Automatic text summarization based on syntactic links

Yerimbetova A.S.[1,2], Batura T.V.[3,4], Murzin F.A.[3,4], Sagnayeva S.K.[4],

[1]Institute of Information and Computing Technologies CS MES RK, Almaty, Kazakhstan
[2]Kazakh Academy of Transport and Communications named after M.Tynyshpaev, Almaty, Kazakhstan
[3]A.P. Ershov Institute of Informatics Systems SB RAS, Novosibirsk, Russia
[4]Novosibirsk State University, Novosibirsk, Russia
[5]L.N. Gumilyov Eurasian National University, Nus-sultan, Kazakhstan

Abstract. The task of information retrieval is to find documents relevant to the query in a certain collection of documents. The document is a text selected by the author as a single fragment. A query is usually a meaningful phrase or set of words describing the information needed. Instead of searching through the whole document, organizing a search by topic or resume of the document becomes enough. By the term "topic" we refer to a set of small reference texts. Therefore, one of the interesting tasks in information retrieval systems is the task of classifying texts by topic. The whole classification process is carried out in four stages: preprocessing the text, weighing the terms, weighing the sentences, extracting meaningful sentences.
In the process of selecting topics, fragments of the text are studied (for example, paragraphs) and compared with the chosen standard. Different fragments can be attributed to different topics. Selected fragments can be combined into a summary on this topic. This paper considers the issues of automatic summarization of text documents taking into account the syntactic relations between words and word forms in sentences that can be obtained at the output of the Link Gramma Parser (LGP) system for the Kazakh and Turkish languages. The authors operate on the results of studies on customizing the LGP parser for agglutinative languages.

Keywords: Information retrieval, word weight, directed graph, text topics, Closeness centrality, LGP, summarization.

1 Extraction and Summarization Methods

1.1 The task of automatic summarization

The basis for writing this article was article [1], which discusses the process of automatic summary. The task of automatic summarization is the creation of a short version of a text document or a collection of documents that reflects the most significant information of the original document or documents [2]. Traditionally, automatic summarization problems are divided into several independent directions of the classification of solved problems and the types of generated annotations.

ICID-2019 Conference

Most modern automatic summary systems operate on the basis of the extractive approach, i.e. selecting entire sentences of the source collection for automatic annotation without changing the sentences themselves.

One of the problems that arise with this approach is that permutation of words in a sentence can significantly change its meaning, which leads to incorrect operation of algorithms that operate on individual keywords, their frequencies, etc. In the above-mentioned work by Niraj Kumar [1], another method was proposed. It allowed to take the word order and showing its effectiveness into account.

1.2 Methods of thematic analysis of text information of automatic summarization

Of the well-known methods such as: Lexical Chains, Latent Semantic Analysis (LSA), and graph theory for constructing automatic annotations, the last approach is of interest to us.

Using graph theory for the automatic summary problem is close to the ideas of the well-known PageRank link ranking algorithm [3]. In the framework of this approach, the input text collection is presented in the form of a fully connected graph, the vertices of which are sentences, and the edges between them reflect the weight of similarity between these sentences. Sentences that have strong ties to a large number of other sentences are likely to be central and should be highly relevant for inclusion in the resulting annotation.

The described algorithm does not require deep linguistic preprocessing, with the exception of highlighting sentences and individual words, therefore it can be used for various languages. At the same time, adding semantic and syntactic information when constructing a sentence graph can improve the results of the algorithm [4].

Differentiation of the schemes for calculating the similarity of sentences belonging to the same and different documents of the input collection allows you to separate local topics of documents from global topics shared by all documents in the collection.

1.3 Calculate word weights

In the general case, we believe that there is some fragment of the analyzed text, or a pre-prepared “test” fragment, i.e. sequence of sentences $S = < S_1, S_2, ..., S_n >$. Next, a sequence S is mapped to some graph. The matching method can be easily demonstrated using the example given in [1]. The following shows the sequence of sentences and the corresponding graph:

Sentences:

S1: a b c d e

S2: c a f g

S3: h i f a b

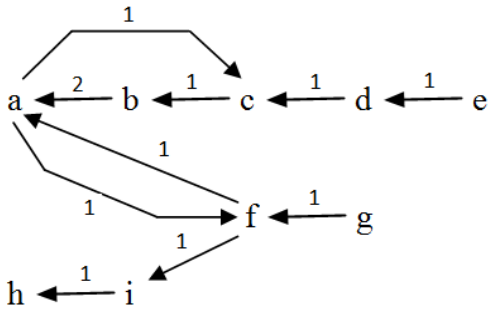

Fig.1 Text sentence graph

Here $a,b,c,d\ldots$ – are the words of the corresponding sentences. The set of all words found in a given set of sentences forms the set of vertices of the graph. The edge of the graph reflects the fact that one word follows another word. In this case, the direction of orientation of the edge is set from the next word to the previous one. The weight of an edge is the number of occurrences of a given pair of words following each other in this order in the entire text fragment $S =< S_1, S_2, ..., S_n >$. In [1], the authors suggest that some text preprocessing is carried out by removing various kinds of service words.

