Visualisation of Russian Newspaper Corpus by Means of Reference Graphs

Dmitry Ilvovsky and Ekaterina Chernyak

National Research University – Higher School of Economics
Moscow, Russia
dilvovsky,echernyak@hse.ru

Abstract. In this paper we present some preliminary results for text corpus visualization by means of so-called reference graphs. The nodes of this graph stand for key words or phrases extracted from the texts and the edges represent the reference relation. The node A refers to the node B if the corresponding key word / phrase B is more likely to co-occur with key word / phrase A than to occur on its own. Since reference graphs are directed graphs, we are able to use graph-theoretic algorithms for further analysis of the text corpus. The visualization technique is tested on our own Web-based corpus of Russian-language newspapers.

1 Introduction

The main idea of any text visualisation technique is to plot important elements of the text (such as key words or key phrases, named entities, or terms). Such pictures can be seen as a tool for text summarization and information extraction / presentation [12]. The most known text visualization technique is tag clouds [7]. The tag cloud shows the key words / phrases (i.e. tags) extracted from a text on a plane. The size of the tag depends on its frequency or any other statistical feature. The majority of text visualization techniques exploit the idea of tag cloud. In [13] the tags extracted from tweets were color-coded according to the politics of the user. Vennclouds, introduced in [3] are an extension of the tag cloud idea. Instead of one tag cloud, a Venncloud presents three tag clouds, which are used to contrast two texts. In [12] the tag clouds are placed inside the nodes of the graph and the nodes are connected by an edge if they have a lot in common. Furthermore, the nodes are sorted according to the time axis. Another extension of the tag cloud idea is the tag graph. To achieve the tag graph one needs to introduce some sort of relation between the tags. For example, in [8] the tags stand for named entities and the edges between them show whether they co-occur.

Our project of text collection visualization follows to this direction. We construct so-called reference graphs, where nodes stand for key words / phrases, which are extracted from the whole collection. There is a directed edge between two nodes A and B if node A refers to node B: $A \implies B$. The referral relation shows that the key word or phrase B occurs with a higher probability if the key word or phrase A occurs in the same text and B is more likely to occur in a document where A is present than its expected value over the entire corpus indicates. We use an asymmetrical association measure to extract referral rules that is based on annotated suffix tree scoring. Hence the reference graph

is a directed graph, which is a very well studied mathematical structure. This gives us plenty of opportunities for further analysis.

2 Data

We choose a number of Russian news portals (“Izvestia”, “Nezavisimaya gazeta”,“Moscow Komsomoltes”, “Kommersant”). We crawled the section of their web-pages devoted to economics. We process and aggregate all the articles published in 2014, which gives us a total of 4061 articles (1109 from “Kommersant”, 1061 from “Izvestia”, 1284 from “Nezavisimaya gazeta”, and 613 from “Moscow Komsomoltes”).

3 Key word and key phrase extraction

First, following [6,9] we form candidate phrases, which satisfy certain part of speech tag patterns, such as NOUN + NOUN or ADJECTIVE + NOUN or NOUN + PREPOSITION + NOUN, etc. The whole list of patterns was adopted from [9]. If we want to extract key words, we restrict ourselves only to nouns. Second, we set a frequency threshold for candidate phrases and select only frequent phrases. We calculate frequency of the candidate phrase in the whole corpus, not in individual texts. Finally, we achieve a list of phrases that satisfy grammar patterns and are frequent enough. We chose the threshold for frequency empirically so that we get the top 250 candidate phrases and the top 100 candidate words. We remove senseless phrases from this list (such as “Izvestia reporter”) manually and consider the remaining key phrases. Since all the texts in the collection belong to the same domain and are written using specific vocabulary, there is no need for a more complex extraction procedure. The replacement of our manual key phrase processing with some computational techniques, which takes a newspaper-specific vocabulary into account, is an important part of future work.

