Synopsis

on

Multi-Document Update and Opinion Summarization

In partial fulfillment of the requirement for
the degree of Masters of Technology
2007-2008

Kumar Puspesh
03CS3025

Under the guidance of,

Prof. Sudeshna Sarkar,
Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur

Multi-Document Update and Opinion Summarization

Kumar Puspesh
Indian Institute of Technology, Kharagpur

I. Introduction

AUTOMATED text summarization has drawn a lot of interest in the natural language processing and information retrieval communities in the recent years. The task of a text summarizer is to produce a synopsis of any document (or set of documents) submitted to it. The level of sophistication of a synopsis or a summary can vary from a simple list of isolated keywords that indicate the major content of the document(s), through a list of independent single sentences that together express the major content, to a coherent, fully planned and generated text that compresses the document(s). The more sophisticated a synopsis, the more effort it generally takes to produce. Several existing systems, including some Web browsers, claim to perform summarization. However, a cursory analysis of their output shows that their summaries are simply portions of the text, produced verbatim. While there is nothing wrong with such extracts, per se, the word 'summary' usually connotes something more, involving the fusion of various concepts of the text into a smaller number of concepts, to form an abstract. We define extracts as consisting wholly of portions extracted verbatim from the original (they may be single words or whole passages) and abstracts as consisting of novel phrasings describing the content of the original (which might be paraphrases or fully synthesized text). Generally, producing a summary requires stages of topic fusion and text generation not needed for extracts.

In addition to extracts and abstracts, summaries may differ in several other ways. Some of the major types of summary that have been identified include indicative (keywords indicating topics) vs. informative (content laden); generic (author's perspective) vs. query-oriented (user-specific); normal vs. update; background vs. just-the-news; single document vs. multi-document; neutral vs. evaluative. A full understanding of the major dimensions of variation, and the types of reasoning required to produce each of them, is still a matter of investigation. This makes the study of automated text summarization an exciting area in which to work. Now the area of Multi-document summarization can be seen further subdivided into various domains like - opinion summarization, update summarization, query-based summarization etc. Various search engines like Google, Yahoo etc. provide a short snippet alongwith each search result for any query given by the user. The automatic text summarization techniques are of great use in these real-world scenarios. Similarly, for any product there are numerous reviews available online and a summarized view of all those can be more inforamtive to the user in much lesser time. News, blogs and product reviews are some importatn sources of opinions, in general. Because queries may or may not be posed beforehand, detecting opinions is somewhat similar to the task of topic detection at sentence level. We try to look into automatic feature extraction mechanisms from product reviews and further opinion summarization techniques which retrieves relevant information from the document set, determines the polar orientation of each relevant sentence and finally summarizes the positive-negative sentences accordingly.

II. System Architecture and Methodology used

We worked on two independent systems - one for query and update summarization and another for opinion summarization. In the follwing paragraphs, we discuss about both the system architecture and modules. For update summarization, dataset used is the one used in DUC 2006. This dataset consists of 1250 documents (25 documets for each 50 topics). For opinion summarization we have collected data from amazon.com using our simple crawler. We collected 50 product reviews from amazon.com for 3 different product classes. For the evaluation purposes of opinion summarization, we used the Hu and Liu's data on feature extraction.

A. Update Summarization

We have used MEAD toolkit which provides the basic architecture above which different modules have been attached for different purposes. MEAD provides a simple interface and robust architecture where new modules can be added to rank and choose sentences from the set of documents. Here we will discuss various modules of the whole summarization system and the flow.

1) Preprocessing: We have used DUC 2006 data and reference summaries for our study. The documents are in a specific xml format. The preprocessor changes the format of the documents and modifies the document a little bit to remove some discrepancies. (The documents given by DUC 2006 are not well formatted as they have mistakenly grouped many sentences under the same tag which makes the system treat those as a single sentence only.)

