International Journal of Hybrid Information Technology
Vol.9, No.4 (2016), pp. 69-82
http://dx.doi.org/10.14257/ijhit.2016.9.4.07

An Extractive Approach for Uyghur Text Summarization

Turdi Tohti[1] Hankiz Yilahun[1] Askar Hamdulla[2]

[1]*School of Information Science and Engineering, Xinjiang University, Urumqi, china*
{turdy, hansumuruh}@xju.edu.cn
[2]*School of Software, Xinjiang University, Urumqi，China 830046*
Corresponding Author: askarhamdulla@gmail.com

Abstract

This paper studies Uyghur single text summarization and proposes some of new or improved approaches in the aspects of keyword extraction and evaluation, sentence selection and redundancy removal, also in readability improvement and so on. Proposes an improved frequent pattern-growth approach to extract the semantic strings which perfect both on its semantics and structural integrity, to evaluate this strings uses multi-feature fusion approach and select most important ones as keywords to describe the text theme effectively. In the aspect of sentence similarity and redundancy removal, proposes the idea of theme including degree, so as to effectively remove the redundant sentences and improves the summary quality significantly. Also introduces sentence alignment between the texts that after being stemming and original text, so as to solve the problems that summary naturalness, coherence and comprehensibility decline and other issues caused by stemming process.

Keywords*: extractive summarization, semantic string, theme including degree, redundant sentence removing, sentence alignment*

1. Introduction

According to the way of generation, automatic summarization can be divided into two categories, namely extractive summarization and abstractive summarization [1]. There is high summary quality of abstractive summarization and has the advantages of concise, comprehensive, accurate, and readable [2], but this approach requires complex natural language understanding and generation technology, needs a huge expert knowledge base and the perfect linguistic rules, so limited to the field and is still in its infancy [3].Extractive approach is organize the summary by key sentences extracted from the text, there is no needs for natural language understanding techniques or linguistic knowledge base, also has many advantages such as unrestricted application field, fast speed and adjustable summary length so on. So, most of the automatic summarization system is adopts this approach commonly [4]. This paper also studies an extractive approach for Uyghur text summarization.

Extractive approach is to extract a number of key words for the text firstly, and then to choose the key sentences according to the keywords appearance in sentence. Therefore, what kind of keywords to extract and the evaluation method to use will directly affects the correctness of key sentence selection [5]. The commonly used method is divide the text into words and evaluate the importance of each word, and then chooses the top-N words as key words according to word weight. However, because of the abstract meaning or ambiguity, a word cannot express the key information related to the theme in many cases. Therefore, more and more researchers are exploring the methods to extract the language units more meaningful than words ,such as phrases, compound words so on [6][7].

ISSN: 1738-9968 IJHIT

In this paper, we use an improved frequent pattern-growth algorithm and Uyghur word association rules to extract semantic strings (meaningful strings) from the text, evaluate them use a multi feature fusion method and select the most important semantic strings as the final keywords. In the aspect of sentence selection and redundancy removal, we proposed the idea of theme including degree and effectively removed the redundant sentences and improved the summary quality significantly. In addition, also introduces sentence alignment between the texts that after being stemming and original text, so as to solve the problems that summary naturalness, coherence and comprehensibility decline and other issues caused by stemming process.

2. Keyword Extraction

Uyghur language belongs to Turkish language group of Altaic language family, and also belongs to the agglutinating language on structure grammar. Looking on the surface, Uyghur text is a word sequences that separated by inter-word spaces and on this feature is similar to English. For this reason, word segmentation always have been ignored in Uyghur natural language processing and uses the inter-word space as a natural separator to simply obtain the words in the text, is the only word segmentation method so far. However, the minimal language unit that can express complete semantics often is not a Uyghur word, but is a semantic string that breaks through the conceptual boundaries of word [8]. Semantic string is a stable combination of contextual characters (words) in the text, and also is an indivisible language unit on its semantics and structural integrity, such as collocations, idioms, pattern strings with lexical meaning and grammatical meaning [9], word group or phrases [10], compound word or domain terms [11], and named entities so on.

In text, the sentence can express a complete, coherent and understandable semantics, and semantic string is contains the key information in the sentence. So, if the semantic string is taken as keywords, it is more effectively to describe the text theme, and also improve the efficiency of key sentence selection.

