Leveraging Aging Theory in Multi-document Timeline Summarization

Chen Jie
Beijing Institute of Technology
5 South Zhongguancun Street
Haidian District, Beijing
sonyfe25cp@gmail.com

ABSTRACT

In order to summarize multiple documents about the same topic, it's often helpful to follow the development progress of them. We present a model based on aging theory which provides an intuitive way to show the life circle of the topic, complying with the habit of people understanding things. We summarize the documents within three steps. First, we reduce the influence of synonym and polysemy with the LSA method. Second, aging theory is introduced to represent the sentence with other four categories of features which are common used in this field. Third, we train a classifier model to recongized the summary sentence. Experiment results show that our method can improve the timeline summarization significantly.

1. INTRODUCTION

Important facts about the summary are those which descripe the development progress of the topic. When people create summaries, they choose the sentences that contain key words such as place, people, date and so on to form a short story about the topic. Understandably in automatic summarization as well, it's useful to use those key words to represent general facts and the important factors.

Everyday thousands of news reporting different events are published on the internet. There are lots of news services(e.g. Google News) have been developed to group news into events, and then produce a short summary for each event. However, most news reports are not prepared for the progress of event, lots of duplication message among reports about the same event. Topic-focused multi-document summarization aims to the main information from those topic-focused documents. The timeline summarization help to reorganize the order and sentences selection to get a better reading experience.

A particular challenge for multi-document summarization is how to computing the importance of each sentence, which is either depend on the words or some other latent information around the sentence. This required effective methods to analyze the information stored in different sentences. From the surface of different sentences, there are many synonym and polysemy words which bring lots of difficulties when computing relationships among sentences. latent semantic analysis (LSA) [**?**] is introduced to find the core words in a document without the influence caused by those words. LSA is a robust unsupervised technique for deriving an implicit representation of text semantics based on observed co-occurrence of words to find semantic units of news.

Another challenge for multi-document summarization is how to deal with duplicate information among these documents around the same topic or event. Reporters usually like to review something happened and then tell the readers what happens now. It's helpful for readers to know the development progress about the topic and it's also a solution to this challenge. Event goes through a life cycle of birth, growth, maturity and death, and this can be reflected from special terms utilized for descripting different events experience a similar life cycle. Aging theory [**?**] is a model exploited in event detection task which tracks life cycles of events using energy function. The energy of an event increases when the event becomes popular, and it diminishes with time. In intuition, it can also be used for summarization to help us find out the daily hot terms of events.

In this paper, we generate timeline summary by considering both temporal and semantic characteristics from the news around the same event. We extract the features from five aspects to represent each sentence. Then, classification model is built with logistic regression. Last, we choose sentences from candidates to form the summary and display them with timeline, so that people can track the progress of event easily and quickly.

The remainder of this paper is organized as follows: Section 2 reviews related works on summarization. We discuss our approach about how to leverage aging theory to gain sentence feature and train the logistic regression classification model in section 3. Experiments and discusses are described in section 4. Section 5 presents our conclusions and future plans.

2. RELATED WORK

2.1 Multi-Document Summarization

Many summarization methods about multi-document have been developed recently. Generally speaking, these methods can be divided into extractive summarization or abstractive summarization. The main idea of extractive summarization is to assign scores to different words in each sentence, then those sentences with high scores will be choosed as the summary. While abstractive summarization usually do some information fusion[**?**], compression[**?**] and reformulation[**?**] to get the summary sentences. In this paper, we focus on the former method.

One of the most popular extractive multi-document summarization methods is MEAD[**?**], which represents each sentence with some sentence-leve features such as term frequency, sentence position, first-sentence overlap, etc. Wan[**?**] proposed an extractive approach based on manifold-ranking about the information richness and novelty. Wan[**?**] used markov random walk model and cluster hits model to analysis the link relationships between sentences in order to gain the important one as the summary. Wong[**?**] investigated co-training method by combining labeled and unlabeled data to train the model, which used four kinds of features can be categorized as surface, content, relevance and event features.

Most recently, the multi-document timeline summarization gains enough attraction from researchers and engineers. The timeline can help readers to know the development progress of the event. This requires the redundancy of summary should be very low and the key properties of event should be retained. Lots of timeline summarization methods and applications have been developed recently. ETS[**?**] formulated the task as an optimization problem via iterative substitution from a set of sentences with four requirements. [**?**] investigated five different sentence features and leveraged SVMRank to optimize the summarization task. [**?**] took social attention involved to compute the importance. Yan[**?**] proposed to model trans-temporal correlations among component summaries for timelines, using inter-date and intra-date sentence dependencies. [**?**] extracted the temporal information and surface features to train the regression model for predicting the summary sentences. [**?**] reused the MEAD and add the timestamp feature to implement their TS. Binh Tran[**?**] proposed a framework for automatically constructing timeline summaries from web news articles. In their framework, they extracted some features for each date about articles and sentences' published time and reference time.

