Bull. Nov. Comp. Center, Comp. Science, 38 (2015), 37–49

On some graphs connected with texts in a natural language, link grammar and the summarization process

T. V. Batura, F. A. Murzin, D. F. Semich,
A. M. Bakiyeva, A. S. Yerimbetova

Abstract. The paper describes the generalization of the summarization algorithm of Niraj Kumar. The method proposed in the article uses the Link Grammar Parser. Our investigations are oriented to processing news articles, reviews from social networks, etc. We consider the possibility of applying this algorithm to estimate the relevance of posts published in the Internet to the selected articles published before. This approach is useful in solving the problem of identifying the source of information dissemination.

Keywords: natural language processing, syntactic analysis, Link Grammar Parser, summarization, relevance.

Introduction

We are interested in solving the following problems. In the Internet, there is a huge number of news and review articles on movies, games, digital equipment, etc. After reading this information, social network users share their impressions and opinions about it. To identify the source of information dissemination, it is essential to establish correspondence between the Internet articles and users' comments. In this case, the set of tweets of one or more users (for example, for a certain period) can be considered as a summary of an article or a piece of news. Naturally, such "summaries" may contain information from both the original article and other sources.

Paper [1] investigates Rouge, a popular metric for the evaluation of automatically written summaries. Different versions of automatic methods for evaluating the content of machine-generated summaries are considered in [2–4]. Yet, there is a significant quality gap between these automated metrics and human evaluation. The basis for this paper was [5], in which the authors consider the process of automatic summarization (resuming).

At the first stage, basic themes in the text are allocated and their weights are calculated. The themes are allocated by hierarchical methods, in particular, by the method of a paired average (the pair-group method using arithmetic averages). The theme weight is defined as a sum of all words related to this theme. The weights of the words setting a context (it is possible to consider them as key words) are calculated by means of the reference ranging algorithm (the page rank algorithm) and proceeding from the assumption that a text can be represented in the form of an oriented graph. At the second stage, a text fragment prepared in advance, related to a certain theme, and a fragment of the text received from the system by means of clusterization are considered. These fragments are compared by keywords using the so-called centrality on affinity (closeness centrality). As a matter of fact, gathering all fragments satisfying to a certain accordance criterion is the process of summarization (resuming).

In this way, fragments corresponding to a given theme can be allocated from the text. They do not necessarily follow one another; fragments related to other themes can be inserted in the text between them. Further, the allocated fragments can be united in the summary of the given theme. Generally, several such summaries corresponding to various themes can be chosen from a text.

One of the problems arising is that a permutation of words in a sentence can essentially change its sense, which leads to an incorrect operation of linguistic algorithms with separate keywords, their frequencies, etc. The paper mentioned above offers a method allowing taking into consideration the order of words and shows its efficiency.

In this paper, the authors suggest a generalization of the method described in [5]. This algorithm involves using the syntactic analyzer Link Grammar Parser [6–8].

1. Basic idea of the algorithm

The basic idea of the algorithm is the relevance evaluation. The algorithm includes several steps.

1. Preprocessing the text of an article, removing unsupported elements and special characters.
2. Calculation of the weights of words.
3. Partition of the article into topics. A generalized diagram of the algorithm is shown in Figure 1. Calculation of the weights of topics. A topic is a set of semantically connected sentences (not necessarily consecutive). Let us note that usually larger articles have several topics.
4. Computation of the relevance between the themes given in advance and the topics.
5. Calculation of the final estimation.

A generalized diagram of the algorithm is shown in Figure 1.

