UofL at TAC 2008 Update Summarization and Question Answering

Yllias Chali, Sadid A. Hasan and Shafiq R. Joty
University of Lethbridge
Lethbridge, AB, Canada
{chali,hasan,jotys}@cs.uleth.ca

Abstract

In this paper, we describe our update summarization and question answering (QA) systems participated in the TAC 2008 competition. We submitted three runs for the update summarization task using unsupervised and supervised techniques. On the other hand, the question answering system is built on our previous system participated in TREC 2007 QA track with different approach followed for the squishy list type questions. We submitted a single run for the QA task. This paper also presents the preliminary evaluation results of our systems.

1 Introduction

The main goal of the TAC summarization[1] track is to foster research on systems that produce summaries of documents. The focus is on systems that can produce well-organized, fluent, query-focused summaries of text. Users looking for information about a series of related events, often face an intimidating task of filtering out redundant information. To help combating this problem, update summarization task is piloted in DUC[2] 2007 with the hope to deliver focused distilled information to a user who has already read a set of older documents covering the same topic. Update summarization is similar to query-focused summarization in that the system is presented with a topic statement (consisting of one or more questions) and a cluster of on-topic documents; however, in this information searching scenario, it is assumed that the user is already familiar with some aspects of the topic (represented by a set of earlier documents). In TAC 2008 update summarization task, each system is required to present the user with the information from a subsequent set of news articles that is both novel and relevant to their query given that the purpose of each update summary will be to inform the reader of new information about a particular topic. In this paper, we describe the ins and outs of the unsupervised and supervised approaches we followed in the submitted three runs for the update summarization task.

The TAC Question Answering[3] (QA) track promotes research on systems that search large document collections and retrieve precise answers to natural language questions (rather than entire documents). Here, the main focus is on systems that can function in unrestricted domains. The TAC 2008 QA task concentrates on finding answers to opinion questions. The 2008 QA task is similar to the main QA task in TREC[4] 2007 in that the test set consists of question series. However, each series in TAC 2008 asks for people's opinions about a particular target (rather than general information about the target), and the questions are asked over only blog documents. There are two types of questions: rigid list questions and squishy list questions. To answer the rigid list type questions, our question answering system follows the similar approach as the one we followed to answer the list questions in TREC 2007 [1]. We answer the squishy list questions using the summarization technique. In this paper, we describe our QA system for answering

[1] http://www.nist.gov/tac/tracks/2008/summarization/

[2] http://www-nlpir.nist.gov/projects/duc/

[3] http://www.nist.gov/tac/tracks/2008/qa/

[4] http://trec.nist.gov/

both rigid list and squishy list type questions.

The rest of the paper is organized as follows: Section 2 focuses on the update summarization systems. In Section 3, we discuss the QA systems. Finally, Section 4 concludes the paper.

2 Update Summarization Systems

2.1 Problem Definition

The TAC 2008 update summarization task can be defined as below:

"Given a topic statement and two on-topic document clusters (A and B), write 2 well-organized summaries (one for Set A and one for Set B) that address the information need expressed in the corresponding topic statement where: the summary for Set A should be a straightforward query-focused summary and the update summary for Set B is also query-focused but should be written under the assumption that the user of the summary has already read the documents in Set A. Each summary must be no longer than 100 words."

Among the three submitted runs, we used an empirical approach for the first run to generate update summaries. The other runs are prepared following a supervised technique, the Support Vector Machines (SVM). For all these approaches, we at first extract features for each of the document sentences that measure the importance of the sentence in the document and its relevancy to the user query.

2.2 Feature Extraction

We analyze the sentences in the document collection in various levels and each of the document-sentences is represented as a vector of feature-values. We consider both query-related features and some other important features. For the run 1, the features we used are: N-gram overlap, Longest Common Subsequence (LCS), Weighted LCS (WLCS), skip-bigram, head and head-related words overlap, lexical semantic features, graph-based similarity measure, Basic Element (BE) overlap, syntactic tree similarity, position of sentences, length of sentences, Named Entity (NE) and cue word match [3, 12, 11, 4].

