

# Loopo: Integrated Text Miner for FACT-Graph-Based Trend Analysis

Ryosuke Saga<sup>1</sup>, Hiroshi Tsuji<sup>2</sup>, and Kuniaki Tabata<sup>1</sup>

<sup>1</sup> Kanagawa Institute of Technology, School of Information Technology,  
1030 Shimo-ogino, Atsugi, Kanagawa, 243-0292, Japan

<sup>2</sup> Osaka Prefecture University, Graduate School of Engineering,  
1-1 Gakuen-cho, Nakaku, Sakai, 559-8531, Japan

{saga, tabata}@ic.kanagawa-it.ac.jp, tsuji@cs.osakafu-u.ac.jp

**Abstract.** This paper proposes an integrated tool to analyze trend visualization graph called “FACT-Graph”. FACT-Graph is generated from text data with time stamp and is useful for trend analysis. However, it faces three key problems: First, it is difficult to configure parameters (such as analysis span, exceptive keywords and thresholds) to generate FACT-Graph; Second, a FACT-Graph does not provide the required information and interface for trend analysis because the process of generating the FACT-Graph eliminates that information; and third, it cannot reflect a user’s awareness in a FACT-Graph. In order to solve these problems, the authors have developed a tool called “Loopo”. Loopo integrates a term database, analysis components, and a graph-drawing function and provides users (i.e., analyzers) with information for trend analysis. Loopo also provides an interactive GUI for configuring parameters at ease and to reflect a user’s awareness in a FACT-Graph instantly.

**Keywords:** Keyword Visualization, Trend Analysis, Co-occurrence graph, Analysis Tool.

## 1 Introduction

The application and utilization of information has become a vital service as the volume of information stored in companies and organizations increases. Such information includes records such as POS data and access logs as well as text data such as questionnaires and reports. This information is stored in a “data warehouse”, and data mining and text mining are applied to it in order to discover useful knowledge [1].

The researched of text mining cover a wide range of areas, for example, keywords extraction, summarization, visualization [2,3]. Especially for the text mining using time-series text data, we developed a visualization technique called FACT-Graph to analyze trends [4]. FACT-Graph extracts keywords from time-series text data and visualizes trends by co-occurrence graph based on the change of attributes. Using a FACT-Graph, we can get macro trends and topics that consist of plural keywords. The usefulness of analysis by FACT-Graph is confirmed in several references [4,5].

To comprehend a FACT-Graph, the user (i.e., analyzer) is shown clues regarding trends from keywords that look important to them. However, FACT-Graph itself doesn't

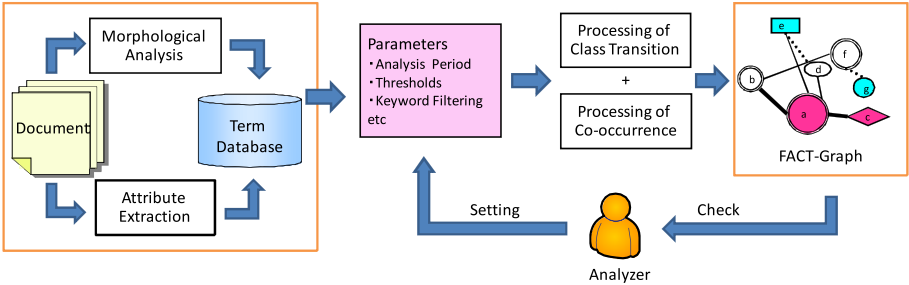


**Table 1.** Keyword Class based on TF and DF

		DF	
		High	Low
TF	High	Class A (Major Word)	Class C (Domain Word)
	Low	Class B (Complementary Word)	Class D (Minor Word)

**Table 2.** Transition of Keyword Class

		After			
		A	B	C	D
Before	A	Hot	Cooling	Bipolar	Fade
	B	Common	Universal	-	Fade
	C	Broaden	-	Locally Active	Fade
	D	New	Widely New	Locally New	Negligible

