BJUT at TREC 2015 Temporal Summarization Track

Yingzhe Yao[1,2,3], **Zhen Yang**[1,2,3,4*], **Kefeng Fan**[1,4]
1. College of Computer Science, Beijing University of Technology, Beijing 100124, China
2. Beijing Key Laboratory of Trusted Computing, Beijing 100124, China
3. National Engineering Laboratory for CTISCP, Beijing 100124, China
4. Guangxi Colleges and Universities Key Laboratory of cloud computing and complex systems,
Guilin University of Electronic Technology, Guilin 541004, China
5. China Electronics Standardization Institue, Beijing 100007, China
∗yangzhen@bjut.edu.cn

Abstract

In this paper, we describe our efforts for TREC Temporal Summarization Track 2015. Since this is the third time we participate in this track,we adopt a different novel method NMFR to solve the newly emerging temporal summarization problem. Our goal of this year is to evaluate the effectiveness of : (1) Considering the semantic structure of document and the manifold structure of document could be as possible to preserve the essential characteristic of the original high-dimensional space of documents during the process of feature reduction.(2)Using the non-negative matrix factorization with our semantic and manifold regularization for summarization is effective and comparable to Affinity Propagation. Finally, we conduct experiments to verify the proposed framework NMFR on TREC Temporal Summarization Track Corpus, and, as would be expected, the results demonstrate its generality and superior performance.

Introduction

The TREC Temporal Summarization Track runs for the third time in this year, and its goal is to develop systems which can detect useful, new, and timely sentence-length updates about a developing event. According to the three different corpus, there are three tasks:

- **Task 1: Filtering and Summarization**
  Participants will be provided high-volume streams of news articles and blog posts crawled from the Web (TREC-TS-2015 a.k.a. KBA Stream Corpus 2014).Each participant will need to process those streams in time order, filter out irrelevant content and then select sentences from those documents to return to the user as updates describing each event over time.
- **Task 2:Pre-Filtered Summarization**
  Participants will be provided pre-filtered high-volume streams of news articles and blog posts crawled from the Web for a set of events (TREC-TS-2015F).Each participant will need to process those streams in time order, filter out irrelevant content and then select sentences from those documents to return to the user as updates describing each event over time.
- **Task 3:Summarization Only**
  Participants will be provided low-volume streams of on-topic documents for a set of events (TREC-TS-2015F-RelOnly).Each participant will need to process those streams in time order selecting sentences from the documents contained within each stream to return the user as updates over time.

In this year's track, we participate in the Pre-Filtered Summarization task using our proposed framework, first pre-processing and filtering TREC-TS-2015F, then summarization, at last post-processing. In order to verify the ability to summarization without filtering, we also have done the Summarization Only task with TREC-TS-2015F-RelOnly that contains on-topic documents. The corpus consists of a set of time stamped documents from a variety of news and social media sources, each with a unique identifier.

Our method (NMFR) is a novel document partitioning method based on the non-negative factorization of the term-document matrix of the given document corpus. We consider the pairwise sample similarity by a predefined similarity matrix K both from text semantic structure and sample neighboring relations as the regularization terms of NMF. The matrix K can be constructed either by using the label information in supervised learning or using certain distance metrics in unsupervised learning. Hence, K essentially encodes the class information or the intrinsic structure of data. Experiment results and TREC TS results show our method is effective.

System Framework

Figure 1 shows our system framework. It mainly consists of five parts: (1) Preprocess and index module, (2) Information Retrieval module, (3) Information filtering and text vectorization, (4) Clustering and Summarization module, and (5) Post-processing module.

Pre-process and index modules

The corpus downloaded locally is encrypted file, which cannot be used directly. In this sense, First step is to decrypt the files using the authorized key from authority, converting the GPG file format to SC file format. Then we use stream corpus toolbox to parse these SC files to TXT files. The stream corpus toolbox is given by TREC and provides a common data interchange format for document processing pipelines that apply language processing tools to large streams of text.

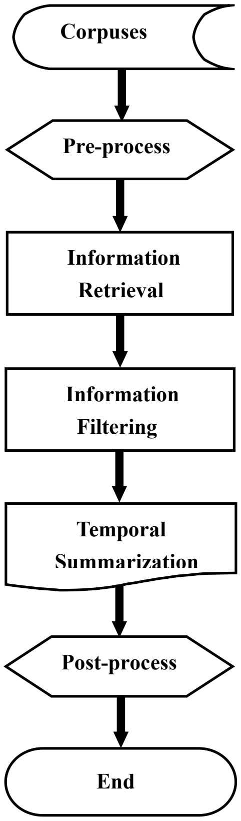

Figure 1: The Framework of System.

