Query-Chain Focused Summarization

Tal Baumel
Dept. of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
talbau@cs.bgu.ac.il

Raphael Cohen
Dept. of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
cohenrap@cs.bgu.ac.il

Michael Elhadad
Dept. of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
elhadad@cs.bgu.ac.il

Abstract

Update summarization is a form of multi-document summarization where a document set must be summarized in the context of other documents assumed to be known. Efficient update summarization must focus on identifying new information and avoiding repetition of known information. In *Query-focused summarization*, the task is to produce a summary as an answer to a given query. We introduce a new task, *Query-Chain Summarization*, which combines aspects of the two previous tasks: starting from a given document set, increasingly specific queries are considered, and a new summary is produced at each step. This process models *exploratory search*: a user explores a new topic by submitting a sequence of queries, inspecting a summary of the result set and phrasing a new query at each step. We present a novel dataset comprising 22 query-chains sessions of length 3 with 3 matching human summaries each in the consumer-health domain. Our analysis demonstrates that summaries produced in the context of such exploratory process are different from informative summaries. We present an algorithm for Query-Chain Summarization based on a new LDA topic model variant. Evaluation indicates the algorithm improves on strong baselines.

1 Introduction

In the past 10 years, the general objective of text summarization has been refined into more specific tasks. Such summarization tasks include: (i) Generic Multi Document Summarization: aims at summarizing a cluster of topically related documents, such as the top results of a search engine query; (ii) in Update Summarization, a set of documents is summarized while assuming the user has already read a summary of earlier documents on the same topic; (iii) in Query-Focused Summarization, the summary of a documents set is produced to convey an informative answer in the context of a specific query. The importance of these specialized tasks is that they help us distinguish criteria that lead to the selection of content in a summary: centrality, novelty, relevance, and techniques to avoid redundancy.

We present in this paper a variant summarization task which combines the two aspects of update and query-focused summarization. The task is related to *exploratory search* (Marchionini 2006). In contrast to classical information seeking, in exploratory search, the user is uncertain about the information available, and aims at learning and understanding a new topic (White and Roth 2009). In typical exploratory search behavior, a user posts a series of queries, and based on information gathered at each step, decides how to further explore a set of documents. The metaphor of *berrypicking* introduced in (Bates 1989) captures this interactive process. At each step, the user may *zoom in* to a more specific information need, *zoom out* to a more general query, or *pan sideways*, in order to investigate a new aspect of the topic.

We define *Query-Chain Focused Summarization* as follows: for each query in an exploratory search session, we aim to extract a summary that answers the information need of the user, in a manner similar to *Query-Focused Summarization*, while not repeating information already provided in previous steps, in a manner similar to *Update Summarization*. In contrast to query-focused summarization, the context of a summary is not a single query, but the set of queries that led to the current step, their result sets and the corresponding summaries.

We have constructed a novel dataset of Query-Sets with matching manual summarizations in the consumer health domain (Cline and Haynes 2001). Queries are extracted from PubMed search logs (Dogan and Murray 2009). We have analyzed this manual dataset and confirm that summaries written in the context of berry-picking are markedly different from those written for similar queries on the same document set, but without the query-chain context.

We have adapted well-known multi-document algorithms to the task, and present baseline algorithms based on LexRank (Erkan and Radev 2004), KLSum and TopicSum (Haghighi and Vanderwende 2009). We introduce a new algorithm to address the task of Query-Chain Focused Summarization, based on a new LDA topic model variant, and present an evaluation which demonstrates it improves on these baselines.

The paper is structured as follows. Section 2 formulates the task of Query-Chain Focused Summarization. Section 3 reviews related work. In Section 4, we describe the data collection process and the resulting dataset. We then present our algorithm, as well as the baseline algorithms used for evaluation. We conclude with evaluation and discussion.