Same features are used for the run 2 and run 3 except head overlap, BE overlap, graph-based measures and syntactic trees. Additionally, we used the Title Match feature as a document sentence yields more importance if any title word is found in it [4].

After feature extraction, we apply our approaches on the extracted features to rank the document sentences and then we perform redundancy checking.

2.3 Testing Corpus

The test data set for TAC 2008 update summarization task comprises 48 topics. Each topic has a topic statement (title and narrative) and 20 relevant documents which are divided into 2 sets: Document Set A and Document Set B. Each document set has 10 documents, where all the documents in Set A chronologically precede the documents in Set B. The document sets came from the AQUAINT-2 collection of news articles.

2.4 Experimental Setup

For run 1, in order to fine-tune the weights of the features, we used a local search technique on the data provided by NIST and then we followed the similar procedure as [3] to rank the document sentences. We generated summaries for the 48 topics of the TAC 2008 update summarization task. The task indicates that we need redundancy checking in two levels: intra-cluster redundancy checking and inter-cluster redundancy checking. The intra-cluster redundancy checking ensures that the sentence that is being included in the summary (query-focused summary for each cluster A and B) is not bearing the same information as the earlier included sentences of that same summary. On the other hand, inter-cluster redundancy checking is done to ensure that the sentences in the summary of cluster B are not similar to the document sentences of cluster A. We modeled the two redundancy checking by BE overlap [6] between an intermediate summary (i.e. intra-cluster) or full document set (i.e. inter-cluster) and a to-be-added candidate summary sentence. We call this overlap ratio R, where R is between 0 and 1 inclusively. Setting $R = 0.7$ means that a candidate summary sentence, s, can be added to an intermediate summary, S, if the sentence has a BE overlap ratio less than or equal to 0.7. We set 0.6 and 0.45 as the values of R for *intra-cluster redundancy* *checking* and *inter-cluster redundancy checking*, respectively.

For run 2 and run 3, we used the Support Vector Machines (SVM), a well known algorithm to perform classification tasks effectively, to generate update summaries for the second and the third run. We used $g(x)$ (the normalized distance from the hyperplane to x) to rank the sentences for reducing the intra-cluster redundancy. In addition, while generating summaries for the cluster B, inter-cluster redundancy minimization was applied using ROUGE[5] [9] similarity measures. We measured the ROUGE scores between the candidate summary sentences of the cluster B and the sentences of cluster A. In the end, the less similar candidate sentences were selected to be included in the final update summaries of the cluster B. We employed the ROUGE package to automatically label the data sets of DUC 2006 and DUC 2007 in order to use them as the training data for SVM. We used the second-order polynomial kernel and applied 3-fold cross validation with randomized local-grid search [7] for estimating the value of the trade-off parameter C. We tried the value of C in 2^i by following some heuristics, where $i \in \{-5, -4, \cdots, 4, 5\}$ and set C as the best performed value of 0.0625 for run 2 while the second best performed value 0.03125 was assigned for run 3. We used the SVM^{light6} [8] package.

2.5 Evaluation Results

Each topic statement and its 2 document sets were given to 4 different NIST assessors. For each document set, the assessor created a 100-word summary that addresses the information need expressed in the topic statement according to the task guidelines. NIST conducted a manual evaluation of summary content (for top two runs) based on the Pyramid Method[7] using the multiple model summaries created by the assessors. The assessor also gave an overall responsiveness score to each peer summary. In addition to the Pyramid evaluation, NIST used automatic evalution tools ROUGE and BE (Basic Elements) to measure the performance of the systems. Table 1 to Table 3 show the manual, ROUGE and BE evaluation results of our systems, respectively. Each column (except the first) of the tables stands for the run id of our systems along with the NIST assigned peer id. From these evaluation results, we can conclude that SVM (run 2 and run 3) performs far better than the empirical approach (run 1) while generating query-focused summaries and in case of inter-cluster redundancy minimization, BE overlap measure (run 1) is better than the ROUGE overlap measure (run 2).