The last step is to build index by lemur for the information retrieval module.

Information Retrieval module

In this part, we use Lemur for information indexing and retrieval. Lemur is a toolkit designed to facilitate research in language modeling and information retrieval (IR). It supports the construction of basic text retrieval systems using language modeling methods such as BM25. Our experiment has two steps to build the index. First, create a parameter file to tell the lemur toolkit how to index; Secondly, use IndriBuildIndex.exe application to build index. Accordingly, the realization of retrieval also has two steps. First, create a parameter file to tell the lemur toolkit how to retrieve. Secondly, use IndriRunquery.exe application to retrieve.

In this way, for a given topic, we get a ranked sentence list by its relevance to its topic query words. In fact, we may use query expansion to increase the number of relevant sentences by using words with similar meaning to those in the query to solve the word mismatch problem, because people often describe the same concepts between the queries and documents.

Information filtering and text vectorization

After IR module, we get a set of sentences related to a topic. Considering the effectiveness of these sentences, we first judge whether these sentences are effective: they should be not only in the range of each topic's begin time and start time ,but also should be not repeated sentences and contain not less than three words in that too short sentences is impossible to describes any information . If a sentence is not effective, abandon it, and at last the rest ones should be treated as candidate sentences. Considering the large amount of these sentences, in order to simplify the computation, we going on another filtering by retaining the candidate sentences whose relevance score are bigger than a given threshold value.

As is known to all, the Vector Space Model is to simplify the handling of the text content for vector operations of vector space, intuitive and easy to understand. When the document is represented as the document vector of the space, and then we can calculate vector similarity of space on it to express the semantic similarity. In the processing of text documents, Commonly used method to quantify term weight is TF - IDF. In this method, the value is proportionally to the number of times of which a word appears in the document, but is offset by the frequency of the word in the corpus, which else to control the fact that some words are generally more common than others. Of course, In order to further reduce the dimension and improve the accuracy of text representation, we must first going on stop words filtering and web fixed format filtering, such as web page link, finally calculate the effective term weight.

Clustering and summarization module

After getting the vectors, the VSM similarity between two documents can be calculated by using the cosine distance:

$$sim(d_i, d_j) = \frac{d_i \cdot d_j}{||d_i|| \cdot ||d_j||} \quad (1)$$

where d_i and d_j are two vectors representing two different documents and $||d||$ is the length of the vector d. Another method to calculate similarity is based on mutual information preserving mapping[1], which is a manifold learning algorithm that computes low-dimensional, neighboring-preserving based on mutual information of high-dimensional inputs.

Multi-document summarization (MDS) approaches takes as input a set of documents about a topic to be summarized and produce a summary of these documents. One of the most widely used approaches to score sentences for inclusion into a summary is clustering with respect to the centroid of the sentences within the input documents[3],thereby selecting those sentences most central to the topic first. In this experiment, we adopt an improved non-negative matrix factorization with similarity preserving feature regularization (NMFR) as clustering summarization technique. There are two prominent advantages: first, comparing with the LSA, its decomposition results have good interpretability owing to its non-negativity constraint; second, different from traditional clustering method needing feature dimension reduction ahead of time, NMF not only can realize clustering but also complete feature dimension reduction at the same time. Moreover, we can add various regularization constraints to restrain dimension reduction and factorization process, thus preserving most important original characteristics of high dimensional space.

Of course, to contrast, we also used another classical clustering algorithms: Affinity Propagation (AP). Compared with the existing clustering algorithms, such as K-center clustering, AP is an efficient and fast clustering algorithm for large datasets without specifying beforehand clustering number which clusters data, taking a set of real-valued pairwise data point similarities as input.

Low-rank matrix factorization method is widely employed in various applications such as document clustering [4,5] and collective filtering [6,7]. Non-negative matrix factorization is a linear, non-negative approximate data representation. Let's assume that our data sets consists of N samples of m non-negative scalar variables. Denoting the (m-dimensional) measurement vectors ($t = 1, \cdots, N$), a linear approximation of the data sample is given by

$$x^t \approx \sum_{i=1}^{k} w_i h_i^t = Wh^t \tag{2}$$

Where W is an $m \times k$ matrix containing the basis vectors as its columns. Note that Note that each sample vector is written in terms of the same basis vectors. The k basis vectors can be thought of as the building blocks of the data, and the (k-dimensional) coefficient vector describes how strongly each building block is present in the sample vector .Arranging the sample vectors into the columns of a matrix X, we can now write

$$X \approx WH \tag{3}$$

where each column of H contains the coefficient vector corresponding to the sample vector. Written in this form, it becomes apparent that a linear data representation is simply a factorization of the data matrix.