4 Annotated suffix tree (AST) scoring

According to the annotated suffix tree model [11], a text document is not a set of words or terms, but a set of so-called fragments, i.e., sequences of characters arranged in the same order as they occur in the text. Each fragment is characterized by a float number. The greater the number is, the more important the fragment is for the text. An annotated suffix tree (see Fig. 1) is a data structure used for computing and storing all fragments of the text and their frequencies. It is a rooted tree in which:

- every node corresponds to one character;
- every node is labeled by the frequency of the text fragment encoded by the path from the root to the node.

To build an AST, we split the text into relatively short fragments, strings of two to four words, and apply them consecutively to ensure that the resulting AST has a relatively modest size. Our algorithm for constructing an AST [4] is a light modification of the well-known algorithms for constructing suffix trees [5].

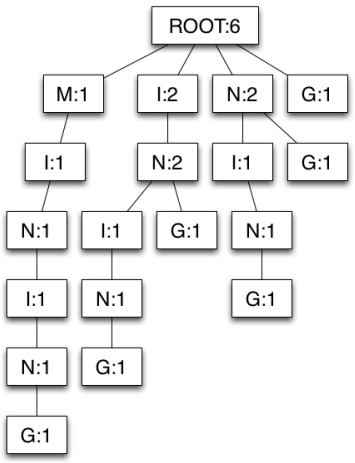

Fig. 1. An AST for string "mining".

To use an AST to score the string to text relevance we need to do the following, First we build an AST for every text. Next we match the strings to the AST to estimate the relevance. This is done in several steps:

1. Every string is split into suffixes;
2. Every suffix is matched to the AST. A match is a path from the root of the AST, that coincides with the beginning of the current suffix. To estimate the match we use scoring function:

$$score(match(\textit{suffix}, ast)) = \\ = \sum_{node \in match} \phi(\frac{f(node)}{f(parent(node))}),$$

   where $f(node)$ is the frequency of the matching node and $f(parent(node))$ is it's parent frequency;
3. Then the relevance is estimated by averaging the score of a symbol:

$$relevance(string, text) = SCORE(string, ast) = \\ = \frac{\sum_{\textit{suffix}} score(match(\textit{suffix}, ast))/|\textit{suffix}|}{|string|},$$

   where $|\textit{suffix}|$ and $|string|$ are the lengths of the suffix and the string.

Note, that "score" is a scaling function that converts a match score into a relevance estimation. We consider three types of the scaling functions, according to [11], where the AST method was used to categorize e-mails:

– Linear function: $\phi(x) = x$

– Logit function:

$$\phi(x) = \log \frac{x}{1-x} = \log x - \log(1-x)$$

– Root function $\phi(x) = \sqrt{x}$

Of them, only the linear scaling function has an obvious meaning: it stands for the conditional probability of characters averaged over matching fragments (CPAMF).

Let us calculate the relevance of string “dining” to the AST in Fig. 1. There are six suffixes of the string “dining”: ‘dining”, “ining”, “ning”, “ing”, “ng”, and “g’ . The scorings of these suffixes are presented in Table 1.

Table 1. Computing the string “dining” score

Suffix	Match	Score
“dining”	None	0
“ining”	“ining”	$\frac{1/1+1/1+1/2+2/2+2/6}{5} = 0.76$
“ning”	“ning”	$\frac{1/1+1/1+1/2+2/6}{4} = 0.71$
“ing”	“ing”	$\frac{1/2+2/2+2/6}{3} = 0.61$
“ng”	“ng”	$\frac{1/2+2/6}{2} = 0.41$
“g”	“g”	$\frac{1/6}{1} = 0.16$

We have used the linear scaling function to score all suffixes of the string “dining”. Now, to get the final relevance value we sum and average them:

$$relevance(dining, mining) = \frac{0 + 0.76 + 0.71 + 0.61 + 0.41 + 0.16}{6} = \frac{2.65}{6} = 0.44$$