2.2 Aging theory

Aging theory. has been proved effictive to track which stage of life cycle for news. [**?**] [**?**] applied this to model the news event's life cycle and utilized the concept of energy to track it. In order to gain the summary of multi-documents of news domain, we consider aging theory is worth using to extract the feature of sentence.

3. OUR APPROACH

3.1 Key Concepts

Topic-focused. : What we value most is an event grouped from several web news articles, such as the "the missing of the malaysia airlines plane" from BBC. These articles show us the cause, the progress and the results about the event.

Timeline Summaries:. Generally speaking, timeline is a kind of display forms for the summaries. Timeline summaries should show us the progress of this topic instand of just displaying the message according to the time sequence. Under this condition of the requirement, timeline summary of each day should describe the most important thing happened in that day.

We give the formal definition of multi-document timeline summarization as follows:

Input. : Given a set of documents $D = \{d_1, d_2, \ldots, d_n\}$ which should cover progress of the topic in the time span $T = \{t_1, t_2, \ldots, t_m\}$. We segment each document to sentences and group them by the date to form sentences $S = \{s_1, s_2, \ldots, s_m\}$.

Output. : The multi-document timeline summarization should output the summaries along the date and each summary is the main idea of what occured in that day, i.e. $O = \{o_1, o_2, \ldots, o_m\}$, where o_i means the summary of sentences from all the sentences of that day s_i.

3.2 Sentence Feature Selection

In order to respresent the most important thing happened in that day, the summary should consider the importance, the movelty, and contains the topic hot terms. Five kinds of features are chosen as follows:

Surface feature. : This contains features computed by basic statistics, such as the length of sentence, the counts of noun words and stop words, the position in this document and paragraph, and whether it contains person name or not.

Importance feature. : This feature aims to respesent the importance of the sentence. The weight of sentence is computed through linear combination of term weights with latent semantic analysis. The function is

$$Weight_{importance} = \sum_{w \in sentence} TF_w * LSA_w \qquad (1)$$

where TF_w is the term frequency of word w and LSA_w is the weight of word in the LSA results.

Aging feature. : We use this feature to show the life cycle of the sentence. The frequency of a word will change as event going on, so we use the association between word and time interval to indicate its energy which is defined as follows:

$$E_{w,t} = F(F^{-1}E_{w,t-1} + \alpha \cdot \chi^2_{w,t}) \qquad (2)$$

where $E_{w,t}$ is the energy of word w in time interval t , and $E_{w,t-1}$ is the energy of word w in time interval $t-1$, α is the transfer factor, and $\chi^2_{w,t}$ is the contribution degree of word

at the time interval t, which can be computed as presented in [**?**].

However, no words descripting a special event point will retain popular forever, they will decay over time. In order to represent the word's life span realistically, we cut down the energy of word by a decay factor Îš at the end of every time interval. And if the decayed energy value became negative, we change it to 0.

According to the description above, if the energies of some words increase greatly, we can draw a conclusion that there is a hot event spot. So we need to calculate the variance of word energy next. Here we use standard deviation:

$$Var_{w,t} = \sqrt{\frac{1}{N} \sum_{t \in period} (E_{w,t} - \overline{E_w})^2} \quad (3)$$

where N is the number of time intervals during the given period, $E_{w,t}$ is the energy of word w in time interval t , $\overline{E_w}$ is the average energy during the period, and $Var_{w,t}$ is the variance of w . Then each word will be assigned a new weight besides the traditional TF.IDF which can be defined as:

$$Weight_{aging} = \sum_{w \in sentence_i} TF * IDF_w + \mu \cdot Var_w \quad (4)$$

This kind of new weight can help us identify both central and hot information, so people can capture the main line and new changes of events simultaneously.

Topic feature. : Each topic contains lots of sentences, that is, every sentence contribute some information to the whole set to express the topic. In this study, we use the link analysis to compute the latent semantic between sentences. First, topic terms and topic elements can be found through the word frequency analysis. Then, the similarity among sentences are computed with the cosine function to construct the event map. Last, pageRank algorithm is used to assigin the weight to each node in this map. We treat the pageRank value as the topic feature of the sentence. The function is :

$$Weight_{topic} = PageRank(sentence_i) \quad (5)$$

Novelty feature. : In order to avoid some sentences with the same meanings be selected as the summary, the novelty of sentence is important. The novelty value is compute by the distance with the summary of last time span. The larger the distance is, the more the novelty this sentence is. In our research, we use the Jaccard similarity to gain this. The function is :

$$Weight_{novelty} = 1 - Jaccard(sentence_i, summary_{ex}) \quad (6)$$

where $summary_{ex}$ is the summary sentence in the last time span.

All the features are used in our experiments are shown in table 1.

3.3 Model Training

Table 1: List of features and their category

Category	Feature
surface	Length
surface	the count of noun words
surface	the count of stop words
surface	position in current paragraph
surface	position in the document
surface	whether it contains person name
Importance	sum of LSA scores
Importance	sum/avg TFIDF
Topic	the count of topic words
Topic	sum of topic words' weight
Topic	sum of the TF top 10 words' weight
Aging	the aging score of sentence
Novelty	distance between current sentence and summary

With the help of labled data, we considered this summarization task as a classification problem. The positive data is sentences labeled to summary, otherwise is negative.