We offer some generalization of the algorithm taking into account the syntactic structure of sentences received at the output of the LGP system for the Kazakh and Turkish languages [5]. We also rejected the requirement to normalize words and word forms. In the end, when comparing sentences, we can take into account a larger or smaller set of specific relationships, and so some links are simply ignored.

For each connection, it is important to consider how many times it met, taking into account the directions. It is necessary to take into account the number of links entering the top and the number of links coming out of the top. For this, the concept of vertex rank (weight of each word) is introduced in [4].

To calculate the weight of each vertex (word), we use the PageRank algorithm. For each vertex, we introduce the notation $IN(V_i)$ – then the many vertices that reference it (predecessors), and $OUT(V_i)$ – the many vertices referenced by the vertex itself V_i (*descendants* V_i). Then the rank of each vertex of the graph (the weight of each word) can be determined by the formula:

$$S(V_i) = \frac{1-\lambda}{N} + \lambda \sum_{V_j \in IN(V_i)} \frac{S(V_j)}{|OUT(V_j)|}, \qquad (1)$$

$S(V_i)$ – vertex rank (word weight) V_i;

$S(V_j)$ – vertex rank (word weight) V_j, from which the connection is directed to the top V_i;

$|OUT(V_j)|$ – number of descendants of the peak V_j;

N – number of graph vertices;

λ – damping factor, for example, in [4] a fixed value equal to «0,85».

The above formula (1) indicates that the vertex referenced from other vertices with a high rank itself receives a high rank. Thus, it is possible to single out in the whole text the most important key words.

Let us try and analyze some issues related to rank calculation. Obviously, if an acyclic graph corresponds to a sentence or a set of sentences, that is, it is a tree, the ranks can be calculated recursively during the tree traversal. First, we compute the ranks of the hanging vertices in which there are no incoming edges, then we need to take only the first term from the definition of rank. Then we calculate the ranks of the vertices, in which there are edges from the vertices for which the ranks have already been calculated, etc. That is, we move along a certain hierarchy of vertices and simply calculate. In the case when the graph has cycles, then we are forced to solve some systems of equations.

1.4 Identification of text topics and calculation of their importance

We understand a topic as a set of sentences related to one concept, phenomenon, sequence of events, etc. To determine the topics contained in a document, various agglomerative clustering procedures are usually applied. [6]. These include: Single linkage (simply connected clustering; also known as the “nearest neighbor method”), Complete linkage (fully connected clustering; also known as the "far neighbor method"), Pair-group method using arithmetic averages (also known as the “pairwise mean method”), Pair-group method using the centroid average, Ward’s method.

These classification methods are often called hierarchical, because objects in them are combined into more and more large clusters [7]. Initially, each object in them is considered a separate cluster. For singleton clusters, the distance function is determined in some way:

$$d_{ij} = d(\{x_i\},\{x_j\}) = \rho(x_i, x_j). \qquad (2)$$

Then, the cluster merging process starts. We denote d_{ij}, d_{ik}, d_{jk} – distance between pairs of clusters C_i, C_j, C_k; these distances are already known. At each iteration, instead of a pair of the closest clusters C_i and C_j a new cluster is formed $C_i \bigcup C_j$.

For the middle bond method, the distance between two different clusters is calculated as the average distance between all pairs combined in a cluster.

To calculate the weight (importance) of a topic (cluster), we calculate the sum of the weights of all the words in a given cluster using the formula:

$$W(C) = \sum W_{wd}, \qquad (3)$$

where

$W(C)$ – the weight of cluster C;

$\sum W_{wd}$ – the sum of the weights of all words in a given cluster.

The percentage (weighted information in the cluster) is calculated by the formula

$$\%W(C) = \left(\frac{W(C)}{\sum W(C)} \times 100 \right), \quad (4)$$

where

$\%W(C)$ – weight percentage of a given cluster C;

$W(C)$ – the weight of given cluster C;

$\sum W(C)$ – sum of weights of all defined clusters.

1.5 Recap process

First, in general terms, we describe the main idea. At this stage, some pre-prepared text fragment is considered Set_1, precisely related to a specific topic. Fragment Set_2 – analyzed fragment. It maps to a fragment Set_1 for relevance to the same topic. In our case, it is a fragment isolated from the original (large) text through clustering, as described in the previous section. The process of collecting all fragments that satisfy a certain criterion of conformity is, in fact, a process of summarizing.