5 Reference graph construction

The reference graph construction method is based on the procedure of scoring relevance of a key phrase to a text. Because the key phrases are extracted from the whole collection, we do not know how relevant they are to individual texts. We use annotated suffix tree (AST) scoring to compute key phrase to text relevance in the same fashion as it is presented in [11]. This scoring takes all fuzzy matches between the key phrase and the text into account. It helps to cope with some typos and replaces stemming in a sense. Using AST scoring we estimate the relevance of every key phrase to every text. If the relevance value is lower than the given threshold, we suppose the text is not about this particular key phrase. Usually we set up the relevance threshold at the level of 0.2, which makes up around a third of the maximum experimental AST relevance value. Given the relevance threshold we define the set of texts, which are relevant for every key phrase. Let us denote key phrases as k_i, $i = 1 : n$, and let $F(k_i)$ be the set of texts relevant to key phrase k_i. Let us consider that key phrase k_i refers to key phrase k_j ($k_i \implies k_j$), if the number of texts which belong both to $F(k_j)$ and $F(k_i)$ makes out a significant part of $F(k_j)$: $\frac{|F(k_i) \cap F(k_j)|}{|F(k_i)|} > r$, where r is the confidence threshold and

belongs to the (0.5, 1) interval. This gives us the structure of the referrals between key phrases which can be represented as a graph where nodes are key phrases and edges are referral. We also introduce the support threshold in a way similar to association rule framework [1]: $Support(F(k_i)) = |F(k_i)|$ and use for further analysis only those key phrases, whose support values are higher than the given threshold. From the associative rule framework we inherit the problem of the confidence and support thresholds selection. Both are very important, but there is no technique to set them automatically. The association rules are found with user defined minimum support and confidence values. So do we. We set the relevance threshold at 0.2, the confidence threshold at 0.7 and the support threshold at 5.

6 Reference graph visualization

As soon as we get the set of referrals $k_i \implies k_j$ and their confidence and support values, we can plot the reference graphs. For the sake of space we replace key words / phrases with their index numbers. The size of the node depends on the support value. The nodes are color-coded in the following way: the green nodes only refer to other nodes, the violet are only referred by other nodes, the rest of the nodes are blue.

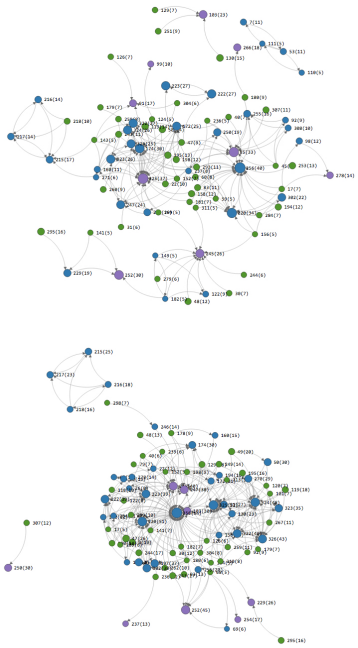

Fig. 2. References graphs for “Moscow komsomolets” and “Nezavisimaya gazeta”, December 2014.

Let us consider two reference graphs constructed for two newspapers 'Moscow Komsomolets" (upper Fig. 2) and "Nezavisimaya gazeta" (lower Fig. 2), based on articles published in December 2014. First of all, the graph are of similar size: there are 88 nodes and 85 nodes correspondingly. The graphs are of different shapes: the first one is centered around the node 256 ("Russian Government"), that has the highest support. The second graph is sparse and there is no obvious center. The highest support get the nodes 256 ("Russian Government") and 325 ("Economic growth"). Both graphs share in common a strongly connected component of four nodes 215, 216, 217, 218, that describes consumer behavior ("Consumer price", "Consumer credit" , "Consumer demand" , "Consumer lending"), which is no surprise. However there is little intersections in the content of the graphs. The nodes "Vladimir Putin" and "Dmitry Medvev" are absent in the second graph. There are four nodes that deal with Ukraine in the first graph, and only two of them appear in the second. At the same time, there is a node "Saudi Arabia" in the second graph. The majority of nodes in the second graph are Russian Government and Ministry of Finance related, while in the first graph the majority of nodes relate to ruble devaluation and business in Russia. These two graphs clearly show the difference between two newspapers. "Nezavisimaja gazeta" being more politics and business oriented presents the year end situation in Russia as crisis, while the "Moscow Komsomolets" is more oriented towards international relations of Russia and consumer needs.