2 Query- Chain Summarization

In this work, we focus on the *zoom in* aspect of the exploratory search process described above. We formulate the Query-Chain Focused Summarization (QCFS) task as follows:

Given an ordered chain of queries Q and a set of documents D, for each query $q_i \in Q$ a summary S_i is generated from D answering q_i under the assumption that the user has already read the summaries S_{i-1} for queries $q_0 ... q_{i-1}$.

A typical example of query chain in the consumer health domain we investigate includes the following 3 successive queries: (*Causes of asthma, Asthma and Allergy,* Asthma *and Mold Allergy*). We consider a single set of documents relevant to the domain of Asthma as the reference set *D*. The QCFS task consists of generating one summary of *D* as an answer to each query, so that the successive answers do not repeat information already provided in a previous answer.

3 Previous Work

We first review the closely related tasks of Update Summarization and Query-Focused Summarization. We also review key summarization algorithms that we have selected as baseline and adapted to the QCFS task.

Update Summarization focuses on identifying new information relative to a previous body of information, modeled as a set of documents. It has been introduced in shared tasks in DUC 2007 and TAC 2008. This task consists of producing a multi-document summary for a document set on a specific topic, and then a multi-document summary for a different set of articles on the same topic published at later dates. This task helps us understand how update summaries identified and focused on new information while reducing redundancy compared to the original summaries.

The TAC 2008 dataset includes 48 sets of 20 documents, each cluster split in two subsets of 10 documents (called A and B). Subset B documents were more recent. Original summaries were generated for the A subsets and update summaries were then produced for the B subsets. Human summaries and candidate systems are evaluated using the Pyramid method (Nenkova and Passonneau 2004). For automatic evaluation, ROUGE (Chin-Yew Lin 2004) variants have been proposed (Conroy, Schlesinger and O'Leary 2011). In contrast to this setup, QCFS distinguishes the subsets of documents considered at each step of the process by facets of the underlying topic, and not by chronology. In addition, the document subsets are not identified as part of the task in QCFS (as opposed to the explicit split in A and B subsets in Update Summarization).

Most systems working on Update Summarization have focused on removing redundancy. DualSum (Delort and Alfonseca 2012) is notable in attempting to directly model novelty using a specialized topic-model to distinguish words expressing background information and those introducing new information in each document.

In Query-Focused Summarization (QFS), the task consists of identifying information in a document set that is most relevant to a given query. This differs from generic summarization, where one attempts to identify central information. QFS helps us distinguish models of relevance and centrality. Unfortunately, detailed analysis of the datasets produced for QFS indicates that these two notions are not strongly distinguished in practice: (Gupta, Surabhi, Nenkova, and Jurafsky. 2007) observed that in QFS datasets, up to 57% of the words in the document sets were closely related to the query (through simple query expansion). They note that as a consequence, a generic summarizer forms a strong baseline for such biased QFS tasks.

We address this limitation of existing QFS datasets in our definition of QCFS: we identify a chain of at least 3 related queries which focus on different facets of the same central topic and require the generation of distinct summaries for each query, with little repetition across the steps.

A specific evaluation aspect of QFS measures responsiveness (how well the summary answers the specific query). QFS must rely on Information Retrieval techniques to overcome the scarceness of the query to establish relevance. As evidenced since (Daumé, Hal, Marcu 2006), Bayesian techniques have proven effective at this task: we construct a latent topic model on the basis of the document set and the query. This topic model effectively serves as a query expansion mechanism, which helps assess the relevance of individual sentences to the original query.

In recent years, three major techniques have emerged to perform multi-document summarization: graph-based methods such as LexRank (Erkan and Radev 2004), language model methods such as KLSum (Haghighi and Vanderwende 2009) and variants of KLSum based on topic models such as BayesSum (Daumé and Marcu 2006) and TopicSum (Haghighi and Vanderwende 2009).

LexRank is a stochastic graph-based method for computing the relative importance of textual units in a natural text. The LexRank algorithm builds a weighted graph $G = (V, E)$ where each vertex in V is a linguistic unit (in our case sentences) and each weighted edge in E is a measure of similarity between the nodes. In our implementation, we model similarity by computing the cosine distance between the $TF \times IDF$ vectors representing each node. After the graph is generated, the PageRank algorithm (Page, Brin, Motwani and Winograd 1999) is used to determine the most central linguistic units in the graph. To generate a summary we use the n most central lexical units, until the length of the target summary is reached. This method has no explicit control to avoid redundancy among the selected sentences, and the original algorithm does not address update or query-focused variants.