Score	UofL1:31	UofL2:56
Modified Pyramid Score-A	0.183	0.241
Number of SCUs-A	2.958	3.813
Linguistic Quality-A	2.396	2.479
Overall Responsiveness-A	2.000	2.333
Modified Pyramid Score-B	0.141	0.126
Number of SCUs-B	2.021	1.771
Linguistic Quality-B	2.479	1.833
Overall Responsiveness-B	1.875	1.563

Table 1. Manual Evaluation

Score	UofL1:31	UofL2:56	UofL3:71
ROUGE2 R-A	0.053	0.065	0.068
ROUGE2 R-B	0.048	0.035	0.032
ROUGESU4 R-A	0.093	0.102	0.103
ROUGESU4 R-B	0.090	0.068	0.067

Table 2. ROUGE Evaluation

Score	UofL1:31	UofL2:56	UofL3:71
BE R-A	0.032	0.033	0.036
BE R-B	0.027	0.025	0.022

Table 3. BE Evaluation

3 Question Answering System

3.1 Problem Definition

The TAC 2008 Question Answering (QA) task asks for people's opinions about a particular target (rather than general information about the target), and the questions are asked over only blog documents. Two types of questions are there: rigid list questions and squishy list

[5] A Package for Automatic Evaluation of Summaries

[6] http://svmlight.joachims.org/

[7] http://www1.cs.columbia.edu/b̃ecky/DUC2006/2006-pyramid-guidelines.html

questions which are the requests for a set of instances of a specified type. Rigid list questions require exact answers to be returned whereas responses to squishy list questions may not be exact. To be specific, for each rigid list question, the system should return an unordered, non-empty set of [answer-string, docid] pairs, where each pair is called an instance. The answer-string does not have to appear literally in a document in order for the document to support it as being a correct answer item. On the other hand, the response for a squishy list question is syntactically the same as for a rigid list question: an unordered, non-empty set of [answer-string, docid] pairs. However, the interpretation of this set is different in the sense that there is no expectation of an exact answer to squishy list questions. We describe how our QA system deals with both types of questions one by one.

3.2 Answering Rigid List Questions

We answer the rigid list type questions taking the same approach as we did for list questions in TREC 2007 QA track which is based on document tagging and question classification [1]. For easier question classification, we normalize some of the questions into a standard form using techniques such as: aprostophy s ('s) resolution, pronoun resolution and "What" normalization. We use java based Lucene [8] as our information retrieval system. The document tagging module of our system tags useful information from the passages retrieved by Lucene. We use the Lingpipe[9] to resolve the coreference. We use OAK Tagger [10] to tag the passages with: 1. Part of Speech 2. Chunked Part of Speech and 3. Named Entity. Each word in WordNet [5] has multiple senses for the different ways the word can be used. To tag the correct sense, we used our Word Sense Disambiguation (WSD) system [2]. We give a score to each possible answer that is extracted, and return the answers with the highest scores which are actually the answers to the rigid list type question. Our system used the approach of extracting named entities to rank the answers following the patterns we used in TREC 2007 [1].

3.3 Answering Squishy List Questions

Squishy list questions are opinion questions that can be considered as complex questions. These questions cannot be answered using the same technique of answering rigid list questions. For example consider the following two questions:

What did American voters admire about Rudy Guiliani?

What qualities did not endear Rudy Guiliani to some American voters?

These questions cannot be answered by just extracting the named entities rather they need more understanding of text. Our approach to answer the squishy list question is to generate a non-redundant query-focused summary that will address the information needs of the question. We used our general query-focused summarizer to produce answers to these questions. The general query-focused summarizer is same as the update summarizer (run 1) except the redundancy checking module. Here we have just the intra cluster redundancy checking as we extract only one cluster of sentences. The summary length is limited to 250 words for each question.