Given a data matrix X, the optimal choice of matrices W and H are defined to be those nonnegative matrices that minimize the reconstruction error between X and WH. Various error functions have been proposed (Paatero and Tapper, 1994; Lee and Seung, 2001), perhaps the most widely used is the squared error (Euclidean distance) function .

$$E(W, H) = ||X - WH||^2 = \sum_{i,j}(V_{i,j} - (WH)_{i,j})^2 \tag{4}$$

where $(X)_{i,j}$ represents an element of a matrix X. There is neighboring relationship among text data points in term of distance, accordingly, there is also semantic approximation relationship from the semantic aspect. The former is a consideration by the manifold structure (the geometric distribution) of data points while the latter is text semantic relations distribution of data points. In the process of feature selection, we hope to make the low dimensional space as much as possible to retain the intrinsic character of original high dimension space of VSM. So we use text vector cosine similarity and improved mutual information semantic similarity calculation formula to compute the pairwise similarity matrix K, as is shown in formula (5).

$$K = \lambda X^T X + (1 - \lambda)Y \tag{5}$$

Where X is data matrix, λ is a tuning parameters ranging from zero and one, controlling the share of two items in the matrix K, the first item is represented as geometric similarity matrix between data points and Y is our calculated semantic similarity matrix between points based on word co-occurrence model, mainly word frequency and document frequency of word.

Considering the control of model complexity and the pairwise similarity matrix, adding two regularization, we proposed our method NMFR, based on matrix factorization while exploiting the pairwise similarity among data points. NMFR is to solve the following optimization problem,

$$\min_{W,H} F = ||X - WH||_F^2 + \alpha||WHH^TW^T - K||_F^2 + \beta||W||_F^2 \tag{6}$$

By removing constants in the objective function, the above equation can be rewritten as,

$$\begin{aligned} &F = Tr(-2X^TWH + H^TW^TWH) + \\ &\lambda Tr(WHH^TW^TWHH^TW^T - WHH^TW^TK - \\ &K^TWHH^TW^T) + \beta Tr(WW^T) \end{aligned} \tag{7}$$

The coupling between W and H makes the problem in Eq. (5) difficult to find optimal solutions for both W and H simultaneously. In this work we use an alternative optimization scheme[5]. Reference to the paper, it is easy to solve the objective function. To save space, we omit it here.

Post-processing module

After topic clustering, we select each topic clustering center as the final summary sentence (if this step we use the MMR (Maximal Marginal Relevance method) may improve the accuracy of the results, but the calculation will be a big.)Finally, we sort the sentence according to the correlation and time factor, forming the final clustering results.

Experimental Results

There are two parts of the results in our temporal summarization works: Pre-Filtered Summarization result and Summarization Only result.

Evaluation Methods

According the TREC authority, there are three metrics:

- Expected Gain. One way to evaluate an update system is to measure the expected gain for a system update. This is similar to traditional notions of precision in information retrieval evaluation.
- Comprehensiveness. Similar to tradition notions of recall in information retrieval evaluation.
- F measure. In order to summarize expected gain and comprehensiveness, we use a F measure based on both Expected Gain and Comprehensiveness.

Table 1: The Results of Pre-Filtered Summarization.