KLSum adopts a language model approach to compute relevance: the documents in the input set are modeled as a distribution over words (the original algorithm uses a unigram distribution over the bag of words in documents D). KLSum is a sentence extraction algorithm: it searches for a subset of the sentences in D with a unigram distribution as similar as possible to that of the overall collection D, but with a limited length. The algorithm uses Kullback-Lieber (KL) divergence $KL(P||Q) = \sum_w \log \frac{P(w)}{Q(w)}$ to compute the similarity of the distributions. It searches for $S^* = \text{argmin}_{|S|<L} KL(P_D||P_S)$. This search is perfomed in a greedy manner, adding sentences one by one to S until the lengh L is reached, and chosing the best sentence as measured by KL-divergence at each step. The original method has no update or query focusing capability, but as a general modeling framework it is easy to adapt to a wide range of specific tasks.

TopicSum uses an LDA-like topic model (Blei, Ng, and Jordan 2003) to classify words from a number of document sets (each set discussing a different topic) as either general non-content words, topic specific words and document specific word (this category refers to words that are specific to the writer and not shared across the document set). After the words are classified, the algorithm uses a KLSum variant to find the summary that best matches the unigram distribution of topic specific words. This method improves the results of KLSum but it also has no update summary or query answering capabilities.

4 Dataset Collection

We now describe how we have constructed a dataset to evaluate QCFS algorithms, which we are publishing freely. We selected to build our dataset in the Consumer Health domain, a popular domain in the web (Cline and Haynes 2001) providing medical information at various levels of complexity, ranging from layman and up to expert information.

The PubMed repository, while primarily serving the academic community, is also used by laymen to ask health related questions. The PubMed query logs (Dogan and Murray 2009) provide user queries with timestamps and anonymized user identification. They are publically available and include over 600K queries per day. We used these logs to extract laymen queries relating to four topics: Asthma, Lung Cancer, Obesity and Alzheimer's disease. We extracted a single day query log. From these, we extracted sessions which contained the terms "*asthma*", "*lung cancer*", "*obesity*" or "*alzheimer*". Sessions containing search tags (such as "[Author]") were removed to reduce the number of academic searches. The sessions were then manually examined and used to create zoom-in query chains of length 3 at most. The queries appear below:

Asthma:

Asthma causes→ asthma allergy→ asthma mold allergy;
Asthma treatment→asthma medication→corticosteroids;
Exercise induced asthma→ exercise for asthmatic;
Atopic dermatitis→ atopic dermatitis medications→ atopic dermatitis side effects;
Atopic dermatitis→ atopic dermatitis children→ atopic dermatitis treatment;
Atopic dermatitis → atopic dermatitis exercise activity → atopic dermatitis treatment;

Cancer:

Lung cancer→ lung cancer causes→ lung cancer symptoms;
Lung cancer diagnosis→ lung cancer treatment→lung cancer treatment side effects;
Stage of lung cancer→ lung cancer staging tests→ lung cancer TNM staging system;
Types of lung cancer→non-small cell lung cancer treatment→non-small cell lung cancer surgery;
Lung cancer in women→ risk factors for lung cancer in women→ treatment of lung cancer in women;
Lung cancer chemotherapy→ goals of lung cancer chemotherapy→ palliative care for lung cancer;

Obesity:

Salt obesity→retaining fluid;
Obesity screening→body mass index→BMI Validity;
Childhood obesity→childhood obesity low income→children diet and exercise;
Causes of childhood obesity→obesity and nutrition→school lunch;
Obesity and lifestyle change→obesity metabolism→superfoods antioxidant;
Obesity and diabetes→emergence of type 2 diabetes→type 2 diabetes and obesity in children;

Alzheimer's disease:

Alzheimer memory→helping retrieve memory alzheimer →alzheimer memory impairment nursing;
Cognitive impairment→Vascular Dementia→Vascular Dementia difference alzheimer;
Alzheimer's symptoms→alzheimer diagnosis→alzheimer medications;
Semantic dementia→first symptoms dementia→first symptoms alzheimer;

We asked medical experts to construct four document collections from well-known and reliable consumer health websites relating to the four subjects (Wikipedia, WebMD, and the NHS), so that they would provide general information relevant to the queries.

