Visualization of Dynamic Reference Graphs

Ivan Rodin and **Ekaterina Chernyak** and **Mikhail Dubov** and **Boris Mirkin**
National Research University Higher School of Economics
Moscow, Russia
ivvrodin@gmail.com

Abstract

We present a tool for dynamic reference graph visualization. A reference graph is a graph based on key phrases retrieved from a time-indexed natural language text corpus. This tool may be useful for the analysis of connected pairs of latent topics, changes in the significance of these topics as well as in the relationship between them over various time periods.

1 Introduction

"Text visualization" is a rather ambiguous term. One of the approaches to text visualization is scene generation, as described in (Chang et al., 2014). In our work, however, text visualization has a different meaning. Our interest lies in plotting meaningful elements from texts such as key phrases, named entities or terms. Retrieved plots may serve as a tool for information extraction and summarization. As shown in (Kucher and Kerren, 2015), there are many techniques for textual data analysis and visualization, and their number is rapidly growing.

In fact, the problem of text visualizaton can be divided into two subproblems: visualization of static and temporal textual data. The most known static text visualization technique is called "tag clouds" (Coupland, 1996). Many visualization techniques extend the idea of a tag cloud. For instance, in (Greene et al., 2015) key phrases are used to construct a concept lattice for a dataset of publications. Lattice is visualized as an interactive tag cloud browser. In (Coppersmith and Erin, 2014), tags extracted from tweets are colored according to the political preferences of tweet authors. Vennclouds, introduced in (Wang et al., 2012) present another extention of tags clouds, comparing two texts by showing three tag clouds: one cloud containing tags from the first text, another one with tags from second text, and the third cloud showing words and phrases that appear in both texts.

Approaches based on tag cloud construction are also quite successful in the visualization of temporal textual data. For example, the ThemeRiver visualization tool, introduced in (Havre et al., 2000), shows thematic variations over time within a large collection of documents. In (Shahaf et al., 2013), tag clouds are placed inside graph nodes. The nodes are connected by an edge if they have a lot in common, and are placed on the time axis as well.

Tag graphs are another extension of tag clouds. To construct a tag graph, one needs to introduce some sort of relation between tags. For example, in (Lloyd and D. Kechagias, 2005), the tags stand for named entities and one draws edges between tags that co-occur. (Berendt and Subasic, 2009) present a dynamic visualization technique for these temporal co-occurence graphs.

In this paper, we suggest an approach to temporal textual data analysis which is based on dynamical reference graphs. The main difference of these reference graphs from co-occurence graphs is that they are oriented. This allows to analyze how one term "refers" to another, as well as to retrieve more patterns describing the relations between terms. Our visualization also allows to analyze how the significance of certain terms changes over time.

Proceedings of the 2016 Workshop on Graph-based Methods for Natural Language Processing, NAACL-HLT 2016, pages 34–38,
San Diego, California, June 17, 2016.

2 Dynamic reference graph

To build a dynamic reference graph for a text collection, one can use Algorithm 1.

Algorithm 1: Dynamic reference graph construction

Input : Time-indexed corpus of text documents and a list of key phrases.
Output: Dynamic reference graph.

1 Divide the corpus evenly into T sequential time periods $\tau = 1..T$.
2 Set a list of N key phrases (concepts) $w_i, i = 1..N$.
3 **for** $\tau = 1..T$ **do**
4 Extract the information about static weighted oriented reference graph G_τ that should include: 1) significance (the support value) of each concept in period $S_\tau(w_i)$, which is the number of documents where the concept appeared; 2) reference significance levels $C_\tau(w_i, w_j)$, $C_\tau(w_j, w_i)$ for all pairs of concepts w_i, w_j.
5 **end**
6 Define the lower threshold for support and confidence levels. Draw a reference graph G_1 for $\tau = 1$ with respect to these thresholds.
7 **for** $\tau = 2..T$ **do**
8 Remove edges from $G_{\tau-1}$ are not in G_τ
9 Remove nodes from $G_{\tau-1}$ are not in G_τ
10 Add nodes from G_τ that are not in $G_{\tau-1}$
11 Add edges from G_τ that are not in $G_{\tau-1}$
12 **end**

Initial steps. The input to our system is a time-indexed corpus. The system divides documents into several time periods assuming that the number of documents in each time period is commensurable.