		nE[Latency Gain]			Latency Comp.			HM(nE[LG],Lat. Comp.)		
		L1AP1	L1NMF2	AVG	L1AP1	L1NMF2	AVG	L1AP1	L1NMF2	AVG
Topic	26	0.0174	0.0176	0.0444	0.2733	0.2658	0.2758	0.0328	0.0330	0.0667
	27	0.0155	0.0172	0.0296	0.2696	0.2883	0.2540	0.0293	0.0324	0.0426
	28	0.0074	0.0063	0.0246	0.2003	0.1626	0.2394	0.0142	0.0121	0.0390
	29	0.0469	0.0511	0.0981	0.3283	0.3117	0.1717	0.0821	0.0877	0.0884
	30	0.0282	0.0299	0.0545	0.2534	0.2551	0.1930	0.0507	0.0535	0.0694
	31	0.0332	0.0539	0.0743	0.2386	0.3492	0.2419	0.0582	0.0933	0.1058
	32	0.0168	0.0085	0.0531	0.0368	0.0189	0.0860	0.0231	0.0118	0.0594
	33	0.0228	0.0232	0.0709	0.3317	0.3328	0.2209	0.0426	0.0434	0.0776
	34	0.0195	0.0137	0.0470	0.4221	0.2798	0.2804	0.0372	0.0261	0.0688
	35	0.0032	0.0040	0.0276	0.1412	0.1791	0.2470	0.0063	0.0078	0.0440
	36	0.0130	0.0123	0.0173	0.4590	0.4195	0.2163	0.0252	0.0239	0.0304
	37	0.0219	0.0241	0.0281	0.2589	0.2665	0.1637	0.0404	0.0443	0.0429
	38	0.0113	0.0113	0.0725	0.1722	0.1566	0.2487	0.0211	0.0211	0.0722
	39	0.0089	0.0090	0.0701	0.3662	0.3649	0.3842	0.0175	0.0176	0.0701
	40	0.0091	0.0090	0.0166	0.4678	0.4137	0.2757	0.0178	0.0177	0.0296
	41	0.0199	0.0235	0.0333	0.4683	0.4874	0.3128	0.0382	0.0449	0.0532
	42	0.0136	0.0149	0.0290	0.2971	0.2912	0.3473	0.0260	0.0283	0.0470
	43	0.0120	0.0189	0.0534	0.3309	0.4792	0.2825	0.0232	0.0364	0.0755
	44	0.0162	0.0179	0.0802	0.3227	0.3297	0.2606	0.0309	0.0340	0.0896
	45	0.0176	0.0126	0.0667	0.3014	0.1816	0.2365	0.0333	0.0236	0.0917
	46	0.0073	0.0075	0.0234	0.1251	0.1251	0.0607	0.0138	0.0141	0.0214
Mean	ALL	0.0483			0.2381			0.0612		
	L1AP1	0.0172			0.2888			0.0316		
	L1NMF2	0.0184			0.2838			0.0337		

Results

There are two parts of the results in our temporal summarization works: Pre-Filtered Summarization results and Summarization Only results.

Table 1 shows the Pre-Filtered Summarization results of our system. In the first line, nE[Latency Gain] signifies the scores of the expected latency-adjusted gain, Latency Comp. signifies the scores of the latency-adjusted comprehensiveness, and HM(nE[LG], Lat.Comp.) signifies the scores of the harmonic mean of the two latency-adjusted measures.This last measure is the primary measure for the track. In the second row, DMSL1AP1 and DMSL1NMF2 (Omit prefix 'NMF' in table1 for brevity) is the runs we submitted, AVG is the mean score for each topic over all pooled runs submitted to the track. In the first column, the meaning of per-topic is obviously, Mean signifies the average values of the scores over the 21 topics are given for each run. In the second column, ALL signifies the mean score over all topics and all pooled runs submitted to the track.The same is with the Table 2, showing the Results of Summarization Only task.

Comparing the results of Table 1 and Table 2, we can find it that for the Results of Summarization Only task, in the term of the three metrics, our system is mostly better than the average performance of the all the participating systems, while for the Pre-Filtered Summarization task,there is better only in the metric of the latency-adjusted comprehensiveness. This indicates that our system is not good at filtering task, and facing too many unrelated documents of corpus is a nightmare for summarization, at the same time, it demonstrates our system is very suitable for summaring with on-topic corpus.

For each task,we both use our NMFR method and the baseline method - AP method. From two tables,we can see the results of them are similar. Now let's take Table 2 for example.Through Table 2, the performance of DMSL2A1 and DMSL2N2 with respect to the metrics latency-adjusted Expected Gain, the latency-adjusted comprehensiveness and the harmonic mean F measure are mostly better than AVG, which means that our methods are effectively. However, there are several topics whose metric value is smaller than the AVG, which means that our methods are not so well in stability. Through the contrast of the last three lines, we come to the conclusion that our run's performance is better than the average.