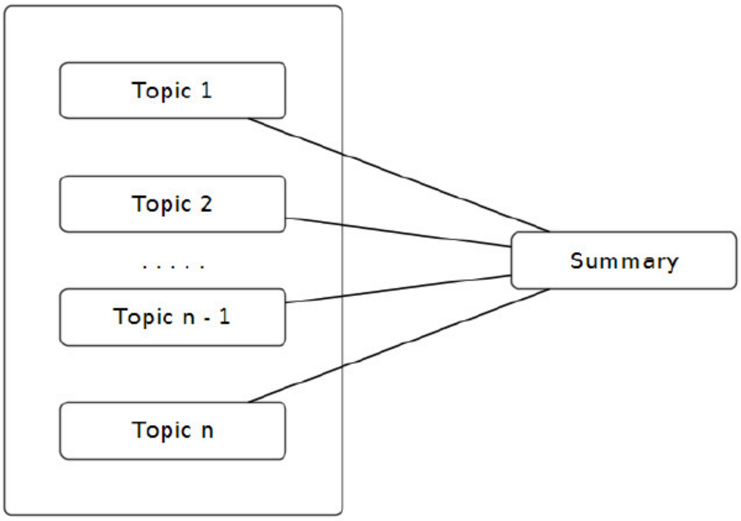


Figure 1. A general diagram of the algorithm

2. Definition of the theme of a text

By a “theme” we mean a set of sentences concerning the same concept, phenomenon, sequence of events, etc. To define the fragments corresponding to a theme contained in a document, various agglomerative procedures of clustering are usually applied [9]. In the English-speaking literature, such a scheme is referred to as the “group average agglomerative clustering scheme” (GAAC).

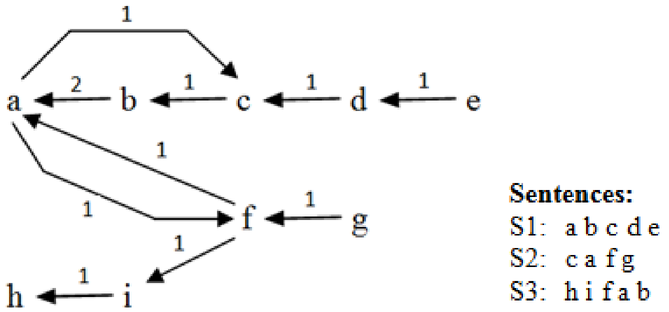
The similarity of the sentences can be estimated as follows:

$$sim(T_i, T_j) = \frac{1}{|T_i \cup T_j|(|T_i \cup T_j| - 1)} \sum_{S_n \in T_i \cup T_j} \sum_{\substack{S_m \in T_i \cup T_j \\ S_n \neq S_m}} sim(S_n, S_m).$$

Each step involves combining the topics that are the most similar. Now, for simplicity, we consider separate paragraphs of the text as topics, i.e. we do not use agglomerative procedures.

3. Calculation of the weights of words

In a general case, we suppose that there is a fragment of the text or a “test” fragment prepared in advance, i.e. a sequence of sentences $S = \langle S_1,\ S_2,\ ...,\ S_n\rangle$. Further, a graph is associated with the sequences S. It is possible to explain the method with a help of an example considered in [5]. The sequence of sentences and the corresponding graph are represented below:



Sentences:
S1: a b c d e
S2: c a f g
S3: h i f a b

Here $a, b, c, d\ldots$ are the words of corresponding sentences. The words of a given set of sentences form a set of the vertices of a given graph. The edge of the graph shows that each word directly follows another. Thus, the direction of the edge orientation is defined as follows: from the subsequent word to the previous. The edge weight is the number of occurrences of a given pair of words following another in the given order in the whole text fragment $S = \langle S_1,\ S_2,\ ...,\ S_n\rangle$.

In [5], the authors assume that the text is preprocessed. Namely, auxiliary syntactic words, such as articles, prepositions, postpositions, particles, and interjections, are taken out. Moreover, the text is passed through a stemmer, and as a result we have, roughly speaking, only roots of the words.

After that, we generalize the algorithm, taking into account the syntactic sentence structure, obtained as a result of the application of the Link Grammar Parser. Also, we consider using the fuzzy logic of Zadeh [10]. Our approach is not to carry out such preprocessing, in particular, not to use stemmers. At the end of sentence comparison, we can take into consideration a larger or a smaller set of these or other links between words, and thus any relation expressed by auxiliary words can be ignored.

The next important thing is the frequency of links (connectors). Also, we can take into account the directions of the links, i.e. the order of words. It is necessary to try to take into account the number of connectors incoming into the vertex and the number of connectors outcoming from the vertex. For this purpose, the concept of the rank of a vertex (the weight of a word) is introduced.

