Clustering Semantically Similar and Related Questions

Deepa Paranjpe
deepap@stanford.edu

1 ABSTRACT

The success of online question answering communities that allow humans to answer questions posed by other humans has opened up a whole new set of search, browse and clustering problems. One of the important problems arises from the need to show similar and related questions for a particular "probe" question. The first exercise is to define a measure for similarity and relatedness that leads to showing interesting results to the user, assuming that the user is interested in the probe question. However, this exercise reveals the inherent ambiguity of defining relatedness for this problem. The solution proposed takes this ambiguity into account and does not rely on a fixed similarity measure to filter questions and re-rank them. Instead the approach uses a two step method to show relevant questions for the probe. The first step involves identifying the main topic of the question using which the base set of questions having this same topic is constructed. This base set is then clustered taking into account the lexical and semantic similarity between the questions. The hypothesis is that each of the identified clusters defines a type of relatedness with the probe – one of these types is the identity relation which encompasses paraphrases and very similar questions. For evaluating the proposed technique, the results were evaluated manually and it was noted that for 90% of the cases, the clustering technique is effective and the results displayed in this manner seem appealing.

2 INTRODUCTION

Online communities that allow you to ask and answer questions have recently become very popular. Unlike automated question answering systems (such as www.ask.com, www.brainboost.com), these online communities do not promise to answer your questions automatically. On the other hand, they provide a platform where other people in the community can answer your question and you can answer other questions posed on the forum. Yahoo Answers (www.answers.yahoo.com), Amazon's AskVille (www.askville.com) and www.blurit.com are some of the examples of such online Q&A services. The huge success of these systems can somewhat be attributed to the failure of large scale automated question answering systems. The growth of such services has been phenomenal (Yahoo Answers had 65 million answers and more than 7 million questions in November of 2006. In six months, the questions and answers have increased to more than 15 million questions and 140 million answers). As a consequence, these services have created a huge repository of human generated question and answers. There's a lot of research happening in treating this "user generated content" differently than most documents on the web are treated. This leads to a different set of search, browse and presentation problems for this "web" of user generated content. Different models for querying, ranking of results, clustering and categorization are evolving for this form of data present in Q&A form. Since this user generated content is in natural language, it is but obvious that several NLP techniques would be useful for handling these problems.

These Q&A services might not wish to offer automated answers to question but they want to provide good presentation in terms of browsing, searching and grouping results. One of the features that most of these services wish to offer is to display similar and related questions for a particular "query" question. Unlike a normal text document, a question and answer page is more structured where the most useful portion describing the intent of the page is the question. Traditional document similarity measures use bag of words kind of models and approximate similarity between two documents to mean high overlap in the terms in those documents. However, in order to determine similarity between just questions, it becomes important to define a similarity measure that is targeted more at the sentence level.

In this project, I have played with several methods for displaying related and similar questions for a given question. The focus was not on identifying perfectly similar questions or paraphrases but to formulate

different ways of grouping questions that have the same aspect of relatedness with the given question. The solution that I propose and implement is effective for the problem that is defined below.

3 PROBLEM DEFINITION

The problem is defined as follows: Given a question **q**, identify questions in the corpus that are similar and related to this question and group/cluster them as per their "relatedness" to the question **q**. I decided to focus on this problem more than identifying only paraphrases for a question for the following reason. When a user is "visiting" a question **q** it is not so interesting to look at only paraphrases or identical questions but to look at things which could be "related" to **q**. The assumption is that the user might be visiting this Q&A page through a web search and by browsing the online community. Thus, if we cluster results which potentially have some relatedness with question **q**, the idea is that the clusters identified would correspond to one of the ways of defining "relatedness". Take an example of **q** being, *"How can I get rid of my fear of flying?"* We show clusters of questions which are "related" to this question – one of the criterion for relatedness being semantic identity or exact question match. Hence, one of the clusters formed would contain questions that are exact duplicates of the above question plus paraphrases of the form *"How can I overcome my fear of flying?"* Another cluster could define a relation which considers other fears such as that of height, depth and ways to overcome those. One could think of many such relations according to which the questions could be grouped. There are some constraints that are imposed in this method on the definitions of paraphrasing and relatedness. In my definition of paraphrasing, there needs to be at least one *lexically* overlapping concept between two questions qualified as paraphrases of each other.