2) Feature Scripts: Feature Scripts are the modules which compute values of various features of the set of sentences. As our system is based on sentence-based summarization algorithms, these modules essentially compute values of various features for each sentence present in the document pool. The feature values for each sentence present in a document are grouped together to form a feature vector for the document. MEAD provides a platform to add different feature vector computing scripts. It uses a three pass feature vector computation model - Cluster level, Document level and Sentence level. The first two levels are optional but computation of feature values at the last level is a must because this is the final step which gives scores to different sentences. Here are some of the Features used in our implementation of the summarizer -

1) **Length** Sentences having length less than the specified threshold are assumed to be non- relevant for the summarization purpose. The data on which we are working is a crawl of different news articles on same topic. So, it does contain some small phrases which are just a topic name or a bullet etc.
2) **Position** This feature is relevant in identifying important sentences as generally in any document, the sentences at the start of the paragraph or article are more important. Position feature assigns each sentence a value as,

$$P(s) = 1/n$$

   where n is the number of the sentence in the document.
3) **Centroid** A centroid is a set of words that are statistically important to a cluster of documents. As such, centroids could be used both to classify relevant documents and to identify salient sentences

in a cluster. The centroid of a cluster is a pseudo-document which consists of words that have tf*idf scores above a redefined threshold. Centroid is a feature which is dependent on the words present in the sentence. The more important words it contains, more central it is in respect of the document cluster. For computation of centroid feature, we first find out the term frequencies of various words present in the document. Then, for each word TF*IDF is computed where IDF is defined as,

$$IDF(i) = log(N/n_i) \tag{1}$$

Where, N is total number of documents and n_i is the number of documents in which the word i is present. Now, for each sentence C_i the combined centroid score is calculated as ,

$$C_i = \sum C_{w,i} \tag{2}$$

Where, $C_{w,i}$ is the TF*IDF score of the word w in the sentence i.

4) **Lexrank** It is inspired the PageRank algorithm used by Google for ranking of webpages across the world wide web. The basic task for any extractive summarization is finding the most central sentences from the cluster of documents - here it is done by finding the most prestigious sentences. (Also, Centrality of a sentence is calculated in terms of centralities of words that it contains). This approach is based on the concept of prestige in social networks, which has also inspired many ideas in computer networks and information retrieval. A cluster of documents can be viewed as a network of sentences that are related to each other. Some sentences are more similar to each other while some others may share only a little information with the rest of the sentences. We hypothesize that the sentences that are similar to many of the other sentences in a cluster are more central (or salient) to the topic. To define similarity, we use the bag-of-words model to represent each sentence as an N-dimensional vector, where N is the number of all possible words in the target language. For each word that occurs in a sentence, the value of the corresponding dimension in the vector representation of the sentence is the number of occurrences of the word in the sentence times the idf of the word. A cluster of documents may be represented by a cosine similarity matrix where each entry in the matrix is the similarity between the corresponding sentence pair. For computing prestige scores of different sentences -

   a) *Degree Centrality* Degree centrality may have a negative effect in the quality of the summaries in some cases where several unwanted sentences vote for each other and raise their centrality. As an extreme example, consider a noisy cluster where all the documents are related to each other, but only one of them is about a somewhat different topic. Obviously, we would not want any of the sentences in the unrelated document to be included in a generic summary of the cluster. However, suppose that the unrelated document contains some sentences that are very prestigious considering only the votes in that document. These sentences will get artificially high centrality scores by the local votes from a specific set of sentences.

   b) *Eigen Centrality* This situation can be avoided by considering where the votes come from and taking the centrality of the voting nodes into account in weighting each vote. A straightforward way of formulating this idea is to consider every node having a centrality value and distributing

this centrality to its neighbors. This formulation can be expressed by the equation

$$p(u) = \sum_{v \epsilon adj(u)} p(v)/deg(v) \tag{3}$$

where $p(u)$ is the centrality of node u, $adj(u)$ is the set of nodes that are adjacent to u, and $deg(v)$ is the degree of the node v. The above equation can be written equivalently as,

$$p = B^T por, p^T = Bp^T \tag{4}$$

where the matrix B is obtained from the adjacency matrix of the similarity graph by dividing each element by the corresponding row sum. This equation states that p^T is the left eigenvector of the matrix B with the corresponding eigen value of 1. The centrality vector p corresponds to the stationary distribution of B. However, we need to make sure that the similarity matrix is always irreducible and aperiodic.