The weights (ranks) of the words can be calculated in different ways, for example, as proposed in [5]:

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

$S(V_i)$ is the rank (weight) of V_i;

$S(V_j)$ is the rank (weight) of V_j, from which a link is directed into the vertex V_i;

$|\,OUT\,(V_j)\,|$ is the number of nodes connected by the edges outgoing from V_j;

N is the number of nodes in a graph;

λ is the “damping” factor; in [5] it is equal to 0.85.

The formula written above means that a vertex with references from other vertices of a higher rank receives a high rank, too. In this manner, we can allocate the most important, in fact, key words in the whole text.

Let us try to analyze some questions connected with rank calculation.

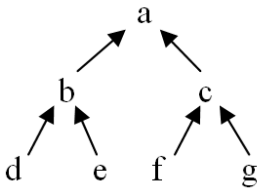
Evidently, if an acyclic graph corresponds to a sentence or a set of sentences, i.e. it is a tree, the ranks can be calculated recursively in the process of bypassing a tree.

In the case when a graph has cycles, we have to solve some systems of equations. Several simple examples are given below. In the first two examples, the ranks are calculated recursively. In the other examples, we have to solve systems of equations.

Example 1.

$$\begin{array}{l} S1:\ a,b,d \\ S2:\ a,b,e \\ S3:\ a,c,f \\ S4:\ a,c,g \end{array}$$

The graph corresponding to the given sentence is represented below:



According to the rank definition, we have the following equalities:

$S(d) = S(e) =\ S(f) = S(g) = \frac{1-\lambda}{7}$, in view of the fact that $IN(x) = \emptyset$ for $x \in \{d, e, f, g\}$.

Further, we obtain

$$S(b) = \frac{1-\lambda}{7} + \lambda\cdot\left(\frac{S(d)}{1} + \frac{S(e)}{1}\right) = \frac{1-\lambda}{7} + 2\lambda\cdot\left(\frac{1-\lambda}{7}\right) = \frac{1-\lambda}{7}(1+2\lambda).$$

It is easy to see that $S(b) = S(c)$.

Now taking into account the above-said, we obtain accordingly

$$S(a) = \tfrac{1-\lambda}{7} + \lambda\cdot\left(\tfrac{S(b)}{1} + \tfrac{S(c)}{1}\right) = \tfrac{1-\lambda}{7} + 2\lambda\cdot S(b) = \tfrac{1-\lambda}{7} + 2\lambda\cdot S(b) =$$
$$= \tfrac{1-\lambda}{7} + 2\lambda\cdot\left(\tfrac{1-\lambda}{7}(1+2\lambda)\right) = \tfrac{1-\lambda}{7}(1+2\lambda+4\lambda^2).$$

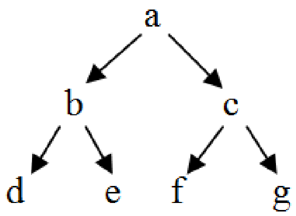
Let us note that the expression in brackets is the sum of a geometrical progression.

The given formulas can be easily generalized for the case of a similar binary tree of a height k, with edges oriented upwards.

Example 2.

$$\begin{aligned} S1 &: \; d, b, a \\ S2 &: \; e, b, a \\ S3 &: \; f, c, a \\ S4 &: \; g, c, a \end{aligned}$$

The following graph corresponds to the given sentence:



According to the rank definition, we have $S(a) = \frac{1-\lambda}{7}$. Further, we obtain

$$S(b) = S(c) = \frac{1-\lambda}{7} + \lambda \cdot \frac{S(a)}{2} = \frac{1-\lambda}{7} + \frac{\lambda}{2}\left(\frac{1-\lambda}{7}\right) = \frac{1-\lambda}{7}(1 + \frac{\lambda}{2}).$$

Further, we have $S(d) = \frac{1-\lambda}{7} + \lambda \cdot \frac{S(b)}{2} = \frac{1-\lambda}{7} + \frac{\lambda}{2}\left(\frac{1-\lambda}{7}(1 + \frac{\lambda}{2})\right) = \frac{1-\lambda}{7}(1 + \frac{\lambda}{2} + \frac{\lambda^2}{4})$.