3.1 Different notions of question similarity and relatedness

There is an inherent ambiguity in the term "relatedness" due to the several different ways to perceive information. In order to put some restrictions on our expectation from such a system that identifies and groups together related questions, I have determined some ways in which questions would be similar. These ways are enumerated below.

3.1.1 Lexical and semantic similarity

This kind of similarity occurs when words in the two questions hugely overlap and meanings of the two questions are similar. These questions are paraphrases of each other. *Eg., "How can I overcome my fear of flying?" and "How can I get rid of my fear of flying?"* Note that these two questions differ in word /phrase overcome and "get rid of". Out of context, these two word and phrase are not necessarily synonyms but in the context of these questions, the phrase/word are similar.

3.1.2 Lexical similarity but no semantic similarity

In this case, words in the questions hugely overlap but the information need is very different and hence they cannot be treated as semantically similar. Eg., *"Who killed Brutus ?"* and *"Who did Brutus kill?"* have a huge word overlap but no semantic similarity. However, these questions can be treated as related. One can imagine that if someone is interested in the first question, there is a high chance that he will click on the second question.

3.1.3 Semantic similarity but no lexical similarity

In this case, the word overlap is minimal but the questions could be looking for the same answer. Eg., *"What is the meaning of life ?"* and *"I wonder what God had in mind when he made a human being"*. It can be argued that these two questions are looking for similar answers but they have no lexical similarity.

3.1.4 Somewhat Relatedness

For many questions, the corpus does not contain other questions that have the exact information need. Consider the question, *"Where can i find a list of publicly traded hedge funds?"* The corpus does not have other questions which can fall in category 1 of similarity. However, there are questions such as *"Where can i find a comprehensive list of hedge funds in California?" "Where can i find more on the hedge fund run by the elliott group?", "Where can i find statistics on average hedge fund performance over the last 20 years?", "What is a good finance resource for hedge fund analysts ?"*. These questions can be termed as highly related to the original

question. We can think about even looser relatedness as in questions such as "*When do babies start teething?*" and "*When do babies start crawling?"*

All the above four ways can give us related questions for a particular question. My idea is to see to what extent use of structure and lexical information is useful to cluster questions as per their relatedness with the original question.

4 RELATED WORK

Identifying paraphrases has always been of interest to computational linguistics. However, there has been recent interest in the problem of detecting paraphrases for the purpose of better question and answering. Authors have tried to use machine translation-based methods to generate paraphrases of a question posed for automatic Q&A [1]. The motivation of generating an array of lexically and structurally distinct paraphrases is that some of these paraphrases may better match the processing capabilities of the underlying QA system than the original question and are thus more likely to produce correct answers. Most of these techniques such as the ones presented in [2], [5] have relied on the presence or generation of a monolingual aligned corpora useful for building translation models. The authors in [3] and [4] have assumed that similarity of answers implies semantic similarity of questions and using this assumption, they have generated the monolingual aligned corpora. Using this aligned corpus, they learn a machine translation model and using it for retrieving similar questions. In [6], the authors use the Encarta logs (user logs containing queries and documents that users clicked on for review) partitioning the questions into small clusters, within which a perceptron classier is used for identifying question paraphrases. My approach of presenting questions in a semantically meaningful manner comes closest to the approach in [6]. However, I have seen much related work that tries to solve the problem that I have tried to.

5 KEY TECHNIQUES

This section contains the description of each of the key techniques that formed a significant part of the overall system.

5.1 Identifying the key topic in the question

For querying a structured database with a suitable schema and a structured query language, information needs can be expressed precisely. We can view the question as a structured query with some noise due to the natural language which is used to generate it [7].