2.1. Semantic String Extraction

The proposed approach is makes downward extension for each word according to the writing direction and discovering all the semantic strings in the text, so this will requires the frequency of each word or word string, length, locations, the part of speech, context and other statistical information. Therefore, we have designed a kind of multilevel dynamic index structure to store the above information [12], extracted semantic strings after frequent patterns discovery and evaluation. Frequent pattern discovery is the improvement of pattern-growth algorithm [13], mainly are the following steps:

(1) Build inverted index for the text after being pretreatment. Such as, for the text only have six word "ABCF# EFCEABCFD# EFCADFECDABCFACD#" (# is punctuation), build word index is shown in figure 1.

In the first Level index, “termID” is the unique ID of index term (a word or word string); “Freq” is frequency of index term in the corpus, “is_stop” is the mark of stop words, “is_adj” is the mark of adjectives, “Unit_count” is the word length of the index terms (number of words to be contained), “Pos_pointer”, “Lv_pointer” and “Rv_pointer” are the offset address corresponding to the second level index. The second level index also is an index term list, and its entrance is obtained by the first level index. Among them, “Position” is an inverted list for index term, “Lv_pointer” is includes all left adjacency of index term and its frequency, and “Rv _pointer” is includes all right adjacency and its frequency.

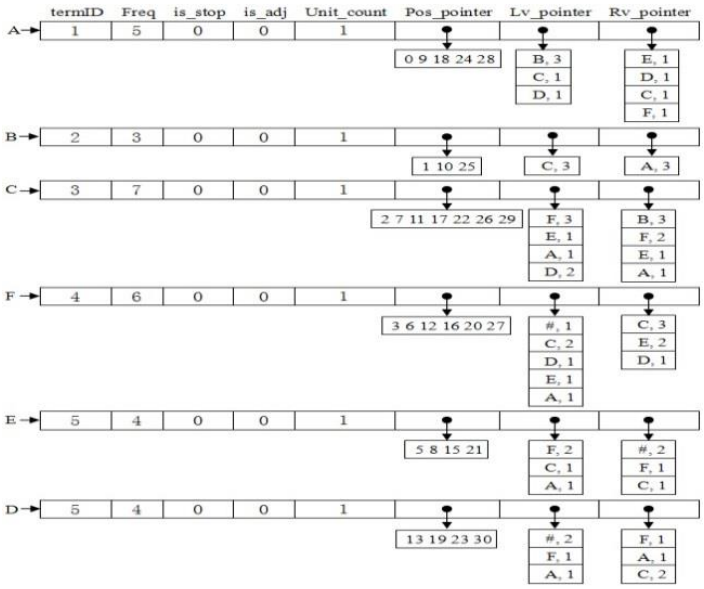
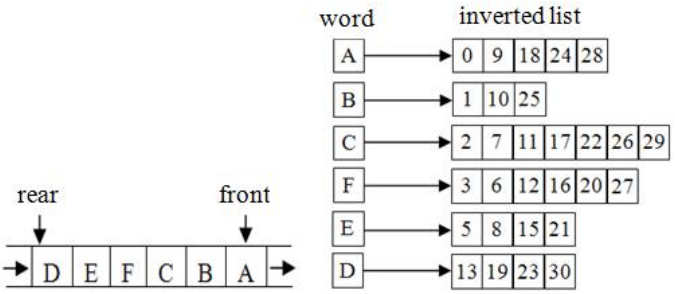


Figure 1. Index Sample

Through this kind of index structure, can describe the attributes of each word or string as much as possible, its dynamics, efficiency and extensibility also can meet the demand of massive text processing.

(2) Word string extension and frequent pattern discovery. At the beginning, let all the words into a queue, and the double-word or three-word strings will be extend from each word according to its index information, then let the words be visited is out of queue and the word strings new generated are enqueue, so that continue to extension that from n word to n+1word or n+2 word iteratively until the queue is empty. Word index for string extension candidates and queue initial state as shown in figure 2.



(a) Initial state of queue (b) Initial state of inverted list

Figure 2. String Extension Initial State

In this example, whether word "A" can be an frequent pattern with its following word "B", we go to judge their association strength using language rules, and the evaluation criteria of *Confidence* and *R-confidence* [14]. *Confidence* refers to the conditional probability of the below word W_i of word association $W_{i-1} \rightarrow W_i$ in the case that it's above word W_{i-1} is appeared.