However, in each document only a few sentences will be labled as the summarization sentence, that is, the data set is imbalanced. When we create a classifier over this dataset, the classifier will perfer the major side [**?**]. In order to impove the precision of minority class, we used SMOTEBoost[**?**] method to sampling datas for training the classification model. This method combines SMOTE [**?**] and boost technology and has been proved effective for imbalanced data set.

4. EXPERIMENTS

4.1 Experimental settings

For evaluate the performance of our methd, we used the dataset of TAC 2010 summary task, whic is an open benchmark dataset published by Text Analysis Conference[1]. This dataset contains 906 documents around 46 topics from the New York Times, the Associated Press, the Xinhua News Agency newswires, and the Washington Post News Service. Each topic has two sets of documents, A and B, each containing 10 documents. The difference is that documents in B were published later than A.

4.2 Baselines

We start our experiment with some preprocessings like indexing, filtering out the stop words and segmenting news documents into sentences. Then we perform our method to the data set and generate a timeline for each event we choosing. We implement some widely used multi-document summarization methods as the baselines.

Centroid extracts sentences based on centroid value, positional value and first-sentence overlap.

Cluster considers that there are different themes in an event, so it first clusters similar sentences together into different clusters and then selects one representative sentence from each main cluster.

Allan is a similar timeline system from different aspects,

[1]http://www.nist.gov/tac/

Table 2: Performance on manual labeled data set

Method	Precision(1)	Recall(1)	Precision(3)	Recall(3)
Centroid	0.129	0.076	0.228	0.129
Cluster	0.057	0.045	0.185	0.133
Allan	0.143	0.083	0.232	0.184
Wong	0.214	0.085	0.363	0.135
L2RTS	0.221	0.095	0.394	0.157
TMTS	0.233	0.090	0.391	0.159

which dividing sentences into on-event and off-event while ranking them with useful and novelty.

Wong combined the supervised and semi-supervised learning and used co-training method to train the labeled data and unlabeled data.

L2RTS considered the summary task as optimistic task and leveaged the SVMRank to gain the summary.

4.3 Evaluation metric

Here we use ROUGE toolkit [**?**] , which is officially applied by Document Understanding Conference (DUC)[2] for document summarization performance evaluation, to evaluate the experimental results and compare these algorithms with each other. The summarization quality is measured by counting the number of overlapping units, such as N-gram, word sequences, and word pairs between the auto-generated summary and the manual summary. Several automatic evaluation methods are implemented in ROUGE, such as ROUGE-N, ROUGE-L and ROUGE-W, each of which can generate two scores (recall (R), precision (P)). The function is:

$$R = \frac{\sum_{s \in manual} \sum_{N-gram \in s} Count_{match}(N-gram)}{\sum_{s \in manual} \sum_{N-gram \in s} Count(N-gram)} \quad (7)$$

$$P = \frac{\sum_{s \in auto} \sum_{N-gram \in s} Count_{match}(N-gram)}{\sum_{s \in auto} \sum_{N-gram \in s} Count(N-gram)} \quad (8)$$

Where N stands for the length of the $N-gram$, $Count_{match}(N-gram)$ is the maximum number of N-grams co-occurring in the auto-generated summary and the manual summary. For all the training data are labeled data, the precision and recall rate can be computed easily, these two metrics are helpful to get a quick understanding about the performance.

In order to evaluate the performance, we design two experiments, respectively is top-1 and top-3. Top-1 means each day we only choose one sentence as the summary, relatively top-3 means three sentences are choosed as the summary.

From the experiment results, we can get the information that summarization methods based on machine learning are performance better than others. The result of *Centroid* are better than *Cluster*, mainly because this method use some surface feature. The *Cluster* method is the worst in our experiments, since this method cluster the same meaning sentences and choose one from cluster, which makes the result ignore the novelty. The *Allan* method's performance better than *Centroid* cause this method consider the novelty and importance. Our method $TMTS$ performed better than $Wong$ and $L2RTS$ on Top-1 summary mainly because we use the SMOTE method to train the model, which makes the minorty class can be classified better. But when testing in Top-3, the $L2RTS$ wins the match, because the SVM-Ranking can obtain a better classification model.

[2]http://duc.nist.gov/

5. CONCLUSION AND FUTURE

In this paper, we present a novel approach of which involved aging theory and SMOTEBoost to timeline summarization. In our approach, we firstly construct features of each sentence, which contains surface feature, importance feature, aging feature, novelty feature and topic feature. Then we treate the multi-document summarization task as pair-wise classification task and generate the training data. At last SMOTEBoost is used to train the model. Experiment results show that our approach performs better compared with other widely used methods.

In the future, we will identify semantic units using other methods since LSA can process synonym but is unable to handle polysemy. And we will also extend our approach to short text such as microblogs and comments.