Automatic Summarization for Chinese Text Using Affinity Propagation Clustering and Latent Semantic Analysis

Rui Yang, Zhan Bu, and Zhengyou Xia

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
zhengyou_xia@nuaa.edu.cn

Abstract. As the rapid development of the internet, we can collect more and more information. it also means we need the abitily to search the information which really useful to us from the amount of information quickly. Automatic summarization is useful to us for handling the huge amount of text information in the Web. This paper proposes a Chinese summarization method based on Affinity Propagation(AP)clustering and latent semantic analysis(LSA). AP is a new clustering algorithm raised by B. J. Frey on science in 2007 that takes as input measures of similarity between pairs of data points and simultaneously considers all data points as potential exemplars. LSA is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of sentences. Experiment results show that our method could get more comprehensive and high-quality summarization.

Keywords: Summarization, AP, LSA, clustering.

1 Introduction

With the rapid development of Internet, it has become increasingly important to find ways for getting useful information. The automatic summarization [1] can generate a short summary expressing the main meaning of a text, which helps us to find the information we really need quickly.

Summarization can be divided into different categories along several different dimensions. Based on whether or not there is an input query, the generated summary can be query-oriented or generic; based on the number of input documents, summarization can use a single document or multiple documents; in terms of how sentences in the summary are formed, summarization can be conducted using either extraction or abstraction — the former only selects sentences from the original documents, whereas the latter involves natural language generation. Overall, automatic summarization systems aim to generate a good summary, which is expected to be concise, informative, and relevant to the original input.

In this paper, we consider extractive summarization. We propose a new clustering algorithm called Affinity Propagation [2,3,4]. In the sentence extraction strategy,

F.L. Wang et al. (Eds.): WISM 2012, LNCS 7529, pp. 543–550, 2012.

clustering is frequently used to eliminate the redundant information resulted from the multiplicity. Clustering for text summarization mainly take the sentences into some clusters according to the distance or similary between each two senteces, and select one or more sentences from each cluster until fitting the length of the summary.

The popular k-centers clustering technique [5] begins with an initial set of randomly selected exemplars and iteratively refines this set so as to decrease the sum of squared errors. k-centers clustering is quite sensitive to the initial selection of exemplars, so it is usually rerun many times with different initializations in an attempt to find a good solution. However, this works well only when the number of clustersis small and chances are good that at leastone random initialization is close to a good solution. AP simultaneously considers all data points as potential exemplars. By viewing each data point as a node in a network, AP transmits real-valued messages along edges of the network until a good set of emplars and corresponding clusters emerges. As described later, messages are updated on the basis of simple formulas that search for minima of an appropriately chosen energy function. At any point in time, the magnitude of each message reflects the current affinity that one data point has for choosing another data point as its exemplar, so we call this method "affinity propagation".

Affinity propagation takes as input a collection of real-valued similarities between data points, where the similarity between two sentences calculated by Latent semantic analysis (LSA) here. LSA is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of sentences. LSA assumes that words that are close in meaning will occur close together in text. This method could mitigate the problem of identifying synonymy and the problem with polysemy [6].

This paper is organized as follows: We first give a brief survey on previous work in Section 2. Then, we detail our proposed automatic summarization method in Section 3. Following that, we present the experiments in Section 4. Finally, conclusions are given in Section 5.

2 Related Work

Automatic summarization has received a lot of attentions in recent years, and various approaches exist for extractive summarization, including the use of word frequency [7], cue words or phrases [8], machine learning [9], lexical chains [10] and sentence compression through syntactical or statistical restrictions [11].

Wang et al. [12] proposed a Chinese automatic summarization method based on thematic sentence discovery. They utilized terminology rather than traditional word as the minimal semantic unit, computed terminology weight with its length and frequency to extract keywords, and discovered thematic sentences using an improved k-means clustering method.

Rada Mihalcea and Paul Tarau [13] introduced TextRank – a graph-based ranking model for text processing. Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of “voting” or “recommendation”. When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

Jen-Yuan Yeh, Hao-Ren Ke [14] proposed a approach to address text summarization: LSA based and T.R.M.approach(LSA+T.R.M.).). They used latent semantic analysis(LSA)to derive the semantic matrix of a document or a corpus and uses semantic sentence representation to construct a semantic text relationship map.