Accordingly, *R-Confidence* refers to the conditional probability of the above word W_{i-1} of word association $W_{i-1} \rightarrow W_i$ in the case that it's below word W_i is appeared, calculated as follows:

$$Confidence\ (W_{i-1}\rightarrow W_i) = P(W_i|W_{i-1}) = \frac{(W_{i-1} \bigcup W_i).count}{W_{i-1}.count} \quad (1)$$

$$R\text{-}Confidence\ (W_{i-1}\leftarrow W_i) = P(W_{i-1}|W_i) = \frac{(W_{i-1} \bigcup W_i).count}{W_i.count} \quad (2)$$

It can be seen that, the *Confidence* evaluate the above word proportion in this word association, and the *R-Confidence* is to measure the below word contributions for this association strength. Thus, when the *Confidence* (W_{i-1}, W_i)>*minconf* or *R-Confidence* (W_{i-1}, W_i)>*minconf* (minconf for minimum confidence threshold), it can be determined that the word string "W_{i-1}, W_i" is a trusted frequent pattern.

In our work, we also found the following language features are very useful for association patterns recognition:

Feature 1:There are small quantities of Uyghur words but frequently appearance in the text that like auxiliary words, conjunctions, adverbs, quantifiers, pronouns, and interjections etc,this kind of words has never associate with other words and to constitute a semantic string, be referred to as *independent word* (IW)in this paper.

Feature 2: word combinations mainly occur between in the nouns (N), adjectives (ADJ), and verbs (V) in Uyghur language. Among them, Adjectives always as an above word of a word association and it will not appear in the below word position when the adjectives associated with nouns or verbs. So, words like "N+ADJ" or "V+ADJ" relations never to be a semantic string in text.

According to the Feature 1 and Feature 2, the word association rule (WAR) uses for association pattern recognition defined as follow.

Definition (WAR): For the adjacent Uyghur words "X Y", Such as the establishment of conditions: "X" ∈{IW} or "Y" ∈{IW} or "Y" ∈{ADJ}, then it is can be determined that "X" and "Y" cannot be combined with an association pattern.

According to above rules and evaluation criteria, for the extension of "X"→"XY", should meet the following conditions:

① "X" is not a stop word, namely is_stop (X) = 0;

②"X" is a frequent pattern, namely the Freq (X) > = 2;

③"Y"is not a stop word or adjective, namely is_adj (Y) = 0 and is_stop (Y) = 0;

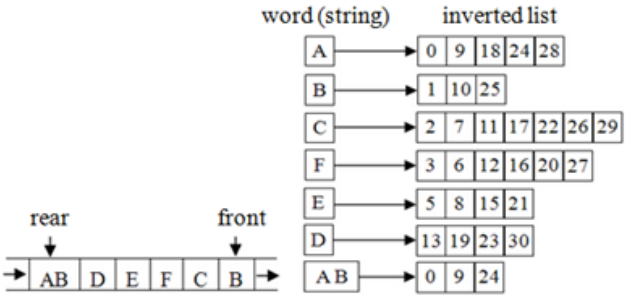
④"Y" is a frequent pattern, namely the Freq (Y) > = 2;

⑤"X Y" is a trusted frequent pattern, namely *Confidence*(X, Y)> *minconf* and *R - Confidence* (X, Y)>*minconf*;

In the example above, because the word "A" is qualified conditions ① and ②, so loads its left adjacency list from the second level index, followed by judge the association degree and new word string generation possibility of "A" and its each left adjacency in turn according to the conditions ③, ④,and ⑤. by In this case, "B" is the first left adjacency of "A" and B qualified the condition ③ and ④,and the new string "A B" also qualified condition ⑤, so the generated string "A B" enter the queue and its information appends to the index file at the same time, then turn to the next one "C". In this way, double-word string extension from "A" will be end when all left adjacency of "A" were visited (the associations "A C" and "A D" are impossible). At this point, changes of queue and index as shown in Figure 3.

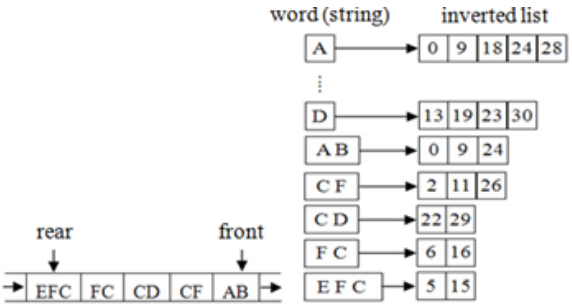
After the extension from "A" is end, it is turn to the "B", but "B" has associated with "A" at the previous, so skip "B" and the next word should be dequeue is "C". In this way, carry out double-word or three-word extension for each word in turn, and the word strings

new generated are enter the queue and waiting for being extension further in next round. At this point, the queue and index changes as shown in figure 4.