Next, we should define a list of concepts. They can be chosen just as the top frequent terms from the corpus. Sometimes it also makes sense to set the list of concepts manually. For example, if we want to analyze some specific topics from a corpus of newspaper articles, we can set a list of keywords to monitor and analyze, for example, only economical or technological news. This approach can be also useful in the analysis of the interaction between characters in fiction books where the concepts are just their names. Other possible approaches for term extraction are presented in (Siddiqi and Sharan, 2015)

Building G_τ. Concepts define the nodes of G_τ. Weighted oriented edges of this graph are defined by the co-occurrence of these concepts. We also retrieve and store information about the significance of each concept in all time periods.

Support estimation. Let us define the significance of a concept w_i in time period τ as $S_\tau(w_i) = |D_\tau(w_i)|$, where $D_\tau(w_i)$ is a set of documents in τ where concept w_i appears more than σ times. By default, we take $\sigma = 0$, but in some cases that threshold may be increased.

Confidence estimation. Once we know the support values for each concept w_i in τ, we can compute confidence values for the connections between pairs of concepts with non-zero support values. We say that concept w_i "refers" to concept w_j in time period τ with confidence $C_\tau(w_i, w_j)$, where $C_\tau(w_i, w_j) = \frac{|D_\tau(w_i) \cap D_\tau(w_j)|}{|D_\tau(w_i)|} \in [0; 1]$.

Having computed these confidence levels for connections between concepts in time period τ, we can finally build a static oriented weighted graph G_τ for that time period. An edge from concept w_i to concept w_j with weight $C_\tau(w_i, w_j)$ in this graph means that w_i refers to w_j with confidence $C_\tau(w_i, w_j)$. We can also specify the threshold for confidence which defines wheter or not edges should be displayed. Note that values of $C_\tau(w_i, w_j)$ close to 1 mean that if w_i occurs in a document, then w_j usually occurs as well. In other words, w_j tends to occur in the context of w_i.

3 Datasets

Our test collection corresponds to the topic of newspaper analysis. We have collected a set of articles on economics from four Russian newsportals ("Izvestia", "Kommersant", "Moscow Komsomoltes", "Nezavisimaya gazeta") that were published in 2014. This corpus (called "RuNeWC" – Russian Newspaper Web Corpus) contains 4061 Russian language articles, divided into 26 time periods. Every period has a length of 2 weeks.

Concepts were extracted automatically using the strategy proposed in (Hulth, 2003), which consists of 2 steps. The first step was to extract candidate words and phrases that satisfy certain speech patterns, adopted from (Mitrofanova and Zaharov, 2009) (ex.: ADJECTIVE+NOUN, NOUN+NOUN, etc.). The next step was to form the list of concepts, which resulted in 250 most frequent phrases

and 100 most frequent words. Finally, we manually removed some concepts that are not semantically important (ex.: “Kommersant reporter” [“Korrespondent Kommersanta”]). For graph visualization, we set the confidence threshold at 28 and support threshold at 0.9. An example of dynamic graph visualisation for RuNeWC is presented in fig. 1.

In order to provide an even more clear example, we created an English language corpus based on four books from a series of popular epic fantasy novels “A Song of Ice and Fire” (ASOIAF) written by George R. R. Martin. If we take the list of characters of that novel as an initial set of concepts the reference graph then shows the co-occurence of characters in the book pages, typical clusters of characters, the significance of characters, as well as the change of these parameters over the time.

The plot in the first four books develops linearly: every next chapter describes actions that occurred after the actions of a previous one. To construct our corpus, we divided each chapter into several equal parts (2–7 parts, depending on the size of that chapter). Each part was then taken to the corpus as a single document. Then, the set of all documents was divided into groups of 50 sequential documents, forming 14 time periods. We didn’t take one chapter as a single time period, as the story of each chapter goes on behalf of one specific character, and he automatically becomes the most important one. In constrast, the division of each chapter into several documents increased the precision of our support and confidence estimations for the concepts from the book.

In our experiment, we set the term frequency threshold $\sigma = 1$, i.e. considered a concept to be mentioned in a document if it appeared in that document two or more times. The threshold for the support value was set at the level of 5, and minimum confidence value to 0.3.

As a result, we obtained a dynamic reference graph visualization that may serve as a schematic retelling of “Song of Ice and Fire” novel. Examples of dynamic graph visualisation for ASOIAF corpus are presented in figs. 2 to 4[1].

[1]Visualization video is available at https://youtu.be/UaUGVPTdM-w.