We now turn to a more detailed description of the algorithm. Fragment Set_1 is matches by a directed graph G=(V,E). Let further w_{ij}– edge weight (Vi,Vj)ϵE, if any, and $w_{ij=0}$ otherwise. Consider the undirected graph associated with the graph G in a natural way. The many vertices are the same. Edges arise through "forgetting" about orientation. The edge weight in the new graph will be equal to $Link\_Strength_{ij}$=2 x min(w_{ij}, w_{ji}).. The large value of this quantity means that these two words are repeatedly found side by side in two versions. Namely, the first word precedes the second, and vice versa, the second word precedes the first.

Next, enter the value $Path\_Strength_{ij}$=1/ $Link\_Strength_{ij}$. That is, the transition to inverse values has been completed. Thus, we find that the smaller, the more pronounced the above-mentioned property, that is, the more these words are “connected” with each other within this topic.

Closeness centrality is usually used in the research of social networks and serves as an indicator of the speed with which information spreads on the network from one participant to the rest. As a measure of the distance between two participants, the shortest path along the graph (geodetic distance) is used. So, the participant’s immediate friends are at a distance of 1, friends of friends are at a distance of 2, friends of friends of friends are at a distance of 3, etc. Next, the sum of all distances is

taken and normalized. The obtained value is called the distance of the vertex V_i from other peaks. Proximity is defined as the reciprocal of the distance.

$$C_C(V_i) = \frac{N-1}{\sum_{t \in V \setminus v} d_G(V_i, t)}, \quad (5)$$

where $d_G(V_i, t)$ – shortest way from the top V_i to the top t.

In other words, centrality in proximity allows you to understand how close the member in question is to all other network members. Thus, it is important not only to have direct friends, but also that these friends themselves also have friends.

In our case, we are dealing with texts, and not with social networks. But in both cases, graph-theoretic constructions are used. Therefore, similarly, one can calculate centrality by proximity for any vertex in the undirected graph under consideration. When calculating the geodetic distance (the distance between the edges of the graph), the weights of the edges are used $Path\_Length_{ij}$.

We describe the process of comparing two fragments Set_1 and Set_2. Suppose each of them consists of three sentences::

Set_1	Set_2
A B C D	M N B D
B N D	C O D
W X B C	A B C

Leave only those words that are included in both text fragments, we obtain respectively

Set_1'	Set_2'
A B C D	N B D
B N D	C D
B C	A B C

To each of the text fragments Set_1' and Set_2' matches your undirected graph. Note that the set of vertices of these graphs will be the same, namely $\{A, B, C, D, N\}$. Accordingly, for all vertices V_i it is possible to calculate centrality by proximity in the first column and in the second. Let them $C_C(V_i)_1$ and $C_C(V_i)_2$ respectively.

We calculate the value of the indicator

$$Diff(V_i) = (|C_C(V_i)_2 - C_C(V_i)_1| / C_C(V_i)_1) \times 100, \quad (6)$$

which characterizes how the use of a given word in the contexts of the first and second text fragments is the same or different. Next, you need to determine the threshold, which serves as a criterion for the "uniformity" of the use of the word. For example, as a threshold, you can take the median of quantities $Diff(V_i)$, further, this threshold can be used in the analysis of other fragments.

At the final stage, it is calculated how many words “crossed” the threshold by the formula:

$$Score(Set_1, Set_2) = (Count_{match}(Set_1, Set_2) / Count(Set_1)) \times 100, \quad (7)$$

where $Count_{match}(Set_1, Set_2)$ is number of all words from Set_1, which are simultaneously included in Set_1 and Set_2, and who "crossed" the threshold, that is, they passed a test for the identity of use in both text fragments;

$Count(Set_1)$ – the number of all words from Set_1, which were subjected to analysis, i.e., they just simultaneously enter Set_1 and Set_2.

The final conclusion is is based on how high $Score(Set_1, Set_2)$, i.e., in the end, , the expert chooses the appropriate threshold, allowing for a conclusion that the subject of the fragment Set_2 matches the theme of the reference text fragment Set_1.

1.6 Algorithm Testing

Implementation of the morphological analysis in code required the use of:

- for Kazakh, an algorithm for reducing words to normal form based on the Porter algorithm was proposed and implemented, due to the lack of a suitable algorithm[8];
- for Turkish, the Snowball Stemmer.

The calculation of the weight of the topic is carried out by summing the weights of the words included in it. This task for each word in the general case reduces to solving a system of linear algebraic equations. To solve this system, we used the Eigen linear algebra library for a programming language C++.