Guided by this framework, one can work backwards: given a question, we can discover structured fragments in it. Specifically, I extract those words which that could appear (almost) unchanged in an answer passage had we been doing Q&A. These words constitute the key topic of the question. For example, consider the question "*How old is Tom Cruise?"* Here the key topic of the question is "Tom Cruise" and the question is looking for the age of this topic. The answer need not contain the word age (in fact, age should get "transformed" into a number). However, the words "Tom Cruise" will appear unchanged in it. These words are extracted using a POS tagger. Proper nouns and nouns that appear after the verb or more specifically, the ones that are not the atype words (as explained in section 4.3) are treated as the main topic words. The adjectives (if present) acting as modifiers of these nouns are also used as the modifiers of the main topic.

5.2 Classification of the question type

The question type is an important attribute of a question, which usually indicates the category of its answer. Table 1 shows a widely accepted question type taxonomy in QA as given by the authors in [8].

abbreviation, explanation
animal, body, color, creative, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
definition, description, manner, reason
group, individual, title, human-description
city, country, mountain, other, state
code, count, date, distance, money, order, other, period, percent, speed, temperature, size, weight

Table 1: Question type taxonomy

Based on the observation that two questions with different question types can hardly be paraphrases, questions in the corpus are first classified into the basic 6 types of question types – abbreviation, entity, description, location, number, human (further differentiation is not made within each question type

as given the above table). A Naïve Bayes classifier is trained on an existing corpus of labeled questions. This trained classifier can then identify the question type of a question.

5.3 Using BLEU score with the probe question

It is required to know how lexically "close" the result question is, to the probe question. A variant of the BLEU (Bilingual Evaluation Understudy) metric is used for that purpose [9]. Although, the BLEU metric is used for evaluating the quality of text which has been translated from one natural language to another using machine translation, we use it to calculate the lexical similarity between questions. We use our variant of the BLEU score to determine the similarity between two questions as follows:

Let q_1 and q_2 be the two questions,

U_1 = {set of all unigrams of other than the stop words in q_1}

U_2 = {set of all unigrams of other than the stop words in q_2}

B_1 = {set of all bi-grams of not including the stop words in q_1}

B_2 = {set of all bi-grams of not including the stop words in q_2}

T_1 = {set of all tri-grams of not including the stop words in q_1}

T_2 = {set of all tri-grams of not including the stop words in q_2}

$$\text{Sim}(q_1,q_2)= \frac{1}{3}*\{\frac{|U1\bigcap U2|}{|U1\bigcup U2|}+\frac{|B1\bigcap B2|}{|B1\bigcup B2|}+\frac{|T1\bigcap T2|}{|T1\bigcup T2|}\}$$

5.4 Answer Type Detection from the question

Consider the question *"Which Indian musician is the world famous Sitar player?"* Some words in the question give us a clue about the answer. The answer passage need not necessarily contain these words but instead these words get transformed into the answer. We call these words as the atype clue of the question. For the above question, these words are *"musician"* with a modifier as *"Indian"*. The atype clue is one of the two most important fragments of the question – the other important fragment being the main topic words.

A question with *wh-words* such as *who, when, where, how many, how long* and so on have an implicit atype clue hidden in the wh-word itself. However, for questions with the wh-word as *what* and *which*, we need to do further processing to extract the atype clue which is not one of the standard atypes (such as *person* or *organization* for *who*, *time* for *when* and so on). It is possible to extract the atype clue from the question automatically for *what* and *which* type of questions looking at the parse structure of the question. Consider the parse of the above sentence using the Stanford Lexicalized Parser:

```
(ROOT
  (SBARQ
    (WHNP (WDT Which))
    (SQ
      (NP (JJ Indian) (NN musician))
      (VP (VBZ is)
        (NP (DT the)
          (ADJP (NN world) (JJ famous))
          (NNP Sitar) (NN player))))
    (. ?)))
```

We can use the following two rules to extract the atype (and its modifier which is the adjective attached to it) given the parse of the question:

- If the question contains a noun between the wh-word and the main verb, then this noun is the atype word and its adjectives are the modifiers of the atype clue. Using this rule for the above case, we get the atype word as *musician* and its modifier as *Indian*.
- If the wh-word is immediately followed by the main verb of the question, then the first noun(s) occurring after the main verb is the atype word. Any adjective for this NN is the modifier for the NN. For the example below, this rule applies and the atype word is *ethnicity*.

```
(ROOT
 (SBARQ
   (WHNP (WP What))
   (SQ (VBZ is)
     (NP
       (NP (DT the) (NN ethnicity))
       (PP (IN of)
         (NP (NNS people))))
     (VP (VBG living)
       (PP (IN in)
         (NP (NNP Cypress)))))
   (. ?)))
```

It appears that using only the POS tags is sufficient is to find out the atype word and its modifiers.

