patterns that are too general match excessively and patterns that are too specific do not match enough. The wide variety of available words and ways of phrasing a question explode the necessary number of expressions. One solution to this problem would use machine learning techniques to generate use full patterns. Another solution is to use a different method.

I opted for the latter approach but wanted to retain the spirit of the previous method. My goal was to use semantic information and regular expressions to determine the syntactic structure of the question. The questions *"How cold is Antarctica?"* and *"How old is George Lucas?"* are very different but have the same syntactic structure. There should be one expression for these and similar questions despite multiple words and prepositional phrases. The question *"How fast in miles per hour is the yellow plane with the purple stripe?"* must also be matched as well as stranger and unexpected questions.

### 2.3 Tools

**OpenNLP** The part-of-speech tagger used is from the OpenNLP project [10], and is open source. It is used as is without any additional training or alterations. Each word in a question is tagged as verb, noun, etc, using the Penn Treebank set of tags [15]. This semantic information forms the basis of our approach.

The tagger works well in most cases but not for words that are missing from its included dictionary. Our system encountered a few cases of words such as *Venus* which were not in the dictionary and were improperly tagged. Extending the dictionary or possibly using WordNet as a dictionary might improve this.

**WordNet** The previous method of specific regular expressions had a category for each expression, but in a more general system another method is necessary for questions where the syntactic structure alone does not determine the category. The system uses WordNet, a lexical reference system that is currently very popular in Question Answering to find the category for *'how'* questions about an attribute and *'what/which'* questions about an entity [8, 11].

"WordNet is an online lexical reference system whose design is inspired

> by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets." [16]

WordNet has a command line tool, but to provide better integration, our system uses the Perl module WordNet::QueryData [14] to access WordNet. The Perl Modules WordNet::Similarity [13] and WordNet::SenseRelate::WordToSet [12] are used for word sense disambiguation.

WordNet provides a lot of semantic relations and word sense information and is constantly expanded and improved. WordNet is very useful for classifying question categories. One problem with WordNet is that there is so much information that it can be difficult to select the material that is relevant to a question. The large number of senses are useful but can be difficult to disambiguate. On the whole, WordNet is a great resource for natural language processing.

# 3 Distinguishing the Question Form

## 3.1 Part-of-Speech Tagging

The DalTREC approach to Question Answering is based on shallow parsing and regular expressions and is intended to be quick and uncomplex. In accordance with this goal, a part-of-speech tagger is used on the questions instead of a parser. Full parsing and processing may not be necessary to understand a question.

## 3.2 Basic Framework

To easily create new patterns that capture the syntactic structure of a question, a basic framework is useful. Regular expressions represent entities, relations, adjectives, and context. These were made for the tagged text and thus are independent of any particular words.

The context expression is for prepositional phrases. These are seperate from the syntactic structure of a question, but often contain important information that is necessary to understand the question. Units, extra description and other linguistic information can be found in prepositional phrases but are difficult to match without part-of-speech tags or other information.

Take the earlier question *"How fast in miles per hour is the yellow plane with the purple stripe?"* as an example. The syntactic structure of this question is
'How *adjective context* is *entity context*.'
Based on the syntactic structure, this question asks about an attribute of the

yellow plane. This does not seem very different from the previous tag-less method, but for questions like *"What Nobel laureate was expelled from the Philippines before the conference on East Timor?"* or *"Who were the architects who designed the Empire State Building?"* this method is much easier.

After examining more than 2000 questions from early TREC competitions [18] I created expressions to handle the syntactic structure of almost any question. This is not a completely new idea, as it is similar to the methods of some QA systems [4].

### 3.3 Reverse Questions

Although most questions are asked with the Wh-word at the beginning, as in *"How cold is Antarctica?"* some questions are reversed. A 'reverse question' does not have the Wh-word at its beginning and is equivalent to a question that does. A reverse question such as *"Winnie the Pooh is what kind of animal?"* will not be matched by any of the ordinary expressions. The solution to this problem lies in the fact that this question is equivalent to *"What kind of animal is Winnie the Pooh?"* To allow reverse questions the system converts from the first type to the second. In most cases the *'W'* in *'Winnie'* would be changed to lowercase but this is not done for proper nouns.