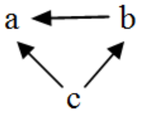
By virtue of symmetry, it is easy to see that $S(d) = S(e) = S(f) = S(g)$.

Let us note that the expression in brackets, as well as in the previous case, is the sum of a geometrical progression. The given formulas can be easily generalized for the case of a similar binary tree of a height k, with edges oriented downwards.

Example 3.

$$S1 : \; a, b, c, a$$

The following graph corresponds to the given sentence:



According to the rank definition, we have the following three equalities:

$$S(a)=\frac{1-\lambda}{3}+\lambda\cdot\frac{S(b)}{1},\ S(b)=\frac{1-\lambda}{3}+\lambda\cdot\frac{S(c)}{1},\ S(c)=\frac{1-\lambda}{3}+\lambda\cdot\frac{S(a)}{1}.$$

Substituting the first equality in the third one, we obtain $S(c)=\frac{1-\lambda}{3}+\lambda\cdot\left(\frac{1-\lambda}{3}+\lambda\cdot S(b)\right)$.

Now, taking into consideration the second equality, we obtain accordingly $S(c)=\frac{1-\lambda}{3}+\lambda\cdot\left(\frac{1-\lambda}{3}+\lambda\cdot\left(\frac{1-\lambda}{3}+\lambda\cdot\frac{S(c)}{1}\right)\right)$.

From this, after algebraic transformations, we have

$$\begin{aligned}S(c)&=\tfrac{1-\lambda}{3}+\tfrac{\lambda(1-\lambda)}{3}+\tfrac{\lambda^2(1-\lambda)}{3}+\lambda^3\cdot S(c)=\\&=\tfrac{(1-\lambda)(1+\lambda+\lambda^2)}{3}+\lambda^3\cdot S(c)=\tfrac{(1-\lambda^2)}{3}+\lambda^3\cdot S(c).\end{aligned}$$

Further, we have $(1-\lambda^2)S(c)=\frac{(1-\lambda^2)}{3}$, which gives $S(c)=\frac{1}{3}$.

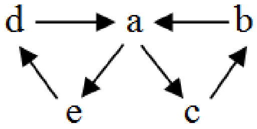
It is easy to see that in view of symmetry, $S(a)=S(b)=S(c)=\frac{1}{3}$.

Analogously, if the sentence looks like $S1:\ x_1,\ldots,x_n,\ x_1=x_n$ and $x_i\neq x_j$ in all other cases, then $S(x_1)=\ldots S(x_{n-1})=\frac{1}{n-1}$.

Example 4.

$$S1:\ a,b,c,a,d,e,a$$

The following graph corresponds to the given sentence:



Let us note that the same graph corresponds to a pair of sentences.

$$S1:\ a,b,c,a$$
$$S2:a,d,e,a$$

By the rank definition, we have the following five equalities:

$$\begin{aligned}&S(a)=\tfrac{1-\lambda}{5}+\lambda\cdot\left(\tfrac{S(b)}{1}+\tfrac{S(d)}{1}\right),\\&S(b)=\tfrac{1-\lambda}{5}+\lambda\cdot\tfrac{S(c)}{1},\ S(c)=\tfrac{1-\lambda}{5}+\lambda\cdot\tfrac{S(a)}{2},\\&S(d)=\tfrac{1-\lambda}{5}+\lambda\cdot\tfrac{S(e)}{1},S(e)=\tfrac{1-\lambda}{5}+\lambda\cdot\tfrac{S(a)}{2}.\end{aligned}$$

It is easy to see that $S(c)=S(e)$, and from this it follows that $S(b)=S(d)$.

Therefore, the first equality can be rewritten in the form $S(a)=\frac{1-\lambda}{5}+2\cdot\lambda\cdot S(b)$.