6 OVERALL APPROACH

Using the above key techniques, a complete system for identifying and clustering semantically related questions was created (As shown in Figure 1).

6.1 Creating the base set for clustering

In order to identify related questions for a given question, ideally one would have liked to consider every question in the corpus of questions. However, since typical Q&A archives would contain several millions of questions, it becomes practically infeasible to do so. The best way to handle this problem is to identify the "base" set of questions which are further consider for generating semantically similar clusters. This set of questions should have at least one overlapping topic with the

"probe" question. For this reason, we identify the key topic (as a set of words) of the probe question using the technique given in section 4.1. This key topic (a keyword query) is then used for querying an index over the questions in the corpus. The resultant set of questions as a result of the search using the keyword query is then used for further analysis.

6.2 Feature Engineering

The idea is to cluster the questions in the base set as per certain features discussed in the key techniques section. The features used for clustering are as follows:

1 The words in the question that are not stop words and not the main topic words as identified in the first step.

2 In order to bridge the "lexical chasm", I also used synonyms of the words obtained in step 1. The WordNet synonyms were used for this purpose.

3 The question type was identified using the pre-trained question type classifier. This feature helped in grouping the questions as per their structure. However, since the training data was from a completely different corpus and also limited, it was observed that the question classifier's output was reliable only for well-formed and questions sharing similar vocabulary as the training data (which is not very common for a question answering community).

4 For questions that contained either what or which as their wh-word, the atype word and modifiers were used as features. For many questions when the question classifier output wasn't reliable, identification of the atype was very useful. It a

5 For the the BLEU score with the probe question

Further, the length of the result question and also the search relevance (as returned by Lucene) are also used as features. In this feature space, the result set questions are then clustered using the K-means clustering algorithm. The best results are obtained using k=5.

