

Automatic Generation of Information-seeking Questions Using Concept Clusters

Shuguang Li

Department of Computer Science
University of York, YO10 5DD, UK
sgli@cs.york.ac.uk

Suresh Manandhar

Department of Computer Science
University of York, YO10 5DD, UK
suresh@cs.york.ac.uk

Abstract

One of the basic problems of efficiently generating information-seeking dialogue in interactive question answering is to find the topic of an information-seeking question with respect to the answer documents. In this paper we propose an approach to solving this problem using concept clusters. Our empirical results on TREC collections and our ambiguous question collection shows that this approach can be successfully employed to handle ambiguous and list questions.

1 Introduction

Question Answering systems have received a lot of interest from NLP researchers during the past years. But it is often the case that traditional QA systems cannot satisfy the information needs of the users as the question processing part may fail to properly classify the question or the information needed for extracting and generating the answer is either implicit or not present in the question. In such cases, interactive dialogue is needed to clarify the information needs and reformulate the question in a way that will help the system to find the correct answer.

Due to the fact that casual users often ask questions with ambiguity and vagueness, and most of the questions have multiple answers, current QA systems return a list of answers for most questions. The answers for one question usually belong to different topics. In order to satisfy the information needs of the user, information-seeking dialogue should take advantage of the inherent grouping of the answers.

Several methods have been investigated for generating topics for questions in information-seeking dialogue. Hori et al. (2003) proposed a method for generating the topics for disambiguation questions. The scores are computed purely based on

the syntactic ambiguity present in the question. Phrases that are not modified by other phrases are considered to be highly ambiguous while phrases that are modified are considered less ambiguous. Small et al. (2004) utilizes clarification dialogue to reduce the misunderstanding of the questions between the HITIQA system and the user. The topics for such clarification questions are based on manually constructed topic frames. Similarly in (Hickl et al., 2006), suggestions are made to users in the form of predictive question and answer pairs (known as QUABs) which are either generated automatically from the set of documents returned for a query (using techniques first described in (Harabagiu et al., 2005), or are selected from a large database of questions-answer pairs created offline (prior to a dialogue) by human annotators. In Curtis et al. (2005), query expansion of the question based on Cyc Knowledge is used to generate topics for clarification questions. In Duan et al. (2008), the tree-cutting model is used to select topics from a set of relevant questions from Yahoo Answers.

None of the above methods consider the contexts of the list of answers in the documents returned by QA systems. The topic of a good information-seeking question should not only be relevant to the original question but also should be able to distinguish each answer from the others so that the new information can reduce the ambiguity and vagueness in the original question. Instead of using traditional clustering methods on categorization of web results, we present a new topic generation approach using concept clusters and a separability scoring mechanism for ranking the topics.

2 Topic Generation Based on Concept Clustering

Text categorization and clustering especially hierarchical clustering are predominant approaches to organizing large amounts of information into top-

ics or categories. But the main issue of categorization is that it is still difficult to automatically construct a good category structure, and manually formed hierarchies are usually small. And the main challenge of clustering algorithms is that the automatically formed cluster hierarchy may be unreadable or meaningless for human users. In order to overcome the limits of the above methods, we propose a concept clusters method and choose the labels of the clusters as topics.

Recent research on automatically extracting concepts and clusters of words from large database makes it feasible to grow a big set of concept clusters. Clustering by Committee (CBC) in Pantel et al. (2002) made use of the fact that words in the same cluster tend to appear in similar contexts. Pasca et al. (2008) utilized Google logs and lexico-syntactic patterns to get clusters with labels simultaneously. Google also released Google Sets which can be used to grow concept clusters with different sizes.

Currently our clusters are the union of the sets generated by the above three approaches, and we label them using the method described in Pasca et al. (2008). We define the **concept clusters** in our collection as $\{C_1, C_2, \dots, C_n\}$. $C_i = \{e_{i1}, e_{i2}, \dots, e_{im}\}$, e_{ij} is j^{th} subtopic of cluster C_i and m is the size of C_i .