3) Classifiers: This step merges the different feature vectors which were already computed in the last step. MEAD provides a default classifier which we have used here. It is user programmable in the sense that it allows us to assign different weights to different features. We have used different combination of features to study the quality of summary produced.

$$Score(Si) = feature1 * weight1 + feature2 * weight2 + ...$$

4) Rerankers: This step is used to remove redundancy from the extract or the summary. In Multi-document summarization, the documents may contain many sentences which talk about more or less the same thing i.e.; they may have similar information content present. In this case, we need to do some processing as we want to have as much information in the summary as possible. We need to filter out the sentences which are similar to each other across the documents. In this case, we are using a simple similarity feature to find out if the sentences which we are considering for the summary are similar to the already selected sentences or not. *Cross-sentence Informational Subsumption* (CSIS) : - It reflects that some sentences repeat the information present in other sentences and may, therefore, may be omitted during summarization. - If the information content of sentence a is contained within sentence b, then a becomes informationally redundant and the content of b is said to subsume that of a. e.g.

1. John Doe was found guilty of murder.

2. The court found John Doe guilty of the murder of Jane Doe last august and sentenced him to death.

The default reranker we have used in our system just sees the cosine similarity between already selected sentences in the summary and the new sentence which is under consideration right now. Novelty-reranker boosts the sentences which are close to the selected sentences as generally we have seen that sentences close to important sentences are also rich in their information content. We have used some other rerankers which are explained later in this document.

5) Post-Processing: Post processing involves several tasks like removing unnecessary phrases and words from the summary because generally the most important thing associated with summaries is the clustering of information with as little of unnecessary things as possible. So, we want to prune the sentences which were selected by the reranker to take out only the important parts of those in the summary/extract. This will allow us to include more sentences in the summary to increase the information content.

Example

Normal Summary

[1] In fact, Ruebush said, 90 percent of all malaria infections and 90 percent of malaria deaths occur in Africa. [2] Malaria, which is reaching epidemic proportions in Africa and parts of Asia, Latin America and the southern fringe of the former Soviet Union, kills about a million people a year, and children are especially vulnerable. Experts say one child dies of malaria every30 seconds. Around the world, malaria kills 3,000 children under 5every day, a higher mortality rate than AIDS. [3] "World spending on malaria control and research for Africa is maybe 10 cents per case per year," said Sachs. "It's quite dreadful. World Bank lending for malaria is de minimus. The big pharmaceutical companies see it as a disease of the very poor, so they never view it as much of an investment priority." [4] MANILA, November 26 (Xinhua) – The Philippines has made a big stride in malaria control with malaria infections rate in the country is now generally low, a senior health official said today.

Update Summary

[1] Malaria kills up to 3 million people a year and sickens another 300 million. Creating a vaccine is crucial because the parasite has begun developing resistance to drugs used to treat malaria, and even mosquitos that spread the disease are withstanding pesticides. [2] You may wonder why I would write a health column about malaria when there is no malaria in the United States. [3] This year, the health care service plans to promote health care awareness, teach people how to prevent malaria by themselves, help village medical stations to detect malaria patients, and provide mosquito-nets to all people in remote, isolated and mountainous areas. [4] It is estimated that 300 to 500 million clinical cases and 1.5to 2.7 million deaths occur due to malaria each year, about twice as many as 20 years ago, according to papers presented at the third Pan-African Conference on Malaria which ended here Wednesday.

B. Opinion Summarization

Opinion summarization summarizes opinions of articles by telling sentiment polarities, degree and correlated events. Here we discuss our system which tries to identify and analyze opininated sentences to generate a summary in some specific format. We have decomposed the problem of opinion summarization into following steps:

- Feature Extraction
- Opinion Identification
- Polarity Classification
- Summary Extraction

1) Feature Extraction: A set of good features/keyphrases (words or nominal compounds of great significance in a text) is a very important part as it works as an alternative representation for documents. Based on inforamtion theory (Shannon, 1948), the information content of a concept c is the negative log likelihood - $logp(c)$, where $p(c)$ is the probability of encountering an instance of concept c. As this probability increases, the informativeness decreases i.e.; a general concept is more frequent than a specific one over a large set of documents. The task of extracting keyphrases from a text consists of selecting salient words and

multi-word units, generally noun compunds no longer than a threshold, from an input document. Various automatic keyphrase extraction techniques have been discussed in literature e.g. (Turney, 1999) and systems like Extractor, Kea (Frank *et al.*, 1999; Witten *et al.*, 1999) and NPSeeker (Barker and Cornacchia, 2000).