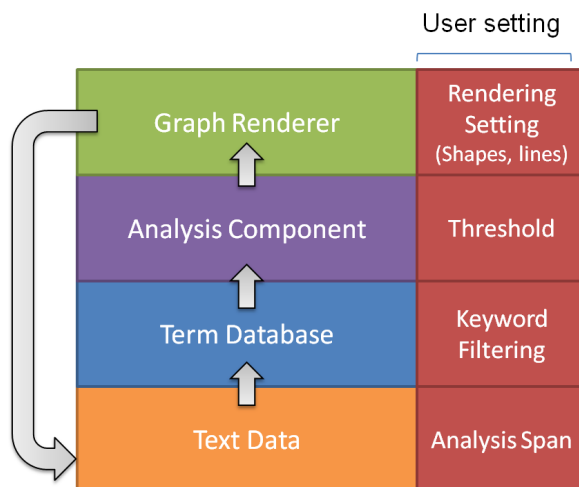


**Fig. 2.** Process of generating FACT-Graph

Additionally, a FACT-Graph visualizes keywords and relationships between keywords by using co-occurrence information. As a result, useful keywords can be obtained from their relationship with other keywords, even though that keyword does not seem to be important at a glance, and the user can extract such keywords by using FACT-Graph. Moreover, from the result of the class-transition analysis, the user can comprehend trends in keywords and in topics (consisting of several keywords) by FACT-Graph.

The steps for generating a FACT-Graph are as follows shown in Figure 2:

1. Separate time-series text data according to the analysis periods
2. Extract keywords in each period by morphological analysis and TF-IDF algorithm [7]
3. Carry out class transition analysis and extract co-occurrence relations.
4. Visualize keywords and relations.



**Fig. 3.** Layered architecture of FACT-Graph

A FACT-Graph is composed of three factors: time, keywords, and a co-occurrence network. The user of a FACT-Graph configures several parameters related to these factors such as analysis period, filtering keywords, and threshold of co-occurrence.

The software structure of FACT-Graph is shown schematically in Figure 3. The FACT-Graph consists of several components, namely, keywords extraction, class transition analysis, co-occurrence calculation, graph renderer, and a term database. Each component is highly independent (i.e., only loosely coupled to the others). For example, references [4] and [5] use Graphviz for rendering and MeCab<sup>1</sup> for keywords extraction [8]. The components do not share information about keywords and co-occurrence relations.

## 2.2 Problem of Analysis by FACT-Graph

Analysis by FACT-Graph necessitates generating FACT-Graph and setting of parameters. Accordingly, the following problems regarding the analysis must be solved.

### – Parameter setting

The user has to configure certain parameters in order to analyze trends by FACT-Graph. The user is forced into using trial-and-error method, so the cost of analysis increases. However, the components of FACT-Graph are so highly independent that there is troublesome for the user to configure each parameter of each component. This also raises the problem that it is difficult for the user to manage parameters consistently.

### – Information reference and interface for analysis

Keywords as well as relations change according to analysis period. By analyzing trends from keywords and relations shown in FACT-Graph, the user checks important keywords, and comprehends implied trends in information source such as the original text.

<sup>1</sup> MeCab, <http://mecab.sourceforge.net/>

Reference of the information source is useful for the analysis of trends. However, it takes a lot of effort to refer to the information about the keywords and relations in FACT-Graph. The keywords and relations in a FACT-Graph are shown as a static image. The effectiveness of a FACT-Graph is therefore governed by the loss of links to the information source, so the user cannot access the information directly from the FACT-Graph.

Another problem is that multiple FACT-Graphs do not share graph information. For example, information about positions of keywords is shown as nodes in a FACT-Graph. However, a FACT-Graph itself does not contain the information about positions of nodes. As a result, even if user analyzes same keywords in two different FACT-Graphs, their positions are changed. This results the problem of understanding trend in noteworthy keywords over periods.

In order to solve these problems simply, FACT-Graph has to keep and share information between two periods. However, a FACT-Graph consists of the loosely coupled software shown in Figure 3. It is good a flexibility handling but not so good at collaborative performance. As a result, the components do not share information about keywords and co-occurrence, and a FACT-Graph cannot be linked with that information (that is, nodes to keywords and links to co-occurrence). it is therefore difficult to refer to these clues seamlessly in order to analyze a FACT-Graph.

#### – Awareness in analysis

The user analyzes a FACT-Graph as he considers it and gets the idea. They may find out and come up with new awareness in the course of analysis. Also, the user may want to analyze keywords which appear to fadeout words from FACT-Graph.