We then asked medical students to manually produce summaries of these four document collections for each query-chain. The medical students instructed construct a text of up to 250 words that provides a good answer to each query in the chain. For each query in chain the summarizers should assume that the person reading the summaries is familiar with the previous summaries in the chain so they should avoid redundancy.

Three distinct human summaries were produced for each chain. For each chain, one summary was produced for each of the three queries, where the person producing the summary was not shown the next steps in the chain when answering the first query.

To simulate the exploratory search of the user we provided the annotators with a Solr[1] query interface for each document collection. The interface allowed querying the document set, reading the documents and choosing sentences which answer the query. After choosing the sentences, annotators can copy and edit the resulting summary in order to create an answer of up to 250 words. After the first two query chain summaries, the annotators held a post-hoc discussion about the different summaries in order to adjust their conception of the task.

The statistics on the collected dataset appear in the Tables below:

Document sets	**# Documents**	**# Sentences**	**#Tokens / Unique**
Asthma	125	1,924	19,662 / 2,284
Lung-Cancer	135	1,450	17,842 / 2,228
Obesity	289	1,615	21,561 / 2,907
Alzheimer's Disease	191	1,163	14,813 / 2,508

Queries	**# Sessions**	**# Sentences**	**#Tokens / Unique**
Asthma	5	15	36 / 14
Lung-Cancer	6	18	71 / 25
Obesity	6	17	45 / 29
Alzheimer's Disease	4	12	33 / 16

Manual Summaries	**# Documents**	**# Sentences**	**#Tokens / Unique**
Asthma	45	543	6,349 / 1,011
Lung-Cancer	54	669	8,287 / 1,130
Obesity	51	538	7,079 / 1,270
Alzheimer's Disease	36	385	5,031 / 966

A key aspect of the dataset is that the same documents are summarized for each step of the chains, and we expect the summaries for each step to be different (that is, each answer is indeed responsive to the specific query it addresses). In addition, each answer is produced in the context of the previous steps, and only provides updated

[1] http://lucene.apache.org/solr/

information with respect to previous answers. To ensure that the dataset indeed reflects these two aspects (responsiveness and freshness), we empirically verified that summaries created for advanced queries are different from the summaries created for the same queries by summarizers who did not see the previous summaries in the chain. We asked from additional annotators to create manual summaries of advanced queries from the query chain without ever seeing the queries from the beginning of the chain. For example, given the chain (*asthma causes* → *asthma allergy* → *asthma mold allergy*), we asked summarizers to produce an answer for the second query (*asthma allergy*) without seeing the first step, on the same input documents.

We used ROUGE to perform this validation: ROUGE compares a summary with a set of reference summaries and source documents. We first computed the mean ROUGE score of the second query summaries. The mean ROUGE score is the mean score of each manual summary *vs*. all other summaries about the same query. We got ($r1 = 0.52, r2 = 0.22, rs4 = 0.13$). The mean ROUGE scores of the same second query summaries by people who did not see the previous query were markedly lower: ($r1 = 0.40, r2 = 0.22, rs4 = 0.01$). We only verified the asthma dataset in this manner. The results, except for the R2 test, had statistically significant difference with 95% confidence interval. All the data, code and an annotated example can be found here:

http://www.cs.bgu.ac.il/~talbau/QSMDS/dataset.html

5 Algorithms

In this section, we first explain how we adapted the previously mentioned methods to the QCFS task, thus producing 3 strong baselines. We then describe our new algorithm for QCFS.