7 Analysis of reference graphs

There are several ways of reference graph analysis. such as link analysis and extension of reference graphs to time-depended case. The most straightforward way to analyse the structure of the directed edges and to measure centrality of the nodes of the reference graph is to apply the PageRank algorithm[10]. The list of the top 5 nodes according to PageRank of "Nezavisimaya gazeta" and "Moscow komsomolets" reference graphs are presented in Tables 2 and 3.

Node (Key word/phrase)	**PageRankValue**
"Russian market"	0.107
"Economic sanctions"	0.063
"Consumer demand"	0.047
"Consumer credit"	0.042
"Economic policy"	0.035

Table 2. Top 5 nodes according to PageRank of "Nezavisimaya gazeta" reference graph

"Economic sanctions" seem to be the most important event of the December, 2014, widely discussed in both newspapers under consideration. Despite the attitude to economics sanctions might be different, the consumer demand is usually discussed in context of possible consequences. The link analysis confirms the fact that "Nezavisimaya

Node (Key word/phrase)	PageRankValue
“World economy”	0.051
“Russian ruble”	0.049
“Economic sanctions”	0.042
“Consumer demand”	0.036
“Minister of Finance”	0.029

Table 3. Top 5 nodes according to PageRank of “Moscow komsomolets” reference graph

gazeta” is more oriented to market situation, while “Moscow komsomolets” looks at the situation from the world-wide perspective. To analyse how the reference graphs change with time we need to construct a series of reference graphs. Let us split the 2014 year in 25 periods, so that every period lasts for 2 weeks. Now we can construct 25 reference graphs, using the same key words / phrases and subcorpus of the articles published in the given period. Let us focus on the “Izvestia” newspaper. First of all we check, which key words / phrases are relevant (i.e. have high support that is document frequency) to each period. To do this we need to merge all the articles of the period into one text and apply the AST relevance measure. Surprisingly this gives us a simple typology of the key words / phrases. There are key words / phrases that are relevant to all periods, such as “Gas supply” . There are key words / phrases that are relevant for several consequent periods. For example, “Tax concession” is relevant to periods 8-13, that is for the middle of the year (after annexation of Crimea), “Weakening rouble” is relevant to periods 16-20. Finally, there are key words / phrases that are relevant to the beginning and to the end of year periods: “Rate increase” , “the Customs Union” , “Oil price”, “Gasprom” . This might be explained by planning and wrapping up the year using the same vocabulary.

The next step is to check which referrals are present during the whole year and which appear / disappear in some periods. Such referrals are “State Duma” $\Longrightarrow$ “Bill” , “Vladimir Putin” $\Longrightarrow$ “Russian rouble” , “Russian gas” $\Longrightarrow$ “Ukraine government” are present in almost all 25 reference graphs. Closer to the end of the year appear such referrals as “Inflation” $\Longrightarrow$ “Russian currency” . There are some referrals that are appear in disjoint periods. For example, the referral “Retail” $\Longrightarrow$ “Criminal responsibility” appears in periods 3-4, 8, 11-12, 17, 19, 23. Since we plan to make an animation of how reference graphs change with time, such referrals might cause a lot of difficulties. We cannot animate straightforwardly the reference graphs and need some sort of smoothing.

8 Future work

Analysis of reference graphs. It is necessary to test some methods for graph analysis such as clustering nodes, measuring centrality (ingluding PageRank and HITS), finding cycles of minimal length, bridges, connected components.

Temporal analysis. The extension of reference graphs to time-depended case will allow us to detect trends and / or events in newspapers by finding temporal references between key word or phrases that occurred yesterday and today’s key word or phrase.

Coloring the nodes. We plan to use the LDA [2] or LDA-like methods to group key phrases into latent topics and color the nodes according to the topic mixture.