4 Visualization

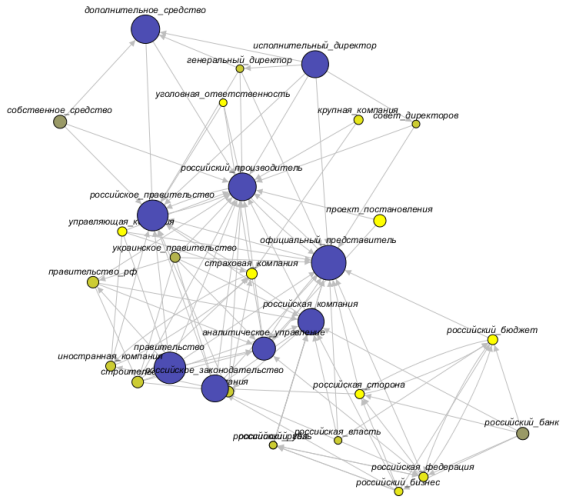

Figure 1: RuNeWC reference graph at time period $\tau = 9$ (weeks 17-18 in 2014)

There are several tools for dynamic graph visualization, such as GraphStream[2], KeyLines[3], Gephi[4]. For our task, we decided to use GraphStream as it is a well-documented Java Library which was specifically created for dynamic graphs editing and visualization. GraphStream is also effective in visualization of large graphs with thousands of nodes as it allows to create zoomable interface.

Our software tool can animate dynamic reference graphs based on their textual description. It ensures that the layout of a reference graph G_τ for time period $\tau > 1$ depends on the layout of $G_{\tau-1}$. It also takes into account the information about the support of concepts: the greater the support, the bigger the corresponding node. The “age” of concepts (number of periods where these concepts consecutively appear) is quite important as well: new nodes are marked with yellow, and when they become older, they gradually turn into blue. Moreover, if a concept appears for the first time (not after a break), the corresponding node receives a wide black border.

Our implementation comes with a graphical user interface. At any time step, the user can pause the animation process to explore the graph structure. All nodes of the input graph can be moved by simple

[2]http://graphstream-project.org/

[3]http://cambridge-intelligence.com/keylines/

[4]https://gephi.org/

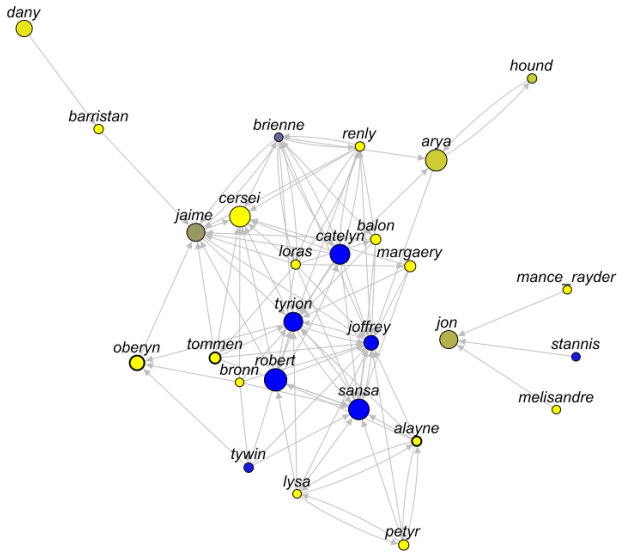

Figure 2: This is dynamic graph at time period $\tau = 11$, which corresponds to the end of the third book. Color and border differentiation of nodes can show the age of concept in graph. For ex.: Oberyn is a character that appears for the first time, as corresponding node is yellow with wide border, while "Tyrion" and other three nodes with deep blue colour have been presented in all periods from the very beginning.

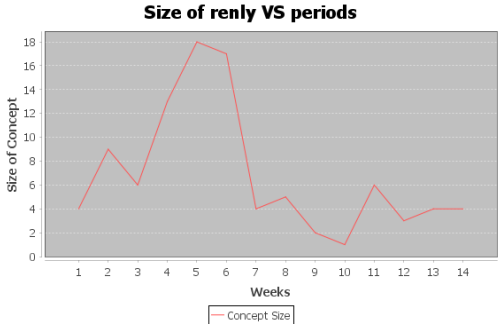

Figure 3: The change of support for concept *Renly* during the time periods.