Further, we obtain

$$\begin{aligned}S(a)&=\tfrac{1-\lambda}{5}+2\cdot\lambda\cdot\left(\tfrac{1-\lambda}{5}+\lambda\cdot S(c)\right)=\\&=\tfrac{1-\lambda}{5}+2\cdot\lambda\cdot\left(\tfrac{1-\lambda}{5}+\lambda\cdot\left(\tfrac{1-\lambda}{5}+\lambda\cdot\tfrac{S(a)}{2}\right)\right).\end{aligned}$$

From this, after algebraic transformations, we have

$$S(a) = \frac{1-\lambda}{5} \cdot (1 + 2 \cdot \lambda + 2 \cdot \lambda^2) + \lambda^3 \cdot S(a).$$

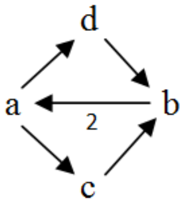
Further, we have$(1-\lambda^2)S(a) = \frac{1-\lambda}{5} \cdot (1 + 2 \cdot \lambda + 2 \cdot \lambda^2)$, which gives $S(a) = \frac{1+2\lambda+2\lambda^2}{5\,(1+\lambda+\lambda^2)}$.

Now we can easily calculate the ranks of other vertices. Note that, unlike the previous case, all of them depend on λ.

Example 5.

$$S1:\ a, b, c, a, b, d, a$$

The following graph corresponds to the given sentence:



Let us note that the same graph corresponds to a pair of sentences.

$$S1:\ a, b, c, a$$
$$S2: a, b, d, a$$

By the rank definition, we have the following four equalities:

$$\begin{aligned} S(a) &= \tfrac{1-\lambda}{4} + \lambda \cdot \tfrac{S(b)}{2}, \\ S(b) &= \tfrac{1-\lambda}{4} + \lambda \cdot \left(\tfrac{S(d)}{1} + \tfrac{S(c)}{1}\right), \\ S(c) &= \tfrac{1-\lambda}{4} + \lambda \cdot \tfrac{S(a)}{1}, \\ S(d) &= \tfrac{1-\lambda}{4} + \lambda \cdot \tfrac{S(a)}{1}. \end{aligned}$$

Hence, it follows that $S(c) = S(d)$ and $S(b) = \frac{1-\lambda}{4} + 2 \cdot \lambda \cdot S(c)$.

Therefore, the first equality can be rewritten as $S(a) = \frac{1-\lambda}{5} + 2 \cdot \lambda \cdot S(b)$.

Further, we obtain

$$\begin{aligned} &S(a) = \tfrac{1-\lambda}{4} + \lambda \cdot \tfrac{S(b)}{2} = \tfrac{1-\lambda}{4} + \tfrac{\lambda}{2}\left(\tfrac{1-\lambda}{4} + 2 \cdot \lambda \cdot S(c)\right) = \\ &= \tfrac{1-\lambda}{4}\left(1 + \tfrac{\lambda}{2}\right) + \lambda^2 \cdot S(c) = \tfrac{1-\lambda}{4}\left(1 + \tfrac{\lambda}{2}\right) + \lambda^2 \cdot \left(\tfrac{1-\lambda}{4} + \lambda \cdot S(a)\right) = \\ &= \tfrac{1-\lambda}{4}\left(1 + \tfrac{\lambda}{2} + \lambda^2\right) + \lambda^3 \cdot S(a). \end{aligned}$$

From this it follows that

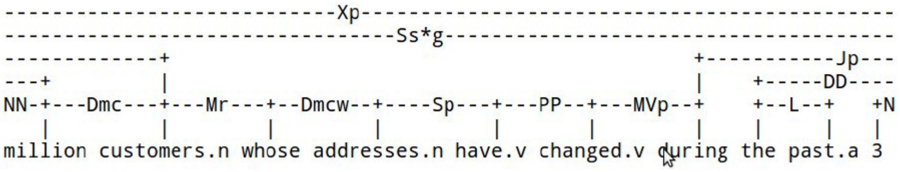
$$(1-\lambda^3)S(a) = \frac{1-\lambda}{4}\left(1 + \frac{\lambda}{2} + \lambda^2\right) = \frac{1-\lambda}{8}(2 + \lambda + 2\lambda^2).$$

After algebraic transformations, we have $S(a) = \frac{2+\lambda+2\lambda^2}{8\,(1+\lambda+\lambda^2)}$.

The ranks of other vertices can be easily calculated from this. Note that all of them also depend on λ.