Shasha Xie [15] investigated several approaches to extractive summarization on meeting corpus. Their thesis first proposed two unsupervised frameworks: Maximum Marginal Relevance (MMR) and the concept-based global optimization approach. Second, They treat extractive summarization as a binary classification problem, and adopt supervised learning methods and propose using various sampling techniques and a regression model for the extractive summarization task. Third, They focus on speech specific information for improving the meeting summarization performance.

Lucas Antiqueira, Osvaldo N. Oliveira Jr. [16] employed concepts and metrics of complex networks to select sentences for an extractive summary. The graph or network representing one piece of text consists of nodes corresponding to sentences, while edges connect sentences that share common meaningful nouns. This method uses a simple network of sentences that requires only surface text pre-processing, thus allowing us to assess extracts obtained with no sophisticated linguistic knowledge.

3 Process of LSA+AP

The process of LSA+AP consists of five phases: (1) preprocessing, (2) calculating similarities between sentences by LSA, (3) clutering the senteces in the text by AP algorithm, (4) sentences selection for summary.

3.1 Preprocessing

Preprocessing delimit each sentence by punctuation. Furthemore, it segments each sentence into words based on dictionary. In addition, we remove the meaningless words like "de", "a", etc.

3.2 Calculating Similarities between Sentences by LSA

LSA was patented in 1988 (US Patent 4,839,853) by Scott Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer, Karen Lochbaum and Lynn Streeter [17]. In the context of its application to information retrieval, it is sometimes called Latent Semantic Indexing (LSI).

We select the LSA to get the similarities between sentences because that LSA could analyzing relationships between a set of sentences by producting a set of concepts related to the sentences. LSA assumes that words that are close in meaning will occur close together in text. So it can mitigate the problem of identifying synonymy and the problem with polysemy.

The specific steps of LSA to calculate similarities between sentences are as follows:

Occurrence Matrix. The following elucidates how to construct the word-by-sentence matrix for the single-document.

Let D be a document, W(|W| = M)be the set of keywords in D, and S(|S| = N)be the set of sentences in D. A is the word-by-sentence matrix whose every row Si indicates a sentence and every column Wi indicates a keyword.

In A, $a_{i,j}$ is defined as Eq.(1), where L_{ij} is the local weight of W_i in S_j, and G_i is the global weight of W_i in D.L_{ij} is defined as $L_{ij} = \log(1 + \frac{aij}{nj})$, and G_i is defined as $G_i = 1 - E_i$, where c_{ij} is the frequency of W_i occurring in S_j, n_j is the number of words in S_j, and E_i is the normalized entropy of W_i , which is defined as (Bellegarda, Butzberger, Chow, Coccaro, & Naik, 1996).

$$aij = Gi \times Lij \tag{1}$$

Rank Lowering. We then perform singular value decomposition (SVD) to A. The SVD of A is defined as AUZV ^{T}, where U is an M×N matrix of left singular vectors, Z is an N×N diagonal matrix of singular values, and V is an N×N matrix of right singular vectors.

Finally, the process of dimension reduction is applied to Z by deleting a few entries in it, and the result of dimension reduction is a matrix $Z^{'}$. A new matrix, A', is reconstructed by multiplying three component matrixes. A' is defined as Eq. (2), where Z' is the semantic space that derives latent semantic structures from A, U' is a matrix of left singular vectors whose ith vector u_i' represents W_i in Z', and V' is a matrix of right singular vectors whose jth vector v_j' represents Sj in Z'.

$$A' = U'Z'V'T \approx A \tag{2}$$

Each column of A' denotes the semantic sentence representation, and each row denotes the semantic word representation.

Calculating Similarities between Sentences. In our method, a sentence S_i is represented by the corresponding sementic sentence representation, instead of the original keyword-based frequency vector. The similarity between a pair of sentences S_i and S_j is evaluated to determine if they are semantic ally related. The similarity is defined as Eq.(3).

$$\text{sim(Si,Sj)} = \frac{\overrightarrow{s_1} \cdot \overrightarrow{s_2}}{|\overrightarrow{s_1}| \cdot |\overrightarrow{s_2}|} \tag{3}$$

3.3 Clutering the Senteces in the Text by AP

Up to now, we have got the similarity between each sentence, which is the input of the AP clustering algorithm.