(a) Queue changes (b) index increase

Figure 3. Sample1 for String Extension



(a) Queue changes (b) index increase

Figure 4. Sample 2 for String Extension

After first round extension is end, the algorithm will enter the next round of extension process and generates longer word strings continually until the queue is empty. At this time, frequent pattern discovery process is all over.

(3) Structural integrity evaluation and semantic string extraction. Semantic string candidates obtained by frequent pattern discovery can only meet the statistical conditions, and it is also needs the further purification. The effectively method is evaluate structural integrity of each candidate semantic string uses the contextual adjacency features. Related research results show that the adjacent entropy is more effective than other three kind of adjacent features (adjacency categories, dual adjacency categories, and dual adjacent entropy) [15]. Therefore, candidates in the frequent pattern result were weighted using the formula (3).

$$AE_{weight}(S) = \min(LAE(S), RAE(S)) \tag{3}$$

where AE_{weight} (S) is stands for the adjacent entropy weight of string S, LAE (S) is the left adjacent entropy of string S, RAE (S) is the right adjacent entropy, left (right) adjacent entropy calculated as shown in formula (4).

$$\mathrm{LAE(s)} = -\sum_{i=1}^{m} \frac{n_i}{N} \log\left(\frac{n_i}{N}\right) \tag{4}$$

where m is the left adjacency variety of S, n_i is the ith left adjacency frequency of S, all left adjacency frequency adding up to N, all the above information needed to calculate the adjacency feature is recorded and stored in the index as early as when the strings(frequent patterns) were found. Finally, out put the strings which achieved the threshold and that is the semantic strings what we ultimately want to get. The process is shown in figure 5.

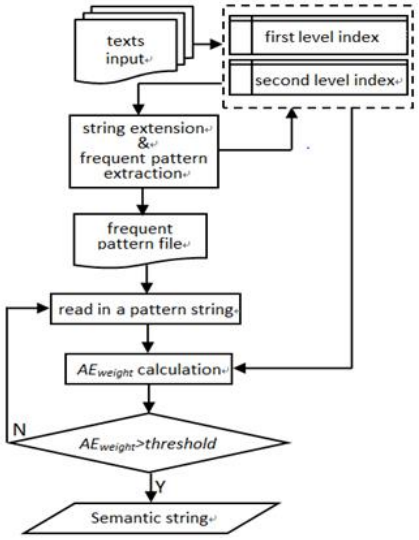


Figure 5. Semantic String Extraction Processes

2.2. Semantic String Evaluation and Keyword Extraction

1) Feature of adjacent entropy.Adjacent features stand for the structural integrity of semantic string in the pragmatic environment and a word string with structural integrity is always express key information related to the theme of the text. So, the semantic strings with greater value of adjacent entropy should be given greater weights accordingly.

2) Feature of TFIDF.A semantic string with especially low frequency or appears in most of the text, its representative is not strong and should not be selected as keywords. There is greater correlation between the theme and semantic strings which with higher frequency and appears in a small amount of text in the text set, so this kind of semantic string also should be given greater weights according to the definition of TFIDF evaluation.

3) Feature of string length. The length of the semantic string is proportional to the amount of information expressed, so the length of the string is longer and the amount of information expressed by the semantic string is also greater. For instance, the amount of information expressed by semantic string “highway charge system” is more specific than “highway “or” highway charge”, such as the semantic strings with longer length be selected as the key words, and this kind of keywords could describe the theme of the text bitterly. So, the weight for such semantic strings with longer length should greater than shorter length.

4) Multi-feature fusion.In the above several features, the size of adjacent entropy not only reflects the frequency of semantic string but also reflects its structural integrity, TFIDF reflects the correlation between semantic string and the theme of the text, and the feature of string length is a measure of information amount of expressed bay this semantic string. Therefore, according to the important degree of different features in semantic string evaluation, the following comprehensive evaluation formula is proposed, which is

$$W_i = (AE_{weight} + TFIDF_{weight}) \times \sqrt{Unit\_count} \quad (5)$$

where W_i is the weight of ith semantic string, AE_{weight} is adjacent entropy calculated by the formula (3). In $TFIDF_{weight}$ calculation, TF is the frequency of the semantic string in the semantic string set, IDF is its inverse document frequency.