4. Model of relevance estimation using the Link Grammar Parser

Let us generalize this algorithm and try to take into account syntactic structures of sentences. We will use the Link Grammar Parser before computing the weights of topics. The Link Grammar Parser is a syntactic analyzer for the English language based on an original theory of English syntax. For a given sentence, the program system assigns its syntactic structure as a set of marked links between pairs of words. An example of a sentence parsed by the analyzer is shown in Figure 2. See a detailed description of the Link Grammar Parser in [6–8].

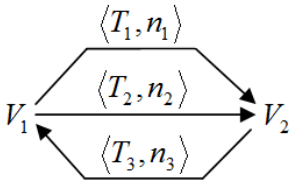


```
----------------------------Xp---------------------------------------------
---------------------------------Ss*g--------------------------------------
-------------+                                            +-----------Jp---
---+         |                                            |    +-----DD----
NN-+---Dmc---+---Mr---+--Dmcw--+----Sp---+---PP--+---MVp--+    +--L--+   +N
   |         |        |        |         |       |        |    |     |   |
million customers.n whose addresses.n have.v changed.v during the past.a 3
```

Figure 2. An example of a parsed sentence

Thus we assign the graph $G_i(V_i, E_i)$ produced by the Link Grammar Parser from the sentence S_i. In this graph, V_i is a set of words and E_i is a set of triplets $\langle v_1, v_2, t\rangle$, where $v_1, v_2 \in V_i$ are vertices and t is the type of a link. Thus we obtain $G_1, \ldots, G_k$, which are the graphs of sentences. The next step is to build a graph $G(V, E)$ by combining the sentence graphs $G_1, \ldots, G_k$. Here $V = \bigcup\limits_{1 \le i \le k} V_i$ is a set of all words from the sentences; E is a set of quadruples $\langle v_1, v_2, t, n\rangle$ where, as previously, $v_1, v_2 \in V_i$ are vertices, t is a link type, and the additional parameter $n = |\{i : \langle v_1, v_2, t\rangle \in E_i\}|$ is the occurrence number of the $\langle v_1, v_2, t\rangle$ triplet.

An example of connections of two vertices is shown below:


$\langle T_1, n_1\rangle$
$\langle T_2, n_2\rangle$
$\langle T_3, n_3\rangle$
V_1
V_2

We assume that there is a positive number α_i corresponding to each connector (link) T_i. This number is called by its weight, or importance. Further, we define:

1) $\mu(T_i, 1) = \alpha_i$,
2) $\mu(T_i, n) = n \cdot \mu(T_i, 1) = n \cdot \alpha_i$,
3) $\mu(T_{i_1}, n_1, \ldots, T_{i_k}, n_k) = \sum\limits_j \mu(T_{i_j}, n_j) = \sum\limits_j n_j \cdot \alpha_j$.

Thus, if there is a set of parallel edges from a vertex V_i into a vertex V_j having marks $< T_{i_1}, n_1 >, \dots, < T_{i_k}, n_k >$, then all of them can be replaced by one edge having the weight equal to $w_{ij} = \mu(T_{i_1}, n_1, \dots, T_{i_k}, n_k)$.

The formula of the calculation of a word rank can be changed appropriately. In this way, it is possible to use the two variants of the formula:

1. $S\left(V_i\right) = \frac{1-\lambda}{N} + \lambda \sum\limits_{V_j \in IN(V_i)} \frac{w_{ji} \cdot S(V_j)}{|OUT(V_j)|}$;
2. $S\left(V_i\right) = \frac{1-\lambda}{N} + \lambda \sum\limits_{V_j \in IN(V_i)} \frac{w_{ji} \cdot S(V_j)}{\left(\sum\limits_{V_k \in OUT\left(V_j\right)} w_{jk}\right)}$.