5.1 Adapted KLSum

We adapted KLSum to QCFS by introducing a simple document selection step in the algorithm. The method is: given a query step *q*, we first select a focused subset of documents from *D, D(q)*. We then apply the usual KLSum algorithm over *D(q)*. This approach does not make any effort to reduce redundancy from step to step in the query chain. In our implementation, we compute *D(q)* by selecting the top-10 documents in *D* ranked by *TFxIDF* scores to the query, as implemented in SolR.

5.2 KL-Chain-Update

KL-Chain-Update is a slightly more sophisticated variation of KLSum that answers a query chain (instead a single query). When constructing a summary, we update the unigram distribution of the constructed summary so that it includes a smoothed distribution of the previous summaries in order to eliminate redundancy between the successive steps in the chain. For example, when we summarize the documents that were retrieved as a result to the first query, we calculate the unigram distribution in the same manner as we did in Adapted KLSum; but for the second query, we calculate the unigram distribution as if all the sentences we selected for the previous summary were selected for the current query too, with a damping factor. In this variant, the Unigram Distribution estimate of word X is computed as:

$$\left(\mathrm{CountWordIn}(X, CurrentSummary) + \frac{\mathrm{CountWordIn}(X, PreviousSummary)}{SmoothingFactor}\right) / \mathrm{NumOfWords}(CurrentSummary)$$

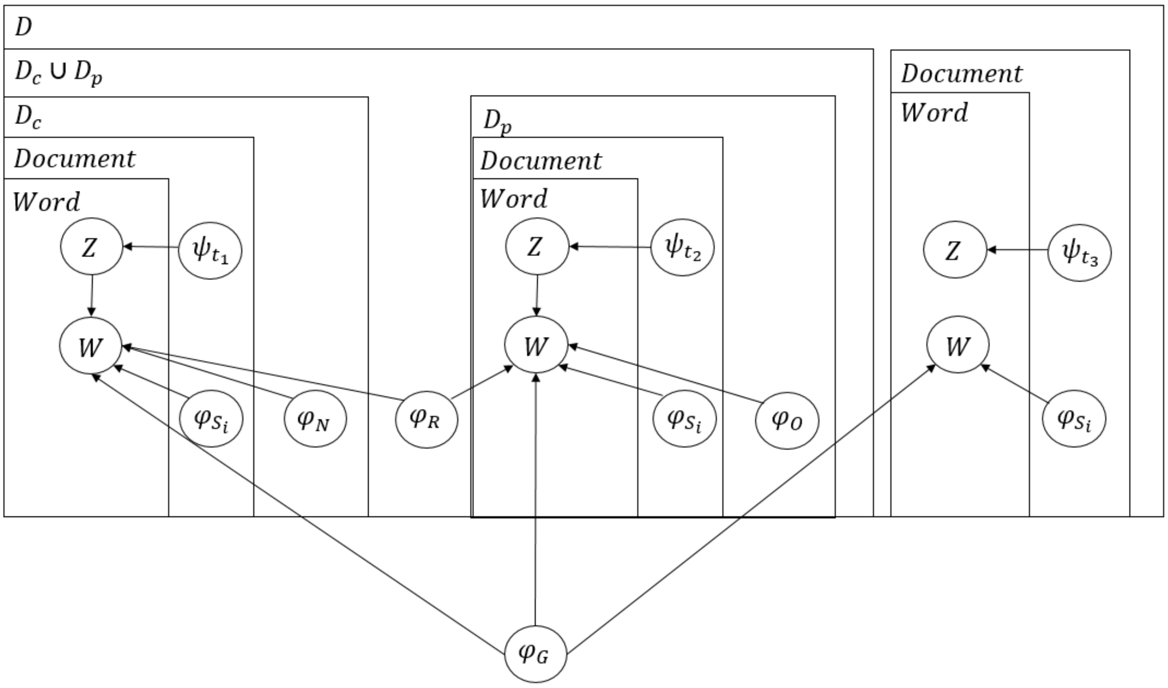

Figure 1: Plate model for our topic model.