Rather than requiring that the number of clusters be prespecified, affinity propagation takes as input a real number s(k,k) for each sentence k so that the sentences with larger values of s(k,k) are more likely to be chosen as exemplars. These values are referred to as "preferences". The number of identified exemplars(number of clusters) is influenced by the values of the input preferences, but also emerges from the message-passing procedure. The preferences should be set to a common value— this value can be varied to produce different numbers of clusters. The shared value could be the median of the input similarities (resulting in a moderate number of clusters) or their minimum (resulting in a small number of clusters).

Because of that we need to generate a summary, so we want the number of clusters should be small, we select the minimum of the input similarities as the " preferences".

In addition, we knows that the first sentence in a paragragh always be important than other sentences in a chinese text. So we add a weight to a sentence if it is the first sentence of a paragragh as Eq.(4).

$$\text{P(i)} = \begin{cases} Pvalue + \gamma \times Pvalue, if \text{ Si is the first sentence} \\ Pvalue, if \text{ not} \end{cases} \tag{4}$$

Where the Pvalue is the minimum of the input similarities, and the γ is 0.9.

There are two kinds of message exchanged between sentences, and each takes into account a different kind of competition. Messages can be combined at any stage to decide which sentence are exemplars and, for every other sentence, which exemplar it belongs to. The "responsibility" r(i,k), sent from sentence i to candidate exemplar sentence k, reflects the accumulated evidence for how well-suited sentence k is to serve as the exemplar for sentence i, taking into account other potential exemplars for sentence i. The "availability" a(i,k), sent from candidate exemplar sentence k to sentence i, reflects the accumulated evidence for how appropriate it would be for sentence i to choose sentence k as its exemplar, taking into account the support from other points that sentence k should be an exemplar. r(i,k) and a(i,k) can be viewed as log-probability ratios. To begin with, the availabilities are initialized to zero: a(i,k) = 0. Then, the responsibilities are computed using the rule:

$$\text{r(i,k)} = \text{s(i,k)} - \max_{k' s.t. k' \neq k} \{a(i,k') + s(i,k')\} \tag{5}$$

Whereas the above responsibility update lets all candidate exemplars compete for ownership of a sentence, the following availability update gathers evidence from sentences as to whether each candidate exemplar would make a good exemplar:

$$\mathrm{a(i,k)} = \begin{cases} \min\{0, r(k,k) + \sum_{i's.t.i'\neq i,i'\neq k} \max\{0.r(i',k)\}\}, \mathit{if}\ \mathrm{i} \neq \mathrm{k} \\ \sum_{i's.t.i'\neq k} \max\{0, r(i',k)\}, \mathit{if}\ \mathrm{i} = k \end{cases} \tag{6}$$

For sentence i, the value of k that maximizes a(i,k) + r(i,k) either identifies sentence i as an exemplar if k = i, or identifies the sentence that is the exemplar for sentence i. The message-passing procedure may be terminated after a fixed number of iterations, after changes in the messages fall below a threshold, or after the local decisions stay constant for some number of iterations. We select in our process that if the exemplar have not changed for 20 times that the procedure could be terminated.

When updating the messages, it is important that they be damped to avoid numerical oscillations that arise in some circumstances. Each message is set to λ times its value from the previous iteration plus 1–λ times its prescribed updated value, where the damping factor λ is between 0 and 1. In all of our experiments, we used a default damping factor of λ = 0.5, and each iteration of affinity propagation consisted of (i) updating all responsibilities given the availabilities, (ii) updating all availabilities given the responsibilities, and (iii) combining availabilities and responsibilities to monitor the exemplar decisions and terminate the algorithm when these decisions did not change for 20 iterations.

Finally, we put the each sentence i which is not the exemplar into the corresponding exemplar which max the value of a(i,k) + r(i,k).

3.4 Sentences Selection for Summary

After clutering, we could select sentences from each cluster orderly until the length of summary is suitable. Morris A.H etal [18] tested many short text from GMAT reading comprehension and determined that the best length of the summary of the reported text was the 20% - 30% of the original length of text, and found that at the length the information of the summary is similar to the original text.