## 4 Entity and Relation Processing

Although entities can be simply described in a question, they often are not. Instead of *'president'* the more descriptive *'youngest president'* or *'youngest president of the United States'* are used to provide extra information. This extra information must be correctly processed in order to accurately understand the question.

The head word of an entity serves as a useful answer type. To find the head word, simply remove any adjectives and prepositional phrases. If multiple words are left, check for compound words. I there are still multiple words, the last word should be the head.

The adjectives and prepositional phrases should be stored in a list and themselves processed. The information that they provide should be integrated into the understanding of the question. Information like *'youngest'*, *'most'*, and *'Canadian'* requires some form of logical understanding in the later sections of a QA system, so we have not fully implemented this step.

The relations in a question must be similarly processed. The differences between *'given'*, *'given by'*, and *'given to'* are important. Treating them as the same can lead to problems and is clearly not fully understanding the question.

# 5 Question Categorization

## 5.1 Question Category Basics

An important part of many modern question processing systems is the concept of question category [1, 4]. This describes the kind of question and often the expected answer type. Categories can be simple such as *PERSON* or *CITY*, or complicated such as *TRANSLATE* or *SYNONYM*. Categories determine how the question should be handled by a Question Answering system.

The category of many questions can be determined from the Wh-word. *'Who'* questions have a category of *PERSON*, *'where'* questions have a category of *LOCATION*, *'when'* questions have a category of *DATE*, and *'why'* questions have a category of *WHY*.

It is more difficult to determine the question category of questions with *'what,' 'which'* or *'how'* as the Wh-words. For *'what'* and *'which'* questions like *"What American commodore demanded that Japan trade with the United States?"* the expected answer type is the question category. For this question, the category should be *PERSON|COMMODORE*, *COMMODORE*, or simply *PERSON*, depending upon the capabilities of the QA system.

Questions like *"What is AmeriCorps?"* are *DEFINITION* questions, but questions like *"What is the brightest star visible from Earth?"* are the same as the above. The syntactic structure of these questions easily distinguishes them, but a regular expression without any semantic knowledge would have trouble seeing a difference between the two.

Question such as *"What did Alfred Noble invent?"* are difficult to categorize but should be categorized as the most distinct class of entities that the relation can occur on; in this case, *INVENTION*.

For *'how'* questions like *"How did Anne Frank die?"*, the question category should be *CAUSE|DEATH*, *CAUSE*, *HOWDIE*, or something similar.

*'How'* questions that ask for the value of an attribute of something, such as *"How cold is the moon?"* should be categorized as the noun that the adjective is an attribute of; in this case, *TEMPERATURE*.

*'How many'* questions such as *"How many islands does Fiji have?"* are categorized as *COUNT|ISLANDS* or *COUNT* with a variable of what to count.

Some statements can also be interpreted as questions. Statements such as *"Name a female figure skater."* and *"List 10 U.S. cities that have an opera house."* are treated the same as *'what'* questions.

There are some types of questions that must be specially handled. *"How do you say X"* describes a *TRANSLATION* question, and there are probably other less common questions that are similarly specialized.

### 5.2 Categorizing an Answer Type

For 'what' and 'which' questions, the category is based on the expected answer type. The system maintains a list of valid categories. We use the categories suggested by Rada Mihalcea [7].

First, the expected answer type is processed as mentioned previously. Then the system checks if the head word of the expected answer type, and any synonyms of it, is a category. If not, the hypernyms of the head word, for each sense of the head word, are generated with WordNet and the first one that is found in the category list is the question category.

The senses of the head word can either be sorted by frequency or sense disambiguated, but sense disambiguation is only partially implemented and much slower.