5.3 ChainSum

ChainSum is our adaptation of TopicSum to the QCFS task. We developed a novel Topic Model to identify words that are associated to the current query and not shared with the previous queries. We achieved this with the following model. For each query in a chain, we consider the documents $D_c$which are "good answers" to the query; and D_P which are the documents used to answer the previous steps of the chain. We assume in this model that these document subsets are observable (in our implementation, we select these subsets by ranking the documents for the query based on *TFxIDF* similarity).

1. G is the general words topic, it is intended to capture stop words and non-topic specific vocabulary. Its distribution φ_G is drawn for all the documents from$Dirichlet(V,\lambda_G)$.
2. S_i is the document specific topic; it represents words which are local for a specific document. ϕ_{S_i} is drawn for each document from $Dirichlet(V,\lambda_{S_i})$.
3. N is the new content topic, which should capture words that are characteristic for D_c. ϕ_N is drawn for all the documents in D_c from $Dirichlet(V,\lambda_N)$.
4. O should capture old content from D_P, ϕ_O is drawn for all the documents in D_P from $Dirichlet(V,\lambda_O)$.
5. R topic should capture redundant information between D_c and D_p, ϕ_R is drawn for all the documents in $D_p \cup D_c$ from $Dirichlet(V,\lambda_R)$.
6. For documents from D_c we draw from the distribution ψ_{t_1} over topics (G,N,R,S_i) from a Dirichlet prior with pseudo-counts (10.0,15.0,15.0,1.0)[2]. For each word in the document, we draw a topic Z from ψ_t, and a word W from the topic indicated by Z.
7. For documents from D_p, we draw from the distribution ψ_{t_2} over topics (G,O,R,S_i) from a Dirichlet prior with pseudo-counts (10.0,15.0,15.0,1.0). The words are drawn in the same manner as in t_1.
8. For documents in $D \setminus (D_c \cup D_p)$ we draw from the distribution ψ_{t_3} over topics (G,S_i) from a Dirichlet prior with pseudo-counts (10.0,1.0). The words are also drawn in the same manner as in t_1.

We implemented inference over this topic model using Gibbs Sampling (we distribute the code of the sampler together with our dataset). After the topic model is applied to the current query, we apply KLSum only on words that are assigned to the new content topic. Figure 2 summarizes the algorithm data flow.

D_c mean size was 967 words and 375 unique words. D_P mean size was 885 words and 295 unique words. D_c and D_P shared 145 words.

We also tested a simplified version of the topic model that did not include φ_O the topic that

[2] All pseudo-counts were selected empirically

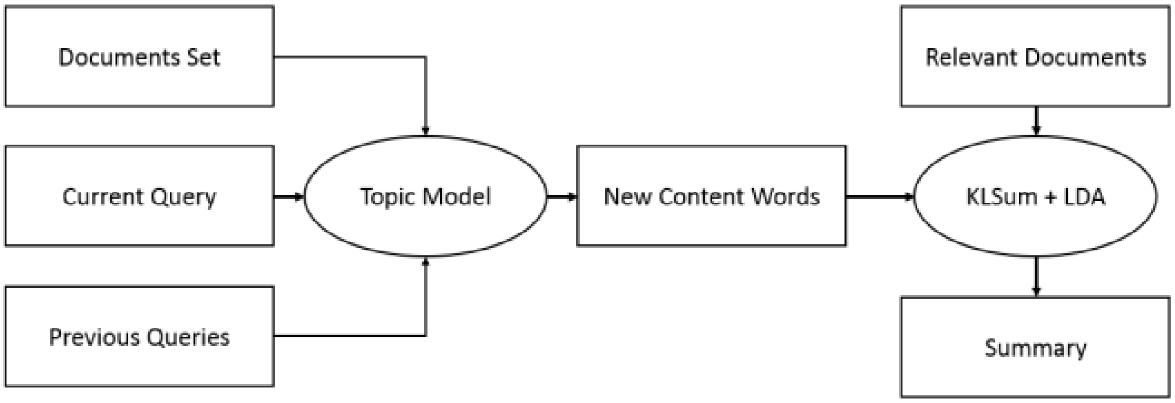

Figure 2: ChainSum architecture

should be assigned to term representing the previous topics in the chain.