When constructing sentence graphs displaying grammatical relationships, the LGP parser is used [9], the result of which is a graph of syntactic links between sentence words. For Kazakh and Turkish, LGP prototypes have been developed that take into account the rules for the formation of words in agglutinative languages, which include all Turkic languages [5,10]. It should be noted that the speed of LGP and prototypes linearly depends on the number of offers. Procession of a 30-page text takes about 30 minutes.

To calculate the distances between the vertices of the graph, the dynamic Floyd-Warshall algorithm is used.

During testing, the developed system sent requests to the search engine www.googleapis.com, from which it received URL lists with their text description. Each provided textual description of the system gave an assessment of compliance with the request in Kazakh and Turkish. The program left relevant links, dropping those that were not relevant for its evaluation. The test results are shown in table 1.

Table 1. Test Results

Request	Language	Total links (snippets) received from the search engine	Number of Relevant Links Approved by the System	The number of relevant links skipped by the system	The number of inappropriate links approved by the system
Мұнда педагог мамандар ақпараттық технологиямен жұмыс жасау негіздерін практикалық түрде меңгереді	Kazakh	130	20	0	0
«Экономика» деген сөз алғашқы ұғымында отбасы шаруашылығын білдірген	Kazakh	167	11	2	2
Ortalıkta dolaşan sıcak petrol paralarını tüketime yönlendirdiler	Turkish	40	5	1	0
Yazımızın birinci bolumunde Dunyadaki ekonomik buhranı inceledik	Turkish	120	12	3	1

For texts in Kazakh, on average, of the 130 links obtained from the search engine, the program identified 11-20 high-quality relevant links, about 2 links the program mistook for relevant and rejected the rest as irrelevant, which corresponded to reality. For texts in Turkish, the results are similar.

1.7 Conclusion

In this paper, we proposed some extension of the Niraj Kumar approach by further examining the syntactic links generated by the Link Grammar Parser system for the Kazakh and Turkish languages. Also important is the fact that the graph is mapped to the text as a whole, and not to a separate sentence, as in [1].

It should be noted, the algorithms for determining topics, classification by topic, etc. have some instability (including the algorithm proposed by us) with respect to subject areas. If there is an established terminology in these subject areas, then the algorithms work quite correctly. Moreover, “playing on the choice of thresholds”, you can see how document classes fall into subclasses, and a tree-like structure is clearly visible. If the terminology is not well-established, then we get incorrect results.

Acknowledgments. The research was supported by the grant of funding of scientific and (or) scientific and technical research for 2018-2020 MES RK (No AP05133550), Integration Project of SB RAS (AAAA-A18-118022190008-8) and RFBR (No 18-07-01457 A).

References

1. Kumar, N., Srinathan, K., Varma, V. Using graph based mapping of co-occurring words and closeness centrality score for summarization evaluation .In: CICLing Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing, 2012, Vol. 2, pp. 353-365.

2. Simon R. Mirrosoft Windows 2000 API SuperBible. - 2000. – 1608 c.

3. Barzilay R., Elhadad M. Text summarizations with lexical chains / Inderjeet Mani and Mark Maybury . In: Advances in Automatic Text Summarization, MIT Press, 1999, pp. 111–121.

4. Page, L., Brin, S., Motwani, R., Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. In: Technical Report, Stanford InfoLab, 1999. URL: http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

5. Batura, T.V., Murzin, F.A., Semich, D.F., Sagnayeva, S.K., Tazhibayeva, S.Zh., Yerimbetova, A.S.. Using the link grammar parser in the study of turkic languages. In: Eurasian journal of mathematical and computer applications vol.4(2) – pp.14 – 22 (2016)

6. Tukeev, U.A., Rakhimova, D.R. Semantic relations in the automatic processing of the text of the Kazakh language. In: Bulletin of KazNTU, Series of mathematics, mechanics, computer science, No. 2., pp. 320–325 (2012).

7. Özlem İstek. A Link Grammar for Turkish. In: Thesis. Bilkent University, 135 p. Ankara, Turkey(2006).

8. Fedotov, A.M., Bapanov, A.A., Nurgulzhanova, A.N., Sagnayeva, S.K., Sambetbayeva, M.A., Tusupov, J.A., Yerimbetova, A.S. Using the thesaurus to develop it inquiry systems.In: Journal of Theoretical and Applied Information Technology, vol.86(1), pp.44–61. (2016).

9. Lafferty, J., Sleator, D., Temperley, D. Index to Link Grammar Documentation [Electronic Resource] . URL: http://www.link.cs.cmu.edu/link/dict/index.html

10. Murzin, F.A., Sagnayeva, S.K., Sambetbaeva, M.A., Yerimbetova, A.S. Agglutinative languages with a link grammar. In: Bulletin of the Kazakh Academy of Transport and Communications,vol.2 (97),pp. 62-67.(2016)