We designed our system to take a question and its corresponding list of answers as input and then retrieve Google snippet documents for each of the answers with respect to the question. In a vectorspace model, a document is represented by a vector of keywords extracted from the document, with associated weights representing the importance of the keywords in the document and within the whole document collection. A document D_j in the collection is represented as $\{W_{0j}, W_{1j}, \dots, W_{nj}\}$, and W_{ij} is the weight of word i in document j . Here we use our concept clusters to create concept cluster vectors. A document D_j now is represented as $\langle WC_{1j}, WC_{2j}, \dots, WC_{nj} \rangle$, and WC_{ij} is the score vector of document D_j for concept cluster C_i :

$WC_{ij} = \langle Score_j(e_{i1}), Score_j(e_{i2}), \dots, Score_j(e_{im}) \rangle$
 $Score_j(e_{ip})$ is the weight of subtopic e_{ip} of cluster C_i in document D_j .

Currently we use tf-idf scheme (Yang et al., 1999) to calculate the weight of subtopics.

3 Concept Cluster Separability Measure

We view different concept clusters from the contexts of the answers as different groups of features that can be used to classify the answers documents. We rank different context features by their separability on the answers. Currently our system retrieves the answers from Google search snippets, and each snippet is quite short. So we combine the top 50 snippets for one answer into one document. One answer is associated with one such big document. We propose the following interclass measure to compare the separability of different clusters:

$$Score(C_i) = \frac{D}{N} \sum_{p < q}^N Dis(D_p, D_q),$$

D is the Dimension Penalty score, $D = \frac{1}{M}$,
 M is the size of cluster C_i ,
 N is the combined total number of classes from all the answers

$$Dis(D_p, D_q) = \sqrt{\sum_{m=0}^n (Score_p(e_{im}) - Score_q(e_{im}))^2}$$

We introduce D , the "Dimension Penalty" score which gives higher penalty to bigger clusters. Currently we use the reciprocal of the size of the cluster. The second part is the average pairwise distance between answers. N is the total number of classes of the answers. Next we describe in detail how to use the concept cluster vectors and separability measure to rank clusters.

4 Cluster Ranking Algorithm

Input:

Answer set $A = \{A_1, A_2, \dots, A_p\}$;
 Documents set $D = \{D_1, D_2, \dots, D_p\}$ associated with answer set A ;
 Concept cluster set $CS = \{C_i \mid \text{some of the subtopics from } C_i \text{ occurs in } D\}$;
 Threshold Θ_1, Θ_2 ; The question Q ;
 Concept cluster set $QS = \{C_i \mid \text{some of the subtopics from } C_i \text{ occurs in } Q\}$

Output:

$T = \{ \langle C_i, Score \rangle \}$, a set of pairs of a concept cluster and its ranking score;

QS;

Variables: X, Y ;

Steps:

1. $CS = CS - QS$
2. For each cluster C_i in CS
3. $X =$ No. of answers in which context subtopics from C_i are present;
4. $Y =$ No. of subtopics from C_i that occurs in the answers' contexts;
5. If $X < \Theta_1$ or $Y < \Theta_2$
6. delete C_i from CS
7. continue
8. Represent every document as a concept cluster vector on C_i (see section 2)
9. Calculate the $Score(C_i)$ using our separability measure
10. Store $\langle C_i, Score \rangle$ in T
11. return T the medoid.

Figure 1: Concept Cluster Ranking Algorithm

Figure 1 describes the algorithm for ranking concept clusters based on their separability score. This algorithm starts by deleting all

the clusters which are in QS from CS so that we only focus on the context clusters whose subtopics are present in the answers. However in some cases this assumption is incorrect¹. Taking the question shown in Table 2 for example, there are 6 answers for question LQ1, and in **Step 1** $CS = \{C_{41}American\ State, C_{1522}Times, C_{414}Tournament, C_{10004}Year, \dots\}$ and $QS = \{C_{4545}Event\}$. Using cluster C_{414} (see Table 2), $D = \{D_1\{Daytona\ 500, 24\ Hours\ of\ Daytona, 24\ Hours\ of\ Le\ Mans, \dots\}, D_2\{3M\ Performance\ 400, Cummins\ 200, \dots\}, D_3\{Indy\ 500, Truck\ series, \dots\}, \dots\}$, and hence the vector representation for a given document D_j using C_{414} will be $\langle Score_j(indy\ 500), Score_j(Cummins\ 200), Score_j(daytona\ 500), \dots \rangle$.