Let us define $LinkLengh_{ij}$, which will be used as the weights of edges in the final graph. This parameter characterizes the easiness of transition from one word to the other:

$$PathLength_{ij} = \frac{1}{LinkLengh_{ij}}.$$

For each vertex we compute the closeness centrality, which means the inverse value of the average geodesic distance to the other vertices:

$$C_c(v_i) = \frac{N-1}{\sum\limits_{v_j \in V, v_j \neq v_i} d_G(v_i, v_j)}.$$

Further, we compute the relative change of closeness centrality for each node:

$$Diff(v_i) = \frac{|C_c^1(v_i) - C_c^2(v_i)|}{C_c^1(v_i)}.$$

Finally, let us determine the predicate indicating whether the vertices are similarly contained in the graphs for the topic and the selected theme is estimated:

$$Sim(v_i) = (Diff_{avg} < 0.5 \wedge Diff(v_i) < 0.5) \vee \\ \vee (Diff_{avg} > 0.5 \wedge Diff(v_i) < Diff_{avg})$$

To estimate the relevance of a topic to the benchmark, we take the ratio of the simultaneously occurring words for which the predicate is true for all words:

$$Score(T) = \frac{|\{v \in V : Sim(v)\}|}{|Set(T)|}.$$

The final estimation is the weighted sum of estimates for topics:

$$Score = \sum_T W(T) Score(t).$$

Thus it becomes possible to determine whether a message or a sequence of messages correspond to a given article. An additional coefficient can be introduced which helps to estimate the user's attitude to a movie, a game or a piece of news described in the article. If the computed relevance estimation satisfies a given criteria, it enables us to obtain more complete information about the attitude of the user to the material he/she has read. It goes without saying that it is more interesting to consider groups of people than separate individuals. Note that this process can be conveniently parallelized.

5. Model using the fuzzy logic of Zadeh

The formulas considered above can be modified for the case of fuzzy logic [10]. Namely, we suppose that $0 \le \alpha_i \le 1$. Further, it is possible to put, for example, $\mu(T_i, n) = (1 - 1/2^n)\mu(T_i, 1) = (1 - 1/2^n)\,\alpha_i$. The formula arises from the following idea. We assume that if a connector enters once, then $\mu(T_i, 1) = \frac{1}{2}\,\alpha_i$; if it enters twice, then $\mu(T_i, 2) = \left(\frac{1}{2} + \frac{1}{4}\right)\alpha_i$, etc.

Accordingly, we obtain $\mu(T_i, n) = \left(\frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^n}\right)\alpha_i = \left(1 - \frac{1}{2^n}\right)\alpha_i$.

If there are several connectors, the most natural variant is to take their disjunction $\mu(T_{i_1}, n_1, \ldots, T_{i_k}, n_k) = \mu(T_{i_1}, n_1) \vee \ldots \vee \mu(T_{i_k}, n_k)$.

It is also possible to use their average value, but it is less natural. In this case, the value of the increase of a number of connectors can lead to a reduction of the edge weight on the whole, if connectors with small weights are added. On the other hand, when disjunction is used, we cannot take into account the number of connectors. Therefore, an additional specification of the method is needed.

Let us assume that $0 \le \sum\limits_i \alpha \le 1$, i.e. this inequality is similar to Kraft's inequality, well known in information theory. Then we have $\mu(T_i, n) = \left(1 - \frac{1}{2^n}\right)\alpha_i \le \alpha_i$, and as a result, it is possible to use a sum instead of a disjunction. Accordingly, we obtain $w_{ij} = \mu(T_{i_1}, n_1, \ldots, T_{i_k}, n_k) = \sum\limits_j \mu(T_{i_j}, n_j) \le \sum\limits_j \alpha_j \le 1$. Also, it is obvious that $w_{ij} \ge 0$.

If we take into account the definition of a word rank, we can use the first variant of the formula, with the sum replaced by a disjunction. Then the value of a rank also lies in the interval $[0, 1]$.

As a result, we obtain $S(V_i) = \frac{1-\lambda}{N} + \lambda \cdot \bigvee\limits_{V_j \in IN(V_i)} \dfrac{w_{ji} \cdot S(V_j)}{|OUT(V_j)|}$. We note that the given formula can have other variants. For example, the disjunction can be replaced by the operation $x \oplus y = x + y - x \cdot y$.

Closeness centrality $C_C(V_i)$ can be calculated in a standard way and then, to get into the interval $[0,\ 1]$, it should be normalized. As a result, we obtain $\bar{C}_C(V_i) = \frac{C_C(V_i)}{\max_j\{C_C(V_j)\}}$.