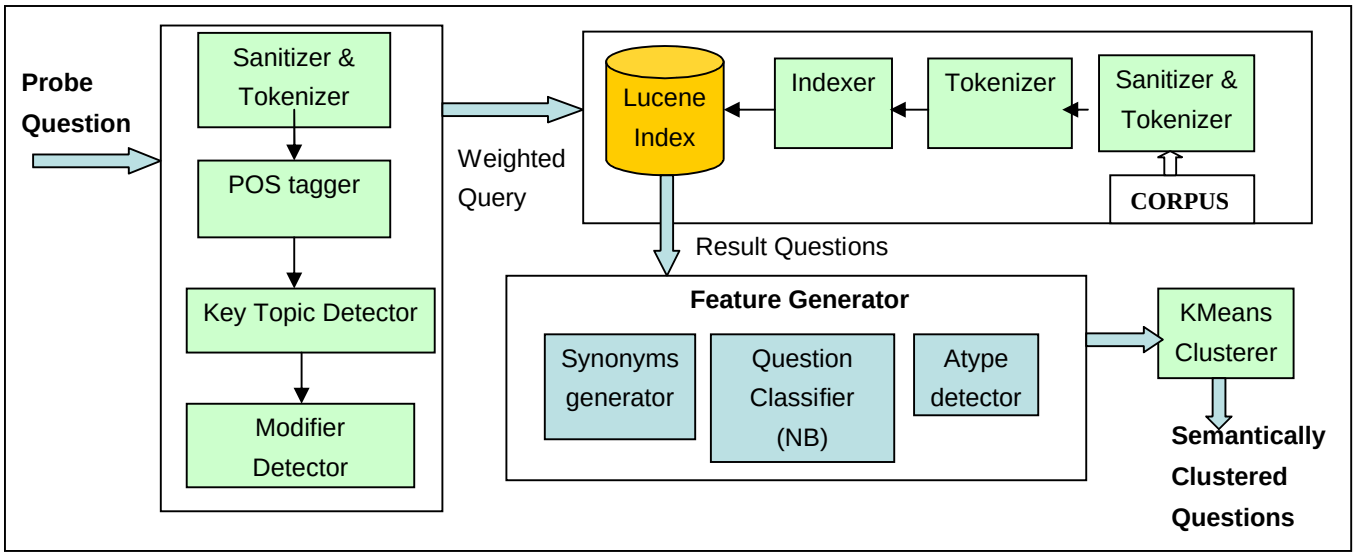

Figure 1: Overall System Architecture

7 EXPERIMENTAL SETUP

7.1 Data used for experiments

The data used for all the experiments was collected from the Yahoo Answers using their API [1]. I collected over a million questions to create the corpus of the questions (using 5000 queries per day to collect the question results). This data was cleaned, tokenized and indexed using the Lucene indexing system. The index is created only on the question title, although there is need to contain the text present in the question details since for many questions the title is very small and un-indicative of any meaningful content. Eg. there are many questions with title "Please help!" and the actual question is in the question details. However, I have ignored these cases for the time being.

Around 200 questions were picked at random as test questions and they were used for evaluating the effectiveness of the proposed technique.

Figure 2 shows the frequency distribution for the various question types. It can be seen that unlike the typical distribution that Q&A systems talk about, the actual distribution of the question types is pretty different. The question distribution is favorable more towards non-factoid based questions (look at the other category in the figure). Also, ambiguous questions types such as what, which are popular, so is the procedural category of how.

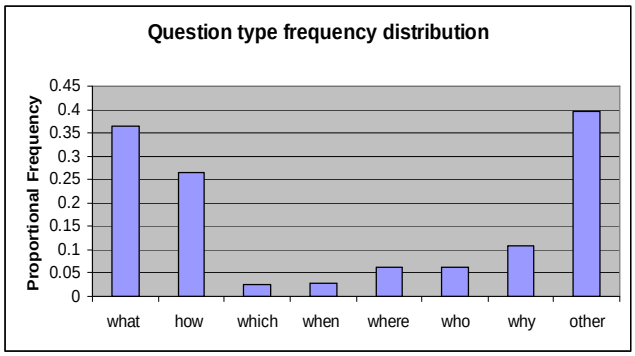

Figure 2: Frequency Distribution for the various question types {other category contains question word as is, can, do you}

7.2 Details

The main tasks are coded in Java while all conversion utilities (from one file format to another) are written in perl. The tasks in the entire system are divided into either online tasks or offline tasks. The following was done as part of the offline tasks:

1] A question classifier is constructed using the freely available training data available at http://l2r.cs.uiuc.edu/.cogcomp/Data/QA/QC/. This is present in the data directory. The model file of the trained classifier is present in the output directory called qClassification.model. The corresponding ARFF file used for generating this model is called qClassification.arff. The classifier cross-validation accuracy is around 72%. Following is the confusion matrix.

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
193	30	16	0	0	1	| a = ABBR
150	1920	841	81	108	204	| b = DESC
4	366	2507	393	285	72	| c = ENTY
7	93	838	2288	153	45	| d = HUM
0	91	486	94	1691	14	| e = LOC
0	200	466	61	78	1675	| f = NUM

2] A lucene index for the wordnet synonyms was constructed for facilitating fast access to the synonyms during the feature generation process. The prolog version of WordNet [2] was used and the lucene API for WordNet for creating the index was used. This index is present in the resources directory as wn-syn-index.

3] The data index was created using Lucene technology. It is present in the data directory.

The online tasks involved the following:

1] Question to query transformer which identifies the key topic and modifiers. The POS tagger was used to tag the probe question and using the tag information a weighted query was constructed.

[1] http://developer.yahoo.com/answers/

[2] http://wordnet.princeton.edu/obtain

2] Using this weighted query, the lucene index is queried. The results are first sanitized, tokenized and then tagged using the POS tagger.

3] The various features for the results are generated and the SimpleKMeans from the WEKA package is used for clustering.