It is important to implement a function to reflect the awareness based on subjectivity in knowledge discovery and knowledge acquisition [9]. In order to reflect the user's awareness, the user needs to configure parameters. However, it is troublesome for the user to achieve it. For example, keywords shown in FACT-Graph are extracted by TF-IDF algorithm. TF-IDF algorithm is based on TF and DF which are also used for the thresholds whether a FACT-Graph uses a term as a keyword. So it is difficult to appear only one remarkable keyword which user wants to survey because of difficulty of parameters setting.

These three above problems may become a barrier to analyzing a FACT-Graph. In light of the above-described problems, we have developed software based on the following requirements:

- Information must be shared among components and be managed consistently.
- A graphical user interface (GUI) for ease of reference to information (such as keywords and co-occurrence) must be supported.
- A user must be able to operate FACT-Graph itself.
- The software must reflect the user's awareness in FACT-Graph immediately.

In this paper, the operation of FACT-Graph is restricted to moving, fixing, and holding nodes (that is, keywords). Moreover, a keyword is regarded as a word that reflects awareness, and the software can add a certain keyword and its relations to the FACT-Graph immediately.

## 3 Loopo

### 3.1 Overview of Loopo

In order to satisfy the requirements outlined in Chapter 2, the authors have developed software called “Loopo”.

Loopo is software to improve analysis by FACT-Graph. Loopo generates FACT-Graph based on the parameters (such as keywords threshold, analysis period), which Loopo can configure easily.

Figure 4 is a screenshot that Loopo draws FACT-Graph from certain text data. Loopo consists of four windows: “FACT-Graph View,” which shows and operates the FACT-Graph itself; “Keyword Manager,” which manages keywords; “Time Manager,” which manages information and parameters concerning analysis periods; and “GraphInfo,” which shows and manages parameters concerning the network of the FACT-Graph. Several parameters such as keywords setting, analysis periods can be configured from these windows and the parameters are shared by the windows during multiple analysis periods.

The analysis by Loopo starts with the import of time-series text data. After importing text data separated according to analysis period, Loopo carries out morphological analysis, keyword filtering, and several initial setting steps along with the process of generating FACT-Graph. Also, Loopo can also export an image of a FACT-Graph drawn in FACT-Graph View.

**FACT-Graph View.** FACT-Graph View shows the analysis results for the text data which is imported to Loopo as FACT-Graph. The user can move, clear and fix keywords for trend analysis easily via the window. For example, the “fixing keyword” function is used to fix the locations of noteworthy keywords between multiple analysis periods. The user can therefore browse through remarkable keywords and their related keywords over the periods at ease. FACT-Graph View also allows the user to refer to original text data from remarkable keywords and helps them to comprehend macro/micro trends.

**Time Manager.** It is important to configure time period for time-series analysis. Usually, the number of articles is shown as a clue of setting of time periods. By displaying the trend in article volume as a chart, Time Manager helps the user configure the parameter concerning analysis period. The window indicates how the time periods are divided up for analyzing a FACT-Graph (which is output according to the time periods). Time Manager also has a function for setting time periods forward or backward. With this function, the user can view a series of FACT-Graphs via FACT-Graph View along with the change of time period.

**Keyword Manager.** Keyword Manager is the window for listing and managing the keywords currently shown in a FACT-Graph. The user can add and delete keywords, and refer to the original text data from a keyword via Keyword Manager or FACT-Graph View. As a result, the window can reflect the user’s awareness in FACT-Graph. The user can also configure the parameter, such as thresholds, concerning keyword extraction.

**GraphInfo.** One of the measures for identifying whether a FACT-Graph is meaningful is network information such as network size and density. GraphInfo shows the network information about FACT-Graph. GraphInfo shows network size, density, and the type of links as an overview of a FACT-Graph. It also shows several centralities such as betweenness centrality and closeness centrality when user selects a node of interest via FACT-Graph View [10]. Moreover, GraphInfo allows the user to change co-occurrence type and thresholds of co-occurrence.