5.4 Adapted LexRank

In LexRank, the algorithm creates a graph where nodes represent the sentences from the text and weighted edges represent the cosine-distance of each sentence's *TFxIDF* vectors. After creating the graph, PageRank is run to rank sentences. We adapted LexRank to QCFS in two main ways: we extend the sentence representation scheme to capture semantic information and refine the model of sentences similarity so that it captures query answering instead of centrality. We tagged each sentence with Wikipedia terms using the Illinois Wikifier (Ratinov, Roth, Downey and Anderson 2011) and with UMLS (Bodenreider 2004) terms using HealthTermFinder (Lipsky-Gorman, S. and N. Elhadad 2011). UMLS is a rich medical ontology, which is appropriate to the consumer health domain.
We changed the edges scoring formula to use the sum of Lexical Semantic Similarity (LSS) functions (Li, Irwin, Garcia, and Ram. 2007) on lexical terms, Wikipedia terms and UMLS terms:

$$\begin{aligned} Score(U,V) = LSS_{lexical}(U,V) + a \\ * LSS_{wiki}(U,V) + b \\ * LSS_{UMLS}(U,V) \end{aligned}$$

Where:

$$LSS(S_1,S_2) = \frac{\sum_i (MAX_j(\frac{Sim(W_i^1,W_j^2)}{Sim(W_i^1,W_i^1)})IDF(W_i^1))}{\sum_i IDF(W_i^1)}$$

Instead of using the cosine distance, in order to incorporate advanced word/term similarity functions. For lexical terms, we used the identity function, for Wikipedia term we used Wikiminer (Milne 2007), and for UMLS we used Ted Pedersen UMLS similarity function (McInnes and Pedersen 2009). Finally, instead of PageRank, we used SimRank (Haveliwala 2002) to identify the nodes most similar to the query node and not only the central sentences in the graph.

6 Evaluation

6.1 Evaluation Dataset

We worked on the dataset we created for QCFS and added semantic tags: 10% of the tokens had Wikipedia annotations and 33% had a UMLS annotation.

6.2 Results

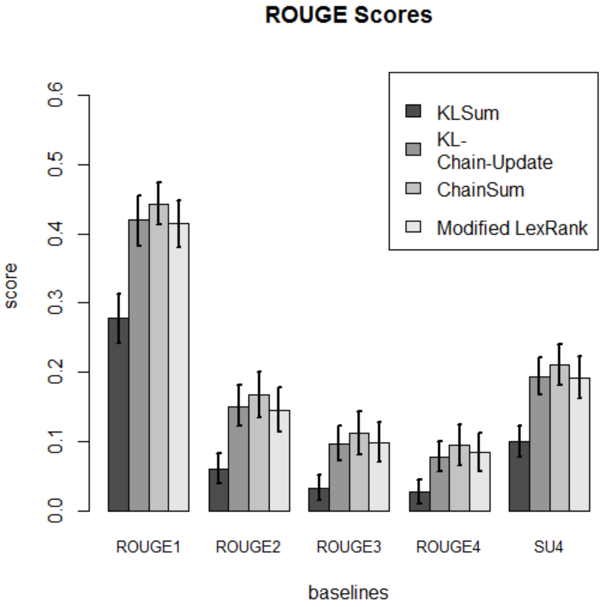

Figure 3: ROUGE scores.

For KLSum we received ROUGE scores of (r1 = 0.278, r2 = 0.060, su4 = 0.099), KL-Chain-Update (r1 = 0.419, r2 = 0.150, su4 = 0.193), ChainSum (r1 = 0.442, r2 = 0.167, su4 = 0.210), ChainSum with simplified topic model (r1 = 0.443, r2 = 0.159, su4 = 0.204) and for Modified-LexRank (r1 = 0.415, r2 = 0.144, su4 = 0.191). All of the modified versions of our algorithm performed better than plain KLSum with more than 95% confidence.