Algorithm [Document level]

1) Identify sentence boundaries
2) **for** every sentence
3) tag each word in the sentence with its corresponding *part-of-speech*
4) find the tag patterns in the sentence
5) select the possible unigram features
6) select the possible multi-word features
7) remove stopwords and outliers
8) **if** external *feature-list* provided for this product class
9) filter the possible list of features to get a more precise list using the hierarchial feature information provided in the *feature-list*
10) **endif**
11) **for** each feature *f* extracted
12) **for** each discriminator *d* phrase
13) calculate the web PMI score as
14) $pmi(f,d) = hits(d+f)/hits(d) * hits(f)$
15) **end loop**
16) $pmi(f) = \max_d(pmi(f,d))$
17) **end loop**
18) rank the features according to PMI score and select the features above the threshold
19) **end loop**

Example

1. In my opinion it 's the best camera for the money if you 're looking for something that 's easy to use , small good for travel , and provides excellent , sharp images .

Extracted features : camera[camera], money[price], images[image]

2. the auto-mode is good enough for most shots but the 4300 also boasts 12 versatile scene modes as well as a manual mode though i admit i have n't played with it too much on manual .

Extracted features : scene modes[scene, mode], auto mode[mode], mode[mode]

3. awesome camera with huge print quality in a tiny package .

Extracted features : camera[camera],print quality[image]

Note: the features are represented as feat1[feat], where feat1 is a specific or specialized form of feat

2) Opinion Identification: The goal of opinion identification is to detect where in the documents opinions are embedded. An opinion sentence is the smallest complete semantic unit from which opinions can be extracted. The sentiment words, the opinion holders, and the contextual inforamtion should be considered as clues when extracting opinion sentences and determining their tendencies. As in the previous step we identify the feature terms or phrases of the document class, we use this extracted information to identify

the sentences which contain or might contain useful information about those features.
These sentences talk about the features of a digital camera and hence they are considered as subjective sentences :
1. i love the continuous shot mode , which allows you to take up to 16 pix in rapid succession – great for action shots .
2. yes , the picture quality and features which are too numerous to mention are unmatched for any camera in this price range .
3. there are so many functions in this little , yet powerful camera !

3) Polarity Classification: Sentiment Analysis or polarity classification is the task of identifying positive and negative opinions, emotions and evaluations. This step uses a list of words with known semantic orientation. These words are assigned their most common polarity and this works as the prior polarity for these words. At the first step a classifier just assumes that a word's polarity is same as its prior polarity and tries to classify the word as either neutral or polar (positive or negative). In the literature various observations have been made about these polar classified words - words with non-neutral polarity frequently appear in neutral contexts. Some times words occur in some other context also (probably with some different meaning) and sometimes modifiers deviate them from their prior semantic orientation. Hence we incorporate a second step which tries to classify the polar-marked words into positive, negative or neutral categories. For the classification step, various features e.g. word token, word part-of-speech, word prior polarity, whether word is preceded by some adjective or intensifiers, whether the word is strong subjective or weak subjective, whether the sentence contains any negation operator etc.

4) Summary Extraction: Traditional Summarization algorithms rely on the important facts of documents and remove the redundant information. Unlike the general techniques, two factors - say, the sentiment degree and the correlated events, play the major roles of opinions summarization. The repeated opinions of the same polarity cannot be dropped beacause they strengthen the sentiment degree. However, the redundant reasons of why they hold this position should be removed while producing the summaries. This step aims to produce a cross-document summary and at the previous step we know the opinionated sentences and the specific features they talk about, we can gather all the opinionated information from the corpus on a specific given topic. Two different types of summaries can be seen useful in case of product reviews - one where a query/topic is provided and the summary contains the opinionated sentences on that topic only and second, where a combined summary on all the different features of the product are summarized.

News and blog articles are also important sources of opinions. Generally speaking, news articles are more objective while blogs are usually more subjective. We have done some experiments on the TREC blog data as well to see the how this summarization model performs. A major differnece in summarization for product reviews and blogs/news comes at the subjectivity analysis phase. In reviews, subjectivity is found by identifying the features of the product - either independently or using an external ontology. Whereas, in case of blogs or news articles subjectivity finding step mailny relies on presence of opinion identification phrases. More results on these are discussed in the thesis.