In the end, we calculate the weight of each semantic string using the formula (5), and then choose the top-N semantic string (N<=20 and greater than the average weight) as key words for each text.

3. Key Sentence Selection and Redundancy Removal

After keywords extraction for each text, the next step is to extract the key sentences for the summary organization, this requires split the original text into sentences firstly, segmentation and refining the sentences secondary, and then get the candidate sentences that relatively complete both on their structure and content [16]. Then weighting and choose the top-N sentence no repeatedly from the sentences sorted by their weight. So, the main work of the next step is sentence weighting, candidate sentences selection and the redundancy removal.

3.1. Sentence Weighting

Generally, sentence weight is measured according to the number of keywords be contained in the sentence, so the sentence which contains more keywords will be given greater weight than others accordingly. However, there is poor efficiency because this measure only considering the number of keywords without its quality. In fact the sentence which contains a few important keywords is more useful for the text than those sentences contains multiple secondary keywords. In our work, we extract the semantic strings as keywords to describe the text, evaluate and weighting each keyword from the aspects of its structural integrity and information quantity be expressed, so we weighting a sentence use the weight sum of not repeat keywords in this sentence . Such as defining U_d is a set of keywords of text d, U_s is a set of not repeat keywords contained by sentence S in the text d, and $U_s \subseteq U_d$, then the sentence weighting formula is:

$$W(s,d) = \sum_{i=1}^{n(n\leq 20)} W(k_i, s) \tag{6}$$

where $W(s, d)$ is the weight of sentence s in text d, $W(k_i, s)$ is the weight of ith keyword k_i in sentence s, n is the number of keywords selected for the sentence s.

3.2. Sentence Redundancy Removal

There may be similar sentences in the candidate sentence set and the similarity between sentences may be one of the following three situations: the first is part of similar or partially contained, is refers to two sentences contains some common words; the second is completely contained, which means that a sentence is the sub sentence of another sentence; third is the two sentences are exactly same. Among them, the third is rare, probability of the second case is not very high, and the first case is common [17].

For the question of sentence redundancy removal, as a most commonly used approach of VSM based sentence similarity is to express a sentence as a vector, and calculate the inner product or included Angle cosine between the sentence vectors. After that recalculate or modify each sentence weight according to the maximum redundancy and average redundancy in turn, reorder them and select several

sentence which highly weighted, and to organize the summary according to the summary length be defined and other conditions[18].

Another method is to consider the sentence A and B as word set U_A and U_B respectively, and measure the similarity between sentence A and B according to the proportion of $U_A \cap U_B$ for U_A, and U_B. Such as the information redundancy between sentences is quantified as $|U_A \cap U_B|$, then the redundancy of A and B relative to the A is expressed as:

$$Overlap\_A = \frac{U_A \bigcap U_B}{U_A} \tag{7}$$

Therefore, the redundancy between sentence A and B can be defined as *Redundancy* (*A*, *B*) =*max* (*Overlap_A*, *Overlap_B*). By definition, such as *Redundancy* (*A*, *B*) =0 is means that the sentence A and B are totally dissimilar; if *Redundancy* (*A*, *B*)> 0, it means that they have common words, when *Redundancy* (*A*, *B*)>T_{sim} (T_{sim} for similarity threshold), it is can be determined that the sentences A and B are similar [19].

On the basis of the above methods and the universality of text summarization, we propose the idea of including degree between sentence and text theme, and follow the following steps to remove the redundant sentences from the summary sentence collection of text d.

Step1: For the text *d*, initialize keyword set U_d, and U_d is empty at the beginning;

Step2: Select the top one from the sentences sorted by their weight and add it to the set of summary sentences, and all of keywords (semantic strings) contained by this sentence are filled to the U_d at the same time;

Step3: The summary sentence selection would be come to the end if the sum of length of the sentences has been selected has reached the summary length be defined, otherwise select a sentence with acceptable length in proper order, and perform the step4.