4 Experiments and Results

In this section, we report our experimental results.

4.1 Data Corpus

We select 100 news and articles from the web as our test text for generate the corresponding summary.

4.2 Evaluation Methods

There are two sorts of methods to evaluate the performance of text summarization: extrinsic evaluation and intrinsic evaluation (Mani & Bloedorn, 1999; Mani & Maybury, 1999). Extrinsic evaluation judges the quality of a summary based on how it affects other task(s), and intrinsic evaluation judges the quality of a summary based on the coverage between it and the manual summary. We chose intrinsic evaluation and used recall (R), precision (P) and F -measure (F) to judge the coverage between the manual and the machine-generated summaries. Assume that T is the manual summary and S is the machine-generated summary, the measurements are defined as Eq. (7) (Baeza-Yates & Ribeiro-Neto, 1999).

$$\mathrm{P} = \frac{|S \cap T|}{|S|}, \quad \mathrm{R} = \frac{|S \cap T|}{|T|}, \quad \mathrm{F} = \frac{2PR}{R+P} \tag{7}$$

4.3 Results

We select the TextRank and k-means clustering method for comparing to our method. Table 1 shows the average performance of each method.

Table 1. The average performance of each method

	R	P	F_measure
TextRank	0.377	0.392	0.376
k-means clustering	0.330	0.326	0.330
AP+LSA	0.665	0.489	0.548

From the table 1, we can see that our method could get better accuracy than the other two method.

5 Conclusions

We used affinity propagation to cluster chinese sentences for summary,and from the results we can see that this method could get a summary which has higher accuracy. LSA not only calculate the similary between two sentences effective and mitigates the problem of identifying synonymy and the problem with polysemy. Affinity propagation cluster need not define the number of the cluters in advance and it only need simple update rules and could get high precision.

There are also any problems in our proposed method, such as the definition of the preferences and the the damping factor λ .We will continue to research how to define these two parameters more effectively.

References

1. Sicui, W., Weijiang, L., Feng, W., Hui, D.: A Survey on Automatic Summarization. In: International Forum on Information Technology and Applications, IFITA (2010)
2. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972 (2007)
3. Mézard, M.: Where Are the Exemplars? Science 315, 972 (2007)

4. Eiler, J.M.: On the Origins of Granites. Science 315, 972 (2007)
5. Kummamuru, K., Lotlikar, R.: A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In: Proceedings of the 13th International Conference on World Wide Web. ACM, New York (2004)
6. Ai, D., Yuchao, Z., Dezheng, Z.: Automatic text summarization based on latent semantic indexing. In: Artificial Life and Robotics (2010)
7. Edmundson, H.P.: New methods in automatic abstracting. Journal of the Association for Computing Machinery 16(2), 264–285 (1969)
8. Paice, C.D.: The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In: Proceedings of the Third Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 172–191 (1981)
9. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 68–73 (1995)
10. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Advances in Automatic Text Summarization, pp. 111–121. MIT Press (1999)
11. Zajic, D., Dorr, B.J., Lin, J., Schwartz, R.: Multi-candidate reduction: sentence compression as a tool for document summarization tasks. Information Processing & Management 43(6), 1549–1570 (2007)
12. Meng, W., Chun-gui, L., Pei-he, T., Xiao-rong, W.: Chinese Automatic Summarization Based on Thematic Sentence Discovery. Computer Engineering 33(8), 180–181 (2007)
13. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain (2004)
14. Yeh, J.-Y., Ke, H.-R.: Text summarization using a trainable summarizer and latent semantic analysis. Information Processing & Management (2005)
15. Xie, S., Liu, Y., Hansen, J.H.L., Harabagiu, S.: Automatic Extractive Summarization on Meeting Corpus (2010)
16. Antiqueira, L., Oliveira Jr., O.N., da Fontoura Costa, L.: A complex network approach to text summarization. Information Sciences (2009)
17. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes (1998)
18. Morris, A.H., et al.: The effects and limitations of automated text condensing on reading comprehension performance. Information Systems Research (1992)