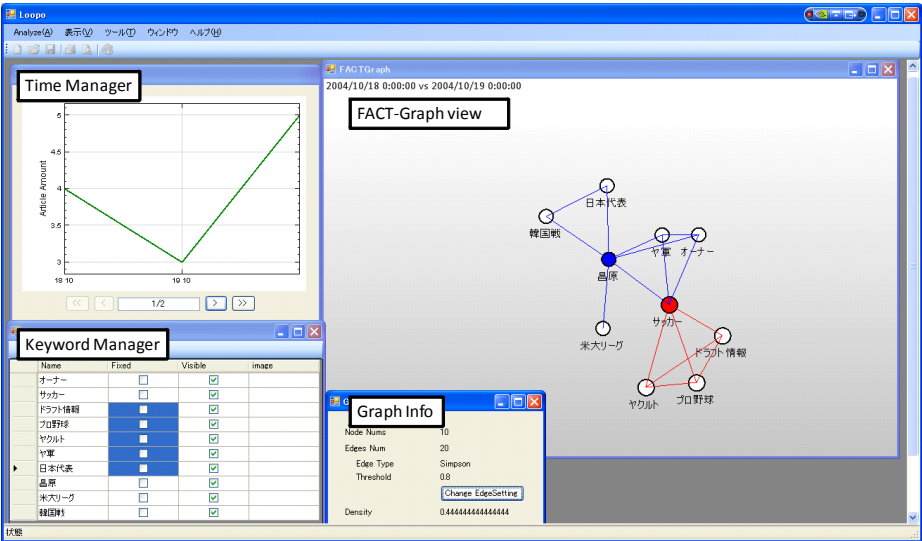


Fig. 4. Screenshot of Loopo

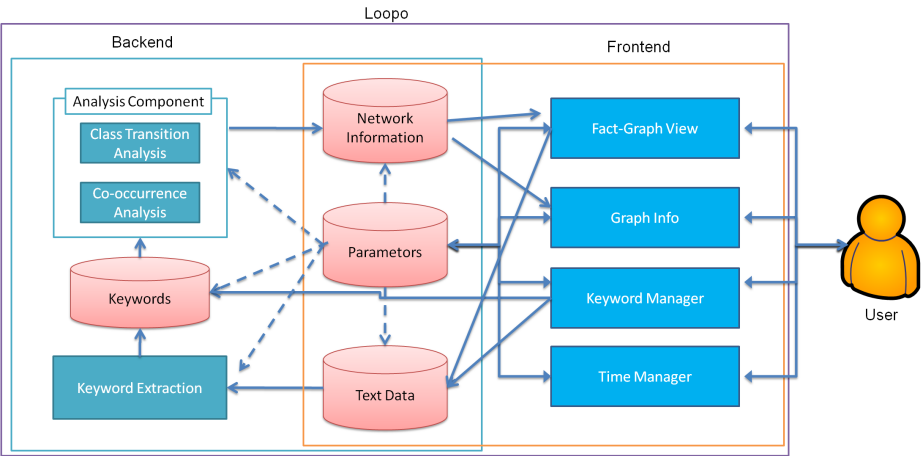


Fig. 5. Software structure of Loopo

### 3.2 Architecture

The software architecture of Loopo is shown schematically in Figure 5. Loopo has four databases concerning network information, parameters, original text data, keywords, and 3 component concerning keyword extraction, class transition, and co-occurrence analysis. It also provided four windows, namely, FACT-Graph View, Time Manager, Keyword Manager, and GraphInfo as user interfaces.

Each window (in front end) is operated by user and has a relation to all of the databases. The network-information database is related with and GraphInfo. The text-data and keyword databases are related to Keyword Manager and FACT-Graph View to allow the original text data to be referred to from remarkable keywords and enable operations such as addition and deletion of keywords. The parameter database is related to all windows because each window provides the functions of configuring parameters.

For keyword extraction (in back end), Loopo adopts the TF-IDF algorithm [11] of Harman. TF-IDF algorithm calculates the weight of terms based on TF and DF, and Top  $n$  weights are regarded as keywords in time period. The value of  $n$  is one of the parameters to generate a FACT-Graph and is configured by Loopo. For all databases, SQLite<sup>2</sup> is adopted in consideration of ease of installation. For rendering a FACT-Graph, Loopo uses the popular drawing algorithm “Spring Model” of Kameda et al. [12].