Step4: U_s is the keyword set of the sentence *S* to be measured, and then the redundancy between sentence *S* and the theme of text is represented as:

$$Overlap\_S = \frac{U_S \bigcap U_d}{U_S} \tag{8}$$

It is can be seen that the sentence *S* is not redundant for the text theme if *Overlap_S* is less than a given threshold, it is indicating that sentence *S* has the contribution to express the theme of the text, so it should be selected as a summary sentence and all of keywords (semantic strings) contained by *S* are filled to the U_d at the same time, and turn to step 3; If *Overlap_S* is greater than a given threshold, it means that the sentence *S* is redundant for the theme ,so don't select it as summary sentences, and turn to step 3.

In the above process, the keywords appears in the selected sentences are continuously filled to the theme word set U_d, and U_d no longer accept keywords when the information accumulation expressed by these keywords in U_d is close to the theme of the text. From another perspective, if the keywords of sentence *S* are not accepted in U_d, it means that the information expressed by sentence *S* was contributed to U_d by keywords appears in the sentences has been selected, so the information expressed by sentence *S* is redundant for the theme of the text.

4. Summary Organization

Stemming is needs for the keyword and key sentence extraction in Uyghur text summarization, but the appearance of keywords and key sentences in the form of

stem sequence would be lose their language naturalness and coherence, and there is poor readability and understandability of the summary.

In view of this situation, we have prepared two kind of text corpus that the one is original corpus and another is a copy of original corpus and being stemmed. First of all, word segmentation and sentence segmentation conducted for the original corpus and the stemmed corpus respectively, and the sentence alignment is established between the two corpuses, and all of the processes about keyword extraction and sentence selection are conducted on the stemmed corpus. Finally, obtained the sentences corresponds to the original text according to the sentence alignment and the final summary is organized by the sentences order in the original text. The Uyghur text summary extraction process is shown in figure 6.

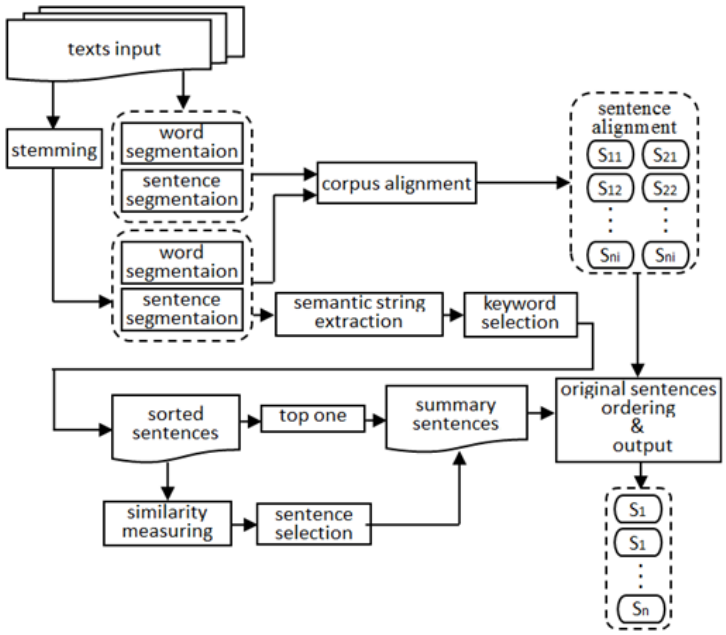


Figure 6. Uyghur Text Summary Extraction Process

5. Experiments and Analysis

5.1. Corpus for Experiment

The corpus for experiments is including the 6021news text from internet and a total of 846 texts from the Xinjiang Daily for July 2013, size of the corpus is 51.14 MB.

5.2. Time Efficiency

We select 821 texts from the corpus and divide it into several subsets according to the size. Statistics of time required for summary extraction are shown in table 1.

Most of the text size is less than 9KB and the required time is within 5ms, the extraction time increase with the increase of the text size. The change trend is shown in Figure 7.