7.3 Use of external software packages

For the purpose of getting POS tags for the questions, the Stanford POS tagger was used. Lucene was used as the search engine for the first step of retrieving the base set of questions. The NaiveBayes classifier from Weka was training the question classifier. Weka's SimpleKMeans Clustering was used for the final clustering of the results.

7.4 Evaluation of results

Measuring the accuracy of clustering algorithms is known to be a difficult problem. In contrast to supervised classification, unsupervised clustering does not admit an obvious characterization (e.g. classification error) of the associated accuracy.

A commonly used means of assessing the accuracy of clustering algorithms is to compare the clusters they produce with "ground truth" consisting of classes assigned manually or by some other trusted means. Even generating the "ground truth" for this particular problem is hard since ideally we would like to look at all questions in the data to make sure that we collected all questions that are semantically related to a probe question. Had an automatic technique for generating the "ground truth" be known, one would have used it for the original problem itself.

In this situation, the only possible way to evaluate the results is by spending some time in manually evaluating the clustering results. It is not wise to have a 0/1 scale for evaluation which says that the results are good or bad. With the minimal complication, I decided to have a 4-point (0,1,2,3) scale where 0 says that the clustering is poor, 1 says it is OK, 2 says that it is good and 3 says that this is the way I would have clustered the results. The average score on this 4-point scale was noted and for an evaluation test of 100 questions, the score was 2.6/3.0.

Examples:

```
Probe → what's the best way to introduce a cat to a home with animals?
Results →
Cluster 1:
What is the best way to introduce a kitten into a house with an indoor/outdoor cat that is about 7 years old?
What is a safe way to introduce a kitten to an adult cat?
Cluster 2:
What is the best way to introduce a new dog to our cat without bloodshed?
What is the best way to introduce my new dog with my cat?
 What is the best way to introduce a new cat, or kitten into a dog household where the dog is 4 years old.?
Cluster 3: what's the best way to introduce a cat to a home with animals?
Cluster 4: What is the best way to introduce my cat to my new chihuahua?
Cluster 5: What's the best way to introduce a new cat to the family?
```

8 CONCLUSION AND FUTURE WORK

The technique of clustering the results in order to identify related questions in different groups of relatedness looks promising. However, most of the work is done by the lexical features and not much has been achieved by taking any semantics into account. For future work, semantic features such as subject/object relationship of the topic words with the main verb, more involved WordNet similarity measures will be explored. One of the aims of this project was to explore the methods for corpus-level statistics for identifying similar words and phrases not available in external resources such as the WordNet. These will be explored in detail in future.

9 REFERENCES

1 Pablo Ariel Duboue and Jennifer Chu-Carroll. Answering the Question YouWish They Had Asked: The Impact of Paraphrasing for Question Answering. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, New York, June 2006, 33-36.

2 Ali Ibrahim and Boris Katz and Jimmy Lin. Extracting Structural Paraphrases from Aligned Monolingual Corpora. In *Proceedings of the second international workshop on paraphrasing*, Sapporo, Japan, July 2002.

3 Jiwoon Jeon , W. Bruce Croft , Joon Ho Lee, Finding similar questions in large question and answer archives, *Proceedings of the 14th ACM international conference on Information and knowledge management*, October 31-November 05, 2005, Bremen, Germany

4 Jiwoon Jeon , W. Bruce Croft , Joon Ho Lee, Finding semantically similar questions based on their answers,

Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, August 15-19, 2005, Salvador, Brazil

5 Regina Barzilay and Kathleen McKeown. Extracting paraphrases from a parallel corpus. *In Proceedings of the ACL/EACL*, Toulouse, 2001.

6 Shiqi Zhao, Ming Zhou, Ting Liu: Learning Question Paraphrases for QA from Encarta Logs. *IJCAI 2007*: 1795-1801

7 Is Question Answering an acquired skill?, Ganesh Ramakrishnan, Soumen Chakrabarti, Deepa Paranjpe, and Pushpak Bhattacharyya. *WWW2004*, New York City

8 Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

9 Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). "BLEU: a method for automatic evaluation of machine translation" in *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics* pp. 311--318