### 5.3 Attribute Questions

For many adjectives, WordNet has a relation showing what noun they are a value of. Using this it is a simple matter to determine *'HEIGHT'* from *'How high'* and *'SIZE'* from *'How big'*. There may arise ambiguity, however, for *'DISTANCE'* or *'DURATION'* could be the category for a *'How long'* question. For those, smarter processing is needed to achieve correct results.

## 6 Results

Examining the results of the system when processing the TREC 2004 question set displays the value of this approach for Question Answering. This required less than 10 seconds, 7 of which were Wordnet loading. Of the 351 questions, 42 have some problem. This gives an accuracy of 88%. Of those 42, 11 had tagging errors, 30 had the wrong category or no category, 16 had errors in the question relation, and 12 had errors with the entities.

A few minor mistakes made by the system could be avoided with suitable training, as the tagger does mislabel some words. For example: *'comet'* in *'Hale Bopp comet'* is considered a verb because it is not in the tagger's dictionary.

The large number of category errors shows that more work is needed on that section. 19 of the 42 questions were correctly processed but not given a question category, and 4 of them were given a category based on an incorrect sense of the answer type.

An example of this is the question *"What is their average life span?"* asked about agoutis. Instead of the category *'AGE'*, span is interpreted as a bridge and the category given is *'BRIDGE'*. Similarly, The Question *"Who is the Queen of*

*England?"* was given the category *'ANIMAL'* in an earlier test, because *'Queen'* was interpreted as *'Queen bee'*. One problem with using WordNet is the wide variety of senses available and the speed of reliable sense disambiguation on questions, which tend to have little context in them.

Some other problems were caused by questions that had *'and'* or *'or'* in them, such as *"What does the name mean or come from?"* This was simply not handled well by the current expressions which do not take them into account.

# 7 Future Work

Some areas of research that it would be useful to investigate are adding more semantic information, better processing of entities and relations and finding uses of such, and dynamically creating categories based on the expected answer type.

We only used a part-of-speech tagger to provide semantic information about the questions. It would be interesting to see if a full parser would provide a larger amount or more accurate information. Tagging is simple and quick, but works very well. It may be that a parser would add needless overhead when a tagger is all that is necessary. It might also be interesting to create a 'question parser' that is specifically tailored for questions and is similar in principle to how we handle questions.

Our system does very simple and limited processing on entities and relations. It also does not use all of the information that is processed. In order to correctly handle questions in an intelligent manner, this area needs further research and exploration.

Although it works well, our method for categorizing 'what' questions could be improved. The expected answer type itself should the category and the generated hypernym category the category type. Our system does not currently do this because it relies on specific categories that describe what an answer looks like. For many answer types, it should be possible to dynamically generate a list of possible answers using WordNet hyponyms. When fully implemented, this would greatly increase the utility and robustness of the system.

# 8 Conclusion

Question Processing is about discovering the meaning of questions. The Question Processing component of a QA system is responsible for converting a natural language question into a machine understandable and unambiguous form. Properly understanding a question is vital to question answering and an interesting topic to Artificial Intelligence as a whole.

The Dalhousie University NLP group has a Question Answering system named Jellyfish. This is our second year of participating in the TREC QA track, and it has brought many improvements. We have developed a new question processing system which attempts to retain the speed and simplicity of the previous system, while increasing the robustness, accuracy and precision.

The main idea of the new system is finding the syntactic structure of a question. Questions with the same syntactic structure can be treated the same and their differences handled by later processing. These similar questions are processed with the same regular expression and their syntactic structure determines their properties.

Other additions to the system include further processing of the entities and relations, and question categorization using WordNet. Further exploration needs to be done into using semantic information, using the information found when processing entities and relations, and dynamically generating categories from answer types.

Despite the speed and simplicity of the system, the results are on par with similar systems. There is currently an 88% accuracy rate and most problems are caused by improper tagging, wrong word senses, and strangely phrased questions.

This system has been much improved over the previous one. It provides a way to process difficult questions, and is extensible. A greatly increased amount of information is now available about a question, but is not currently used by Jellyfish. Uses of this information need to be researched and put into effect.