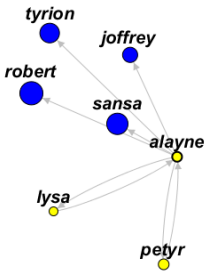

Figure 4: As node of graph is clicked only nodes connected with chosen one are displayed.

drag and drop actions. Our software also handles node clicks: once a node is clicked, only those edges and nodes that are adjacent to it are displayed. Moreover, the information about the support of the corresponding concept during all time periods is displayed in a separate window.

5 Future Work

Let us mention some future directions of our work:

Integration. We are going to integrate our visualization software with a corpus browser that will allow users to generate and visualize their own corpora. To achieve that, we will have to build a unified system that will be responsible for both graph construction and visualization. The concept extraction system will be improved as well: it is possible to use the annotated suffix tree method (Dubov, 2015) for automatic concept extraction in case the user does not provide any.

Analysis of connections. The system presented in this paper can be improved by adding support for contextual synonym extraction, as, for example, *Petyr – Lord Baelish – Littlefinger* in the ASOIAF corpus or *Russian president – Vladimir Putin* in the RuNeWC corpus. We will also compare presented mesure for confidence estimation with other similarity measures.

Graph analysis. Temporal cluster analysis can be particularly informative. Retrieving and highlighting of strongly connected components would allow users to detect some strong ongoing trends.

Graph layout. The layout of dynamic graphs still can be improved. In particular, we have to solve the problem of overlapping nodes that occasionally appears in our software.

Acknowledgements

We would like to express our sincere gratitude to Dmitry Ilvovsky, Anna Shishkova and Maxim Yakovlev, members of research and study group "Methods of web corpus texts analysis and visualisation" for their large contribution to this work. This study (research grant No 15-05-0041) was supported by The National Research University Higher School of Economics Academic Fund Program in 2015.

References

B. Berendt and I. Subasic. 2009. *Stories in time: A graphbased interface for news tracking and discovery*. In WI-IAT09.

A. X. Chang, M. Savva, and C. D Manning. 2014. *Semantic parsing for text to 3d scene generation*. In Proceedings of the ACL 2014 Workshop on Semantic Parsing.

G. Coppersmith and K. Erin. 2014. *Dynamic Wordclouds and Vennclouds for Exploratory Data Analysis.* Association for Computational Linguistics.

D. Coupland. 1996. Microserfs, Flamingo.

Mikhail Dubov. 2015. Text analysis with enhanced annotated suffix trees: Algorithms and implementation. In *Analysis of Images, Social Networks and Texts - 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9-11, 2015, Revised Selected Papers*, pages 308–319.

Gillian J Greene, Marcel Dunaiski, Bernd Fischer, Dmitry Ilvovsky, and Sergei O Kuznetsov. 2015. Browsing publication data using tag clouds over concept lattices constructed by key-phrase extraction.

Susan Havre, Beth Hetzler, and Lucy Nowell. 2000. *ThemeRiver: Visualizing Theme Changes over Time.* Proceedings of the IEEE Symposium on Information Vizualization 2000.

A. Hulth. 2003. *Improved automatic keyword extraction given more linguistic knowledge*. The conference on Empirical methods in natural language processing.

K. Kucher and A. Kerren. 2015. *Text visualization browser: A visual survey of text visualization techniques*. Proceedings of IEEE Pacific Visualization Symposium (Visualization Notes), Englewood Cliffs, NJ.

L. Lloyd and S. Skiena D. Kechagias. 2005. *Lydia: A system for large-scale news analysis, String Processing and Information Retrieval*. Springer Berlin Heidelberg.

O. A. Mitrofanova and V. P. Zaharov. 2009. *Automatic Analysis of Terminology in the Russian Text Corpus on Corpus Linguistics [Avtomatizirovannyy analiz terminologii v russkoyazyichnom korpuse tekstov po korpusnoy lingvistike],*. Dialog, available at: http://www.dialog-21.ru/digests/dialog2009/materials/.

Dafna Shahaf, Jaewon Yang, Caroline Suen, Heidi Wang Jeff Jacobs, and Jure Leskovec. 2013. *Information cartography: creating zoomable, large-scale maps of information*. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.

S. Siddiqi and A. Sharan. 2015. Keyword and keyphrase extraction techniques: A literature review. *International Journal of Computer Applications*, 109(2).

H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. 2012. *A system for realtime twitter sentiment analysis of 2012 US presidential election cycle.* Proceedings of the ACL 2012 System Demonstrations. Association for Computational Linguistics.