## 4 Discussion

Loopo was developed as a mining support tool for analyzing trends. There are many methods concerning visualization such as multi-dimensional scaling and self-organization maps as same as FACT-Graph. However, there are not many tools for mining support. Polaris is one analysis tools [13]. It was developed for easy analysis of “chance discovery” by KeyGraph [14]. The concept of Loopo is similar for that of Polaris. However, Loopo is used for trend analysis, so the goal of Loopo is different from Polaris.

Loopo was also developed on the assumption that an inexperienced user is analyzing a FACT-Graph. If the user has data for FACT-Graph, Loopo outputs FACT-Graph for the moment and can provide an opportunity for analyzing a FACT-Graph. It is also assumed that even a user unfamiliar with FACT-Graph in detail can carry out simple analysis of trends by FACT-Graph.

One of the other problems concerning FACT-Graph itself is that there is no systematic methodology for analyzing a FACT-Graph. Moreover, the results of trend analysis, which are derived from individual subjectivity, are often shared with other people. We consider that these problems can be solved by the functions of information-sharing tools (such as whiteboard and memo) and a wizard (which guides analysis process). These implementations are future works.

## 5 Conclusion

This paper has described a tool called Loopo for analyzing trends from a FACT-Graph. In analysis of FACT-Graph, we have discussed problems concerning

---

<sup>2</sup> <http://www.sqlite.org>



parameter setting, information reference and interface, and reflection of awareness. To resolve these problems, GUI-based software which manages essential parameters via for windows (FACT-Graph View, Time Manager, Keyword Manager, and GraphInfo) has been developed.

## References

1. Inmon, W.H.: Building the Data Warehouse. Wiley Publishing, Inc., Chichester (2005)
2. Nanba, H., Okuda, N., Okumura, M.: Extraction and Visualization of Trend Information from Newspaper Articles and Blogs. In: Proceedings of the 6th NTCIR Workshop, pp. 243–248 (2007)
3. Yamanishi, K., Li, H.: Mining Open Answers in Questionnaire Data. *IEEE Intelligent Systems* 17(5), 58–64 (2002)
4. Saga, R., Terachi, M., Sheng, Z., Tsuji, H.: FACT-Graph: Trend Visualization by Frequency and Co-occurrence. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) *KI 2008. LNCS (LNAI)*, vol. 5243, pp. 308–315. Springer, Heidelberg (2008)
5. Terachi, M., Saga, R., Sheng, Z., Tsuji, H.: Visualized Technique for Trend Analysis of News Articles. In: Nguyen, N.T., Borzowski, L., Grzech, A., Ali, M. (eds.) *IEA/AIE 2008. LNCS (LNAI)*, vol. 5027, pp. 659–668. Springer, Heidelberg (2008)
6. Terachi, M., Saga, R., and Tsuji, H.: Trends Recognition in Journal Papers by Text Mining. In: Proceedings of IEEE International Conference on Systems, Man & Cybernetics (IEEE/SMC 2006), pp. 4784–4789 (2006)
7. Salton, G.: Automatic Text Processing. Addison-Wesley Publishing Company, Reading (1989)
8. Ellson, J., Gansner, E.R., Koutsofios, E., North, S.C., Woodhull, G.: Graphviz - open source graph drawing tools. *Graph Drawing*, 483–484 (2001)
9. Tabata, K., Mitsumori, S.: An assertion-based information-probe system: Document-skeleton and glossary-skeleton approach. *Information, Knowledge, and Systems Management* 3(4), 123–152 (2003)
10. Freeman, L.C.: Centrality in social networks: Conceptual clarification. *Social Networks* 1(3), 215–239 (1979)
11. Harman, D.: Ranking algorithms. In: *Information Retrieval*, ch.14. Prentice Hall, Englewood Cliffs (1992)
12. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. *Information Processing Letters* 31(1), 7–15 (1989)
13. Okazaki, N., Ohsawa, Y.: Polaris: An Integrated Data Miner for Chance Discovery. In: Proceedings of Workshop of Chance Discovery and Its Management (in conjunction with International Human Computer Interaction Conference (HCI 2003), Crete, Greece (2003)
14. Ohsawa, Y., Benson, N.E., Yachida, M.: KeyGraph: Automatic Indexing by Segmenting and Unifing Co-occurrence Graphs. *IEICE D-I J82-D-I(2)*, 391–400 (1999)