Text preprocessing improvement. Further directions of text processing module development include word sense disambiguation and disambiguation for morphological analysis. We need to develop some filters that distinguish between newspaper-specific vocabulary and general vocabulary and take synonyms such as "Russian currency" and "Rouble" into account.

Quantitative evaluation. Since there is no golden standard for text visualisation problems, it is a common idea to conduct a user study for quantitative evaluation. There are three possible designs of the study. To test the information discovery power of referral relation $A \implies B$ we ask the user "What is the context of concept B?" and provide 4 possible answers: concept A (right answer), concept C, such that $B \implies C$, concept D, such that $D \implies A \implies B$, a random concept. The proportion of right answers will show the validity of referral relation. To test the reference graphs as a search tool, we provide the user with a search engine which is able to search in our text collection, a series of tag clouds for every time period and every source and corresponding series of referral pairs. We ask the user to find concepts, associated to his/her own query. Using only the search engine he/she will hardly find any associated concepts. By looking at tag clouds the user might discover some concepts that are related to the search. But the reference graphs will provide not only the concepts the query refers to or is referred by, but also the concepts achieved according to transitivity of referral relation. To validate this we a) record the average time of determining the associated concepts, b) ask the user to estimate the ease of finding associated concepts by means of every tool. To test the temporal reference graphs, we ask the user to search for his/her own query and understand since when his/her query is important. To answer the question the user may read all the text, rely on the size of the tags of the tag clouds or check, whether the query appears in a certain reference graph. Since the key word / phrase appears in the reference graph, in and only if it has high support and is a part of a reference rule with a high confidence, using the reference graphs might help the user a lot. We use the same validation techniques as we use for the previous test.

9 Conclusion

In this paper we tried to analyse a corpus of articles published in the most popular Russian newspapers in 2014 by means of so-called reference graphs. Every node of a reference graph is a key word or key phrase. The edges of the graph represent reference relation, which means that if node A refers to node B, B is more likely to co-occur with A. The reference graphs can serve not only as a visualization tool, but also as a tool for further text analysis.

References

1. Agrawal, R., T., I., A., S.: Mining association rules between sets of items in large databases. ACM SIGMOD Record 22(2) (1993)

2. Blei, D.M., Ng, A.Y., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research 3(4-5), 9931022 (2008)
3. Coppersmith, G., Erin, K.: Dynamic wordclouds and vennclouds for exploratory data analysis. Association for Computational Linguistics pp. 22–29 (2014)
4. Dubov, M.: Text analysis with enhanced annotated suffix trees: Algorithms and implementation. In: Analysis of Images, Social Networks and Texts, Communications in Computer and Information Science, vol. 542, pp. 308–319. Springer International Publishing (2015)
5. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, New York, NY, USA (1997)
6. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing. pp. 216–223 (2003)
7. Hulth, A.: Tag clouds: Data analysis tool or social signaller? In: Proceedings of the Hawaii International Conference on System Sciences (2008)
8. Lloyd, L., Kechagias, D., S, S.: Lydia: A system for large-scale news analysis. Springer Berlin Heidelberg (2005)
9. Mitrofanova, O., Zaharov, V.: Automatic analysis of terminology in the russian text corpus on corpus linguistics (in russian). In: Conference Proceedings Computational Linguistics and Intellectual Technologies (2009)
10. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Stanford InfoLab (1999)
11. Pampapathi, R., Mirkin, B., Levene, M.: A suffix tree approach to anti-spam email filtering. Machine Learning 65(1), 309–338 (2008)
12. Shahaf, D., Yang, J., Suen, C., Jacobs, J., Wang, H., Leskovec, J.: Information cartography: Creating zoomable, large-scale maps of information. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1097–1105. KDD '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2487575.2487690
13. Wang, H., Can, D., Kazemzadeh, A., Bar, F., Narayanan, S.: A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In: Proceedings of the ACL 2012 System Demonstrations. pp. 115–120. ACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012), http://dl.acm.org/citation.cfm?id=2390470.2390490