6. Conclusion

In this paper, a generalization of Niraj Kumar's summarization algorithm is considered. The algorithm involves using the syntactic analyzer Link Grammar Parser. The purpose of our investigations is to process news articles and reviews from social networks and other sources. In fact, the main problem is to estimate the relevance of texts to a given theme.

In [11–13], the methods for the comparison of sentences in a natural language are given in order to estimate their similarity. To solve this problem, we use semantic-syntactical relations between words obtained with the help of the Link Grammar Parser.

In Niraj Kumar's algorithm, the theme weight is defined as the sum of all words related to this theme. The weights of the words (which can be considered key words) are calculated by the reference ranging algorithm (PageRank algorithm) proceeding from the assumption that the text as a whole can be presented in the form of an oriented graph. It is the most interesting and important part of the algorithm. It works not with the selected sentences but with the whole text, unlike the methods from [11–13].

We have tried to improve Niraj Kumar's method by using the Link Grammar Parser. In [5], only the order of words was considered, whereas we have taken into account syntactic relations. Also, we have modified some formulas for the case of fuzzy logic. Let us note that the algorithm obtained can use other approaches to text semantics and program tools [14] instead of the Link Grammar Parser, which is an important advantage.

References

[1] Loukina A., Zechner K., Chen L. Automatic evaluation of spoken summaries: the case of language assessment // Proc. Ninth Workshop on Innovative Use of NLP for Building Educational Applications, Maryland USA, 2014. – P. 68–78.

[2] Yatsko V.A., Vishnyakov T. N. A method for evaluating modern systems of automatic text summarization // Automatic Documentation and Mathematical Linguistics. – 2007. – Vol. 41, Iss. 3. – P. 93–103.

[3] Newman D., Lau J.H., Grieser K., Baldwin T. Automatic Evaluation of Summaries Using Document Graphs // Proc. HLT '10 Human Language Technologies: The 2010 Annual Conf. of the North American Chapter of the Association for Computational Linguistics, 2010. – P. 100–108.

[4] Rokaya M. Automatic Summarization based on Field Coherent Passages // Intern. J. of Comp. App. – 2013. – Vol. 79, No. 9. – P. 38–44.

[5] Kumar N., Srinathan K., Varma V. Using graph based mapping of co-occurring words and closeness centrality score for summarization evaluation // Proc. 13th Intern. Conf. on Computational Linguistics and Intelligent Text Processing, 2012. – P. 353–365.

[6] Temperley D., Sleator D., Lafferty J. Link Grammar Documentation [Electronic resource]. – 1998. – http://www.link.cs.cmu.edu/link/dict/index.html

[7] Sleator D., Temperley D. Parsing English with a Link Grammar. – Pittsburgh: School of Computer Science Carnegie Mellon University, 1991.

[8] Link Grammar Documentation [Electronic resource]. – 2015. – http://www.abisource.com/projects/link-grammar

[9] Vorontsov K.V. Lectures on algorithms of clusterization and multidimensional scaling [Electronic resource]. – 2010. – http://www.machinelearning.ru/wiki/images/c/ca/Voron-ML-Clustering.pdf (In Russian).

[10] Zadeh L.A. The concept of a linguistic variable and its application to approximate resoning // Information Sciences. – 1975. – Vol. 8. – P. 199–249.

[11] Murzin F., Perfliev A., Shmanina T. Methods of syntactic analysis and comparison of constructions of a natural language oriented to use in search systems // Bull. NCC. Series: Computer Science. – 2010. – Iss. 31. – P. 91–109.

[12] Murzin F., Perfliev A., Shmanina T. Methods of syntactic analysis and comparison of constructions of a natural language oriented to use in search systems // Novosibirsk State University Journal of Information Technologies. – Novosibirsk, 2011. – Vol. 9, Iss. 4. – P. 13–28 (In Russian).

[13] Batura T.V., Murzin F.A., Perfliev A.A., Shmanina T.V. Methods of the Increase of the Efficiency of Information Search on the Basis of Syntactic Analysis. – Novosibirsk: Publishing Company of SB RAS, 2014 (In Russian).

[14] Batura T.V., Murzin F.A. The Machine-Oriented Logic Methods of Representation of Semantics of the Text in Natural Language. – Novosibirsk: Publishing Company of NGTU, 2008 (In Russian).