In **Step 2** through **11** from Figure 1, for each context cluster C_i in CS we calculate X (the number of answers in which context subtopics from C_i are present), and Y (the number of subtopics from C_i that occurs in the answers' contexts). We would like the clusters to hold two characteristics: (a) at least occur in Θ_1 answers as we want to have a cluster whose subtopics are widely distributed in the answers. Currently we set Θ_1 as half the number of the answers; (b) at least have Θ_2 subtopics occurring in the answers' documents. We set Θ_2 as the number of the answers. For example, for cluster C_{414} , $X = 6$, $Y = 10$, $\Theta_1 = 3$ and $\Theta_2 = 6$, so this cluster has the above two characteristics. If a cluster has the above two characteristics, we use our separability measure described in section 3 to calculate a score for this cluster. The size of C_{414} is 11, so $Score(C_{414}) = \frac{1}{11 \times 6} \sum_{p < q}^N Dis(D_p, D_q)$. Ranking the clusters based on this separability score means we will select a cluster which has several subtopics occurring in the answers and the answers are distinguished from each other because they belong to these different subtopics. The top three clusters for question LQ1 is shown in Table 2.

5 Experiment

5.1 Data Set and Baseline Method

To the best of our knowledge, the only available test data of multiple answer questions are list questions from TREC 2004-2007 Data. For our first

¹For the question "In which movies did Christopher Reeve acted?", cluster Actor{Christopher Reeve, michael caine, anthony hopkins, ...} is quite useful. While for "Which country won the football world cup?" cluster Sports{football, hockey, ...} is useless.

list question collection we randomly selected 200 questions which have at least 3 answers. We changed the list questions to factoid ones with additional words from their context questions to eliminate ellipsis and reference. For the ambiguous questions, we manually choose 200 questions from TREC 1999-2007 data and some questions discussed as examples in Hori et al. (2003) and Burger et al. (2001).

We compare our approach with a baseline method. Our baseline system does not rank the clusters by the above separability score instead it prefers the cluster which occurs in more answers and have more subtopics distributed in the answer documents. If we still use X to represent the number of answers in which context subtopics from one cluster are present and Y to represent the number of subtopics from this cluster that occurs in the answers' contexts, for the baseline system, we will use $X \times Y$ to rank all the concept clusters found in the contexts.

5.2 Results and Error Analysis

We applied our algorithm on the two collections of questions. Two assessors were involved in the manual judgments with an inter-rater agreement of 97%. For each approach, we obtained the top 20 clusters based on their scores. Given a cluster with its subtopics in the contexts of the answers, an assessor manually labeled each cluster 'good' or 'bad'. If it is labeled 'good', the cluster is deemed relevant to the question and the cluster's label could be used as dialogue seeking question's topic to distinguish one answer from the others. Otherwise, the assessor will label a cluster as 'bad'. We use the above two ranking approaches to rank the clusters for each question. Table 1 provides the statistics of the performance on the the two question collection. List.B means the baseline method on the list question set while Ambiguous.S means our separability method on the ambiguous questions. The 'MAP' column is the mean of average precisions over the set of clusters. The 'P@1' column is the precision of the top one cluster while the 'P@3' column is the precision of the top three clusters². The 'Err@3' column is the percentage of questions whose top three clusters are all labeled 'bad'. One example associated with the manually constructed desirable questions