## 9 Acknowledgements

This work was supported by an NSERC USRA

## References

[1] David Alm, Valentin Jijkoun, and Stefan Schlobach. *Annotation guidelines for question classification*. Available online at `http://ilps.science.uva.nl/Resources/AnswerTypeChecking/question-type-annotation-guidelines.pdf`, June 8, 2004.

[2] Terence Clifton and William Teahan. Bangor at trec 2004: Question answering track. In *The Thirteenth Text REtrieval Conference (TREC 2004) Proceedings*, Gaithersburg, Maryland, USA, November 2004.

[3] Hang Cui, Keya Li, Renxu Sun, Tat-Seng Chua, and Min-Yen Kan. National university of singapore at the trec-13 question answering main task. In *The Thirteenth Text REtrieval Conference (TREC 2004) Proceedings*, Gaithersburg, Maryland, USA, November 2004.

[4] In-su Kang, Jong-hyeok Lee, Sang-yool Lee, and Seung-hoon Na. Question answering approach using a wordnet-based answer type taxonomy. In *Proceedings of the TREC 2002 Conference*, pages 512–519, February 2002.

[5] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, January 2000.

[6] Vlado Keselj. *DalTREC - Dalhousie TREC Project*. Dalhousie University, `http://flame.cs.dal.ca/~trecacct/`, 2005.

[7] Rada Mihalcea. *QA Data Set: Annotated questions*. University of North Texas, Available online at `http://www.cs.unt.edu/~rada/downloads.html#qa`, 2005.

[8] George A. Milleri, Richard Beckwith, Christiane Fellbaum, Derek Gross, K. Miller, and Randee Tengi. *Five papers on WordNet*. Available online at `ftp://ftp.cogsci.princeton.edu/pub/wordnet/5papers.pdf`, August 1993.

[9] Dan I. Moldovan, Marius Pasca, Sanda M. Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering

system. *ACM Transactions on Information Systems (TOIS)*, 21(2):133–154, April 2003.

[10] openNLP. *openNLP tools*. `http://opennlp.sourceforge.net`, 2005.

[11] Marius Pasca and Sanda M. Harabagiu. The informative role of wordnet in open-domain question answering. In *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh, June 2001.

[12] Ted Pedersen. *WordNet::SenseRelate::WordToSet*. `http://search.cpan.org/~tpederse/WordNet-SenseRelate-WordToSet-0.02/lib/WordNet/SenseRelate/WordToSet.pm`, 2005.

[13] Ted Pedersen. *WordNet::Similarity*. `http://search.cpan.org/~tpederse/WordNet-Similarity-0.15/Similarity.pm`, 2005.

[14] Jason Rennie. *WordNet::QueryData*. `http://search.cpan.org/~jrennie/WordNet-QueryData-1.38/QueryData.pm`, 2005.

[15] Beatrice Santorini. *Part-of-Speech Tagging Guidelines for the Penn Treebank Project*. University of Pennsylvania, School of Engineering and Applied Science, Dept. of Computer and Information Science, Philadelphia, 1990.

[16] Princeton University. *WordNet*. `http://wordnet.princeton.edu/`, 2005.

[17] Anthony Cox. Vlado Keselj. Daltrec 2004: Question answering using regular expression rewriting. In *The Thirteenth Text REtrieval Conference (TREC 2004) Proceedings*, Gaithersburg, Maryland, USA, November 2004.

[18] E. M. Voorhees. Overview of the TREC 2003 question answering track. In *Text REtrieval Conference (TREC) TREC 2003 Proceedings*, pages 54–?? Department of Commerce, National Institute of Standards and Technology, 2003.

[19] Lide Wu, Xuanjing Huang, Lan You, Zhushuo Zhang, Xin Li, and Yaqian Zhou. Fduqa on trec2004 qa track. In *The Thirteenth Text REtrieval Conference (TREC 2004) Proceedings*, Gaithersburg, Maryland, USA, November 2004.