Conclusion

In this paper, we presented the implementation details of our runs for Temporal Summarization Track, and our system is fit for summarization the on-topic corpus but not does well in summaring the corpus with lots of off-topic content. Through the experiment results, we find our NMFR method is effective and comparable to AP method. This indicates considering the semantic structure of document and the manifold structure of document could be as possible to preserve the essential characteristic of the original high-

Table 2: The Results of Summarization Only.

		nE[Latency Gain]			Latency Comp.			HM(nE[LG],Lat. Comp.)		
		L2A1	L2N2	AVG	L2A1	L2N2	AVG	L2A1	L2N2	AVG
Topic	26	0.0253	0.0204	0.0213	0.4753	0.3533	0.3810	0.0480	0.0386	0.0358
	27	0.0224	0.0146	0.0274	0.3860	0.2434	0.3685	0.0424	0.0275	0.0459
	28	0.0254	0.0151	0.0099	0.3561	0.1592	0.1834	0.0475	0.0276	0.0186
	29	0.0296	0.0341	0.0205	0.2235	0.2517	0.1837	0.0523	0.0601	0.0359
	30	0.0441	0.0354	0.0288	0.3417	0.2673	0.2620	0.0781	0.0626	0.0458
	31	0.0845	0.0861	0.0326	0.5820	0.5706	0.3298	0.1475	0.1496	0.0514
	32	0.0169	0.0183	0.0240	0.0395	0.0416	0.1170	0.0236	0.0254	0.0358
	33	0.0285	0.0291	0.0141	0.3992	0.3752	0.2852	0.0531	0.0541	0.0250
	34	0.0284	0.0278	0.0211	0.3678	0.3678	0.3753	0.0527	0.0517	0.0385
	35	0.0153	0.0105	0.0134	0.5725	0.3878	0.3927	0.0297	0.0205	0.0251
	36	0.0111	0.0096	0.0164	0.4839	0.4237	0.3737	0.0217	0.0189	0.0228
	37	0.1075	0.1234	0.0444	0.5564	0.6152	0.2583	0.1801	0.2055	0.0735
	38	0.0279	0.0403	0.0238	0.2083	0.2698	0.3386	0.0491	0.0701	0.0390
	39	0.0184	0.0268	0.0176	0.3560	0.4771	0.3473	0.0350	0.0507	0.0327
	40	0.0524	0.0588	0.0465	0.4139	0.3835	0.3579	0.0931	0.1020	0.0538
	41	0.0253	0.0272	0.0107	0.3124	0.3124	0.2625	0.0467	0.0501	0.0202
	42	0.0295	0.0224	0.0225	0.4476	0.3273	0.3192	0.0553	0.0420	0.0380
	43	0.0821	0.0947	0.0359	0.4909	0.4909	0.3043	0.1406	0.1587	0.0494
	44	0.0587	0.0740	0.0642	0.4919	0.4967	0.3170	0.1049	0.1288	0.0689
	45	0.0250	0.0257	0.0260	0.3823	0.3516	0.3322	0.0470	0.0479	0.0424
	46	0.0123	0.0132	0.0049	0.1465	0.1531	0.0899	0.0227	0.0242	0.0090
Mean	ALL	0.0251			0.2943			0.0385		
	L2A1	0.0367			0.3826			0.0653		
	L2N2	0.0385			0.3485			0.0674		

dimensional space of documents during the process of feature reduction.

And our runs performed well respect to for Summarization Only task, but not so well respect to Pre-Filtered Summarization task. The reason may be our filtering function is not good. On the other hand,for some topics, our NMFR method is not better than AP method and the average performance. The possible reason is that we excessive emphasis on the rate of convergence and operating efficiency, and ignored the locally optimal solution of our method. Therefore, the future work emphasis should be on how to improve the filtering ability and stability of our method.

Acknowledgment

This work was partially supported by the Beijing Excellent Talent Development Foundation, the Importation and Development of High Caliber Talents Project of Beijing Municipal Institutions (No. CIT&TCD201404052), the Guangxi Colleges and Universities Key Laboratory of cloud computing and complex systems, the Basic Research Programs of Science and Technology of BJUT, the BJUT Science and Technology Fund (ykj-2014-11508) .

References

[1] Zhao Y, Yao F, Sun H, Yang Z, et al. BJUT at TREC 2014 Temporal Summarization Track[C], Proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014).NIST Special Publication 500-308.

[2] http://www.trec-ts.org/.

[3] Lin C Y, Hovy E. From Single to Multi-document Summarization: A Prototype System and its Evaluation[J]. Proceedings of the Acl, 2002:457–464.

[4] http://www.psi.toronto.edu/affinitypropagation/vsh/.

[5] Ding C, Li T, Jordan M I. Nonnegative Matrix Factorization for Combinatorial Optimization: Spectral Clustering, Graph Matching, and Clique Finding[C]// 2008 Eighth IEEE International Conference on Data MiningIEEE Computer Society, 2008:183-192.

[6] Tang J, Gao H, Hu X, et al. Exploiting homophily effect for trust prediction[C]// Proceedings of the sixth ACM international conference on Web search and data miningACM, 2013:53-62.