²'P@3' is the number of 'good' clusters out of the top three clusters

Table 1: Experiment results

Methods	MAP	P@1	P@3	Err@3
List_B	41.3%	42.1%	27.7%	33.0%
List_S	60.3%	90.0%	81.3%	11.0%
Ambiguous_B	31.1%	33.2%	21.8%	47.1%
Ambiguous_S	53.6%	71.1%	64.2%	29.7%

Table 2: TREC Question Examples

LQ1:	Who is the winners of the NASCAR races?
1 st	C_{414} (Tournament):{indy 500, Cummins 200, daytona 500, ...}
Q1	Which Tournament are you interested in?
2 nd	C_{41} (American State):{houston, baltimore, los angeles, ...}
Q2	Which American State were the races held?
3 rd	C_{1522} (Times):{once, twice, three times, ...}
Q3	How many times did the winner win?

is shown in Table 2.

From Table 1, we can see that our approach outperforms the baseline approach in terms of all the measures. We can see that 11% of the questions have no ‘good’ clusters. Further analysis of the answer documents shows that the ‘bad’ clusters fall into four categories. First, there are noisy subtopics in some clusters. Second, some questions’ clusters are all labeled ‘bad’ because the contexts for different answers are too similar. Third, unstructured web document soften contain multiple subtopics. This means that different subtopics are in the context of the same answer. Currently we only look for context words while not using any scheme to specify whether there is a relationship between the answer and the subtopics. Finally, for other ‘bad’ cases and the questions with no good clusters all of the separability scores are quite low. This is because the answers fall into different topics which do not share a common topic in our cluster collection.

6 Conclusion and Discussion

This paper proposes a new approach to solve the problem of generating an information-seeking question’s topic using concept clusters that can be used in a clarification dialogue to handle ambiguous questions. Our empirical results show that this approach leads to good performance on TREC collections and our ambiguous question collections. The contribution of this paper are: (1) a new concept cluster method that maps a document into a vector of subtopics; (2) a new ranking scheme to

rank the context clusters according to their separability. The labels of the chosen clusters can be used as topics in an information-seeking question. Finally our approach shows significant improvement (nearly 48% points) over comparable baseline system.

But currently we only consider the context clusters while ignoring the clusters associated with the questions. In the future, we will further investigate the relationships between the concept clusters in the question and the answers.

References

- Tiphaine Dalmas, Bonnie L. Webber: Answer comparison in automated question answering. *J. Applied Logic (JAPLL)* 5(1):104-120, (2007).
- Chiori Hori, Sadaoki Furui: A new approach to automatic speech summarization. *IEEE Transactions on Multimedia (TMM)* 5(3):368-378, (2003).
- Sharon Small and Tomek Strzalkowski, HITQA: A Data Driven Approach to Interactive Analytical Question Answering, in *Proceedings of HLT-NAACL 2004: Short Papers*, (2004).
- Andrew Hickl, Patrick Wang, John Lehmann, Sanda M. Harabagiu: FERRET: Interactive Question-Answering for Real-World Environments. *ACL*, (2006).
- Sanda M. Harabagiu, Andrew Hickl, John Lehmann, Dan I. Moldovan: Experiments with Interactive Question-Answering. *ACL*, (2005).
- John Burger et al.: Issues, Tasks and Program Structures to Roadmap Research in Question and Answering (Q&A), DARPA/NSF committee publication, (2001).
- Patrick Pantel, Dekang Lin: Document clustering with committees. *SIGIR 2002*:199-206, (2002).
- Marius Pasca and Benjamin Van Durme: Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs. *ACL*, (2008).
- Sanda M. Harabagiu, Andrew Hickl, V. Finley Laccatusu: Satisfying information needs with multi-document summaries. *Inf. Process. Manage. (IPM)* 43(6):1619-1642, (2007).
- Huizhong Duan, Yunbo Cao, Chin-Yew Lin and Yong Yu: Searching Questions by Identifying Question Topic and Question Focus. *ACL*, (2008).
- Jon Curtis, G. Matthews and D. Baxter: On the Effective Use of Cyc in a Question Answering System. *IJCAI Workshop on Knowledge and Reasoning for Answering Questions*, Edinburgh, (2005).