7 Conclusions

We presented a new summarization task tailored for the needs of exploratory search system. The user expects new information for every refinement of her query. This task combines elements of question answering by sentence extraction with those of update summarization.

The main contribution of this paper is the novel dataset containing human summaries. This dataset is annotated with Wikipedia and UMLS terms for over 30% of the tokens.

Four methods were evaluated for the task. The baseline methods based on KL-Sum show a significant improvement when penalizing redundancy with the previous summarization.

This paper concentrated on “zoom in” query chains, other user actions such as “zoom out” or “switch topic” were left to future work.

The task remains extremely challenging, and we hope the dataset availability will allow further research to refine our understanding of topic-sensitive summarization and redundancy control.

Acknowledgments

References

Marchionini, G. (2006). "Exploratory search: from finding to understanding." Commun. ACM **49**(4): 41-46.

White, R. W., & Roth, R. A. (2009). Exploratory search: Beyond the query-response paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services, 1(1), 1-98.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. Online Information Review, 13(5), 407-424.

Cline, R. J. W. and K. M. Haynes (2001). "Consumer health information seeking on the Internet: the state of the art." Health Education Research **16**(6): 671-692.

Islamaj Dogan, R., G. C. Murray, et al. (2009). "Understanding PubMed® user search behavior through log analysis." Database **2009**.

Erkan, G. and D. R. Radev (2004). "LexRank: Graph-based lexical centrality as salience in text summarization." J. Artif. Intell. Res. (JAIR) **22**: 457-479.

Haghighi, A. and L. Vanderwende (2009). Exploring content models for multi-document summarization. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics.

Nenkova, A., & Passonneau, R. J. (2004). Evaluating Content Selection in Summarization: The Pyramid Method. In HLT-NAACL (pp. 145-152).

Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop (pp. 74-81).

Conroy, J. M., Schlesinger, J. D., & O'Leary, D. P. (2011). Nouveau-rouge: A novelty metric for update summarization. Computational Linguistics, 37(1), 1-8.

Bodenreider, O. (2004). "The unified medical language system (UMLS): integrating biomedical terminology." Nucleic Acids Research **32**(Database Issue): D267.

Delort, J.-Y. and E. Alfonseca (2012). DualSum: a Topic-Model based approach for update summarization. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics.

Gupta, S., Nenkova, A., & Jurafsky, D. (2007, June). Measuring importance and query relevance in topic-focused multi-document summarization. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (pp. 193-196). Association for Computational Linguistics.

Daumé III, H., & Marcu, D. (2006, July). Bayesian query-focused summarization. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (pp. 305-312). Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. the Journal of machine Learning research, 3, 993-1022.

Ratinov, L. A., Roth, D., Downey, D., & Anderson, M. (2011, June). Local and Global Algorithms for Disambiguation to

Wikipedia. In ACL (Vol. 11, pp. 1375-1384).

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research, 32(suppl 1), D267-D270.

Lipsky-Gorman, S. and N. Elhadad (2011). ClinNote and HealthTermFinder: a pipeline for processing clinical notes. Columbia University Technical Report, Columbia University.

Li, B., Irwin, J., Garcia, E. V., & Ram, A. (2007, June). Machine learning based semantic inference: Experiments and Observations at RTE-3. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing (pp. 159-164). Association for Computational Linguistics.

McInnes, B. T., T. Pedersen, et al. (2009). UMLS-Interface and UMLS-Similarity: open source software for measuring paths and semantic similarity. AMIA Annual Symposium Proceedings, American Medical Informatics Association.

Milne, D. (2007, April). Computing semantic relatedness using wikipedia link structure. In Proceedings of the new zealand computer science research student conference.

Jeh, G., & Widom, J. (2002, July). SimRank: a measure of structural-context similarity. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 538-543). ACM.