Table 1. Text Size and the Time of Required

Text size (KB)	Number of text	Mean time (ms)
0.3~0.99	142	0.550
1.0~1.99	103	0.758
2.0~2.98	108	1.302
3.0~3.99	88	1.775
4.0~4.98	69	2.264
5.0~5.99	67	2.565
6.0~6.93	51	3.063
7.0~7.99	42	3.720
8.0~8.98	42	3.917
9.0~9.96	24	4.557
10.0~10.9	27	4.631
11.0~11.8	17	4.596
12.0~12.9	17	4.728
13.0~13.7	9	5.208
14.0~14.9	15	6.371

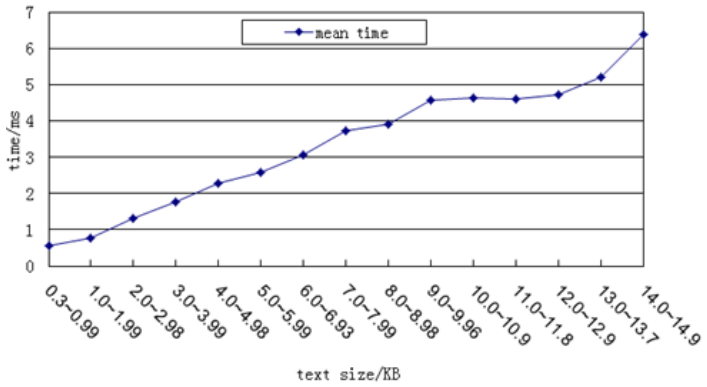


Figure 7. Changes in Text Size and Time Efficiency

5.3. Summary Quality

Quality evaluation is a difficult point in the study of automatic summarization, and still faces many challenges. First of all, it is very difficult to reflect the summary contains amount about original information and this will increases the difficulty and complexity of the automatically evaluation. In addition, summary contains specific information accordance with the actual needs of the user or applications and it should be evaluated from the needs of different users or applications, but it will also make the summary quality automatic evaluation become more complicated.

However, it will needs large quantity of manpower if using the artificial way for evaluation, so we only select a small amount of 20 texts from the test corpus and to manually inspect the summary quality. The results showed that the summary express the theme of text is basically correct and its naturalness and coherence are better also. Keyword and summary extraction results for short text and long text are shown in figure 8 and figure9.

Figure 8. Summary Extraction for Short Text

Figure 9. Summary Extraction for Long Text

Of course, we will further improve the summary quality according to the user feedback in the open testing and in our future work.

6. Conclusion

As an open field approach, extractive summarization more able to meet the demand for massive text summarization, its time efficiency and the summary quality also able to meet the practical needs. So, this paper studied an extractive approach for Uyghur text summarization and the main contributions of this paper are as follows: First, we extract the semantic string in the text as the keyword, because it expresses more complete and specific information than a word, so it is possible to describe the text theme better than in the traditional way of extract the word as keyword. In keywords selection, we proposed a multi feature fusion approach to evaluate and select the most important semantic strings as the keywords, so as to make the correct choice of the key sentences. In the aspect of sentence similarity and redundancy removal, we proposed the idea of the theme including degree, so as to remove the redundant sentences and improved the summary quality effectively. In addition, we also introduced sentence alignment between the texts that after being stemming and original text, so as to solve the problems that summary naturalness, coherence and comprehensibility decline and other issues caused by stemming. In

the end, we also realized an extractive system for Uyghur text summarization, this system performed relatively satisfactory on its time efficiency and summary quality in the evaluation.

Acknowledgements

This work has been supported by the National Natural Science Foundation of China (61262062, 61163033, and 61562083), Scientific Research Program of the Higher Education Institution of Xinjiang (XJEDU2012I11) and Innovation Program for Excellent Ph.D. Candidates of Xinjiang University (XJUBSCX-2013011).