III. Evaluation and Preliminary Results

There are various metrics present which can be used to evaluate summarization systems. We have used a version of **ROUGE** (Recall-oriented Understudy for Gisting Evaluation) for evaluating the summaries generated by our system. ROUGE is a N-gram based evaluation metric which can be used to measure the similarity between two summaries both - *precision-wise* and *recall-wise*.

$$C_n = \frac{\sum_{C \epsilon \{ModelUnits\}} \sum_{n-grams \epsilon C} Count_{match}(n-gram)}{\sum_{C \epsilon \{ModelUnits\}} \sum_{n-grams \epsilon C} Count(n-gram)} \tag{5}$$

Where C_n is the score of nth sentence, Count match (n-gram) is the number of n-grams matched between the peer summary and the reference summary where as Count(n-gram) is the number of n-grams present in each of the model-units. Model-units can be defined as sentences present in the model summary. This metric is a recall-based one. If the denominator is changed to consider the sentences present in the peer summary instead of the reference summary, it will become a precision-based metric.

$$p_n = \frac{\sum_{C \epsilon \{Candidates\}} \sum_{n-grams \epsilon C} Count_{clip}(n-gram)}{\sum_{C \epsilon \{Candidates\}} \sum_{n-grams \epsilon C} Count(n-gram)} \tag{6}$$

TABLE I
FOR COMBINED FEATURES - LEXRANK AND CENTROID

	Reference 1	Reference 2	Reference 3	Reference 4
Rouge-1 Recall	0.352	0.38	0.412	0.396
Rouge-1 Precision	0.424	0.525	0.548	0.553
Rouge-2 Recall	0.042	0.072	0.089	0.064
Rouge-2 Precision	0.0476	0.0762	0.0762	0.0856

We were able to do ROUGE evaluation only for multi-document summaries to evaluate different features that we incorporated in our framework. But, in case of update and opinion summaries, due to lack of reference summaries we weren't able to automatically evaluate the summaries produced. We have done some amount of manual evaluation for the update summaries and tried to evaluate the different modules of opinion summarizer separately. For example, the Feature Extraction module, when run on a set of 34 reviews for a single camera identified 562 features (174 unique features) whereas the test system of Hu and Liu had identified a total of 389 (115 unique features). Our Feature extraction clearly outperforms with a feature per sentence ratio of 1.624 against Hu and Liu's benchmark data which has 1.12 features per sentence on an average. More detailed evaluation on recall, precision and quality of these extracted features will be described later.

We are planning to particiapte in TAC 2008 and where we can evaluate and tune our system better.

References

[1] T. Wilson, J. Wiebe and P. Hoffman *Recognizing Contextual Polarity in Phrase-level Sentiment Analysis*, Proceedings of HLT/EMNLP 2005.

[2] Ku Lun-Wei, Liang Yu-Ting and Chen Hsin-Hsi *Opinion Extraction, Summarization and Tracking in News and Blog Corpora*, Proceedings of AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs.

[3] M. Jarmasz and C. Barriere,*Keyphrase Extraction : Enhancing Lists*, Proceedings of the Computational Linguistic in the North-East(CLINE) 2004.

[4] Ana-Maria Popescu and O. Etzioni, *Extracting Product Features and Opinions from Reviews*, Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 2005.

[5] Soo-Min Kim and E. Hovy, *Automatic Identification of Pro and Con Reasons in Online Reviews* Proceedings of COLING/ACL Poster Sessions, 2006.

[6] G. Erkan and D.R. Radev, *Lexrank:graph-based Lexical Centrality as Salience in Text Summarization*, Journal of artificial Intelligence Research 2004.

[7] G. Erkan and D.R. Radev, *LexPageRank - Prestige in Multi-document text summarization*, In the proceedings of EMNLP, 2004.

[8] L. Vanderwende, H. Suzuki, C. Brockett and A. Nenkova, *Beyond SumBasic - task focused summarization with sentence simplification and lexical expansion*, Information processing and management 2007.

[9] W. Yih, J. Goodman, L. Vanderwende and H. Suzuki, *Multi-document summarization by maximizing informative content-words*, In the proceedings of IJCAI 2007.