3.4 Test Data

The test data set consists of 50 targets, each with a series of 2-4 questions about that target. Each series is an abstraction of a user session with a QA system which contains a number of rigid list questions and a number of squishy list questions. The answers for all questions in the test set are drawn from the TREC Blog06 collection[10].

3.5 Evaluation Results

In TAC 2008, individual rigid list questions are scored by first computing instance recall (IR) and instance precision (IP) using the final answer set, and combining those scores using the F measure with recall and precision equally weighted. Squishy list questions are scored using nugget recall (NR) and an approximation to nugget precision (NP) based on length which are combined using the F measure with beta=3 meaning recall weighted more

[8] http://jkarta.apache.org/lucene/

[9] http://www.alias-i.com/lingpipe

[10] Created by the University of Glasgow for the TREC 2006 Blog Track

heavily than precision. As, these questions (rigid list and squishy list) have different scoring metrics, NIST computes both the rigid-list-score and squishy-list-score for each series. Table 4 shows the comparison of our QA system's (UofL1) scores (average F-measure for rigid lists, average pyramid F score for squishy lists and average per-series score) with the best, median and worst average scores (computed over all the systems) over 90 rigid list questions, 87 squishy list questions and 50 series.

Avg. Score	UofL1	Best	Median	Worst
F (rigid)	0.062	0.156	0.063	0.000
Pyr. F (squishy)	0.122	0.186	0.091	0.018
Per-series	0.102	0.168	0.093	0.011

Table 4. QA Evaluation Results

4 Conclusion

In this paper, we described our participation in TAC 2008. The evaluation results show that in update summarization task, our systems did reasonably well although there is still room for improvement. In future, we have the plan to take the advantages of complex question decomposition in order to get more focused summaries. Moreover, we would also experiment with different inter-cluster redundancy minimization approaches to get improved overall scores. In QA track, our systems performed moderately fine as all our scores are well above the median (except the average F-score for the rigid list questions which is pretty close to the median). As TAC 2008 question types were opinion, we can improve our scores further if we can judge each document sentence individually to extract the implicit positive or negative view point.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) research grant and the University of Lethbridge.

References

[1] Y. Chali and S. R. Joty. University of Lethbridge's Participation in TREC-2007 QA Track. In *Proceedings of the sixteenth Text REtrieval Conference*, Gaithersburg, 2007. NIST.

[2] Y. Chali and S. R. Joty. Word Sense Disambiguation Using Lexical Cohesion. In *Proceedings of the 4th International Conference on Semantic Evaluations*, pages 476–479, Prague, 2007. ACL.

[3] Y. Chali and S. R. Joty. Selecting sentences for answering complex questions. In *Proceedings of EMNLP*, 2008.

[4] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285, 1969.

[5] C. Fellbaum. WordNet - An Electronic Lexical Database. Cambridge, MA, 1998. MIT Press.

[6] E. Hovy, C.-Y. Lin, and L. Zhou. A BE-based Multi-document Summarizer with Query Interpretation. In *Proceedings of the Document Understanding Conference*, Vancouver, B.C. Canada, 2005.

[7] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A Practical Guide to Support Vector Classification, National Taiwan University, Taipei 106, Taiwan, http://www.csie.ntu.edu.tw/ cjlin. 2008.

[8] T. Joachims. Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*, 1999.

[9] C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain, 2004.

[10] S. Sekine. Proteus Project OAK System (English Sentence Analyzer), http://nlp.nyu.edu/oak. 2002.

[11] S. Sekine and C. A. Nobata. Sentence extraction with information extraction technique. In *Proceedings of the Document Understanding Conference*, 2001.

[12] K. Zechner. Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. In *Proceedings of the 16th COLING*, pages 986–989, 1996.