References

[1] D.Vipul, M.Latesh, "A survey of extractive and abstractive text summarization techniques", International Conference on Emerging Trends in Engineering and Technology, **(2013)** December16-18; Nagpur, India.
[2] C.C.Li, S.Wang, "Study of automatic text summarization based on natural language understanding",2006 IEEE International Conference on Industrial Informatics,**(2006)** August16-18; Singapore, Singapore.
[3] H.T.Le,T.M.Le, "An approach to abstractive text summarization", 2013 International Conference on Soft Computing and Pattern Recognition,**(2013)** December15-18; Hanoi, Vietnam.
[4] S.C.Wang,W.J.Li,F. Wang,H. Deng, "A survey on automatic summarization",Proceedings - 2010 International Forum on Information Technology and Applications,**(2010)** July 16-18; Kunming, China.
[5] C.H.Dang,X.J.Luo, "Key sentence based text summarization using Keywords and WordNet" WSEAS Transactions on Computers, vol. 6, no. 5, **(2007)**.
[6] S.V.S.S. Lakshm, K.S.Deepthi, C.h.Suresh, "Text summarization basing on font and cue-phrase feature for a single document", 49th Annual Convention of Computer Society of India: Emerging ICT for Bridging the Future, **(2015)** December 12-14; Hyderabad, India.
[7] X.L.Liu, "Automatic summarization method based on compound word recognition", Journal of Computational Information Systems, vol. 11, no. 6, **(2015)**.
[8] T.Turdi, M.Winira, H.Askar, "Unsupervised Learning and Linguistic Rule Based Algorithm for Uyghur Word Segmentation", Journal of Multimedia, vol. 9, no. 5, **(2014)**.
[9] M.Candito,M.Constant, "Strategies for contiguous multiword expression analysis and dependency parsing", 52nd Annual Meeting of the Association for Computational Linguistics, **(2014)** June 22-27; Baltimore, MD, United states.
[10] N.H.Rais, M.T.Abdullah, R.A.Kadir, "Multiword phrases indexing for malay-english cross-language information retrieval", Information Technology Journal, vol. 10, no. 8, **(2011)**.
[11] M.Masaki, U.Masao, " Compound word segmentation using dictionary definitions- extracting and examining of word constituent information", ICIC Express Letters, Part B: Applications, vol. 3, no. 3, **(2012)**.
[12] Y.Ma, L.Wang, "Dynamic indexing for large-scale collections", Journal of Beijing Normal University (Natural Science), vol. 45, no. 2, **(2009)**.
[13] R.U.Kiran,P.K.Reddy, "An improved frequent pattern-growth approach to discover rare association rules", 1st International Conference on Knowledge Discovery and Information Retrieval,**(2009)** October 6-8; Funchal, Portugal.
[14] J.K.Jain, N.Tiwari,M. Ramaiya, "Mining positive and negative association rules from frequent and infrequent pattern using improved genetic algorithm", Proceedings - 5th International Conference on Computational Intelligence and Communication Networks,**(2013)** September 27-29; Mathura, India.
[15] H.P.Zhang,K.Gao,H.Y.Huang,Y.P.Zhao, "Big data search and mining",science press, Beijing **(2014)**.
[16] L.Z.Feng,R.F.Li,Y.Q. Zhou, "Extracting key sentiment sentences from internet news via multiple source features", Proceedings of 4th IEEE International Conference on Network Infrastructure and Digital Content, **(2014)** September 19-21; Beijing, China.
[17] L.B.Yang,X.Y.Cai,Y.Zhang,P.Shi, "Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization", Information Sciences, vol. 260, **(2014)**.
[18] X.Liang,D.J.Wang,M.Huang, "Improved sentence similarity algorithm based on VSM and its application in question answering system",Proceedings-2010 IEEE International Conference on Intelligent Computing and Intelligent Systems,**(2010)** October 29-31; Xiamen, China.
[19] H.H.Vu,J.Villaneau,F.Saïd,P.F.Marteau, "Sentence similarity by combining explicit semantic analysis and overlapping n-grams", 17th International Conference on Text, Speech, and Dialogue, **(2014)** September 8-12; Brno, Czech republic.

Authors

Turdi Tohti. He was born in 1975, received B.E. in 1999 from Nanjing University of China, received M.E. in 2009 from Beijing University of technology of China, received Ph.D. in 2014 from Xinjiang University of China. He is an associate professor and Graduate Supervisor in the School of Information Science and Engineering, Xinjiang University. He has published more than 30 papers on international journals, national journals, and conferences in recent years. His research interests include natural language processing, information retrieval, web mining and content security. He is a Senior of CCF, member of IEEE CS and CIPSC.

Hankiz Yilahun. She received B.E. and M.S. degree in Computer Science and Technology from Xinjiang University and Beijing University of Technology, China, in 2002 and 2009, respectively. Currently, she is a PhD candidate in Computer Applications in Xinjiang University, and working as a teacher at Mathematics and System Science, Xinjiang University, China. Her research interests include Natural Language Processing, Uyghur ontology and its applications.

Askar Hamdulla. He received B.E. in 1996, M.E. in 1999, and Ph.D. in 2003, all in Information Science and Engineering, from University of Electronic Science and Technology of China. In 2010, he was a visiting scholar at Center for Signal and Image Processing, Georgia Institute of Technology, GA, USA. Currently, he is a professor in the School of Software Engineering, Xinjiang University. He has published more than 150 technical papers on speech synthesis, natural language processing and image processing. He is a senior member of CCF and an affiliate member of IEEE.