

Constructing Semantic Networks of Development Activities from Weekly Reports

Motoyuki Takaai and Yohei Yamane

Research and Technology Group, Fuji Xerox Co., Ltd.
{motoyuki.takaai,yohei.yamane}@fujixerox.co.jp

Abstract. In the development departments of some manufacturing companies, there are weekly reports describing the status of events, but they are poorly structured plain texts. In this report, we propose a method for constructing semantic networks of development activities from weekly reports. Our ontology-based method extracts things such as the events, statuses and agents from the reports, constructs the relations between them, and creates Semantic MediaWiki pages from the semantic networks to visualize development activities. We show a use case in which the method is applied to the actual weekly reports and internal documents of a development department.

Keywords: Development activity, Information extraction, Semantic MediaWiki

1 Introduction

From discussions with the development departments in some manufacturing companies, the developers wanted to review their activities and internal documents from various perspectives, for example, components, status transitions, and participants. The authors of [3] conducted a trial to extract information from documents for decision making in a business process, but it does not cover the development domain.

In this research, we propose an ontology-based approach for extracting information about development activities from weekly reports. The weekly reports of departments contain rich information about their activities, but most of them have been poorly reused because each weekly report does not describe the contexts of activities and the status transitions of events. Our ontology and information extraction method constructs semantic networks of development activities from the weekly reports and identifies the relations to internal documents. Our system constructs a Semantic MediaWiki site from the networks; these form the user interface. The site allows users to browse the development activities from various perspectives and related documents as the contexts of activities.

2 Methodology

Fig. 1 describes the pipeline of our system. To design this pipeline, we borrow a part of the architecture of DIG [4]. The inputs to the pipeline are the weekly

reports and internal documents in the development departments. The output of the system is a Semantic MediaWiki site. The pipeline requires a dictionary and an ontology. We create the dictionary by merging some technical terminology dictionaries in the organization. The words in the dictionary are categorized within the classes of the ontology. The upper part of Fig. 2 shows a simplified version of our ontology for explanation. The oval nodes define classes, and the solid boxes define object properties. Each object property has domain/range restrictions; for example, the restriction on the “object” property means that each instance of the “Event” class might have one or more relations between the “object” properties and the instances of the “Machine” class.

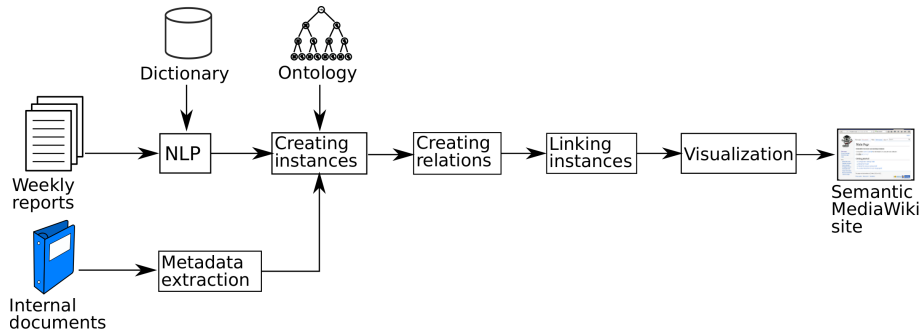


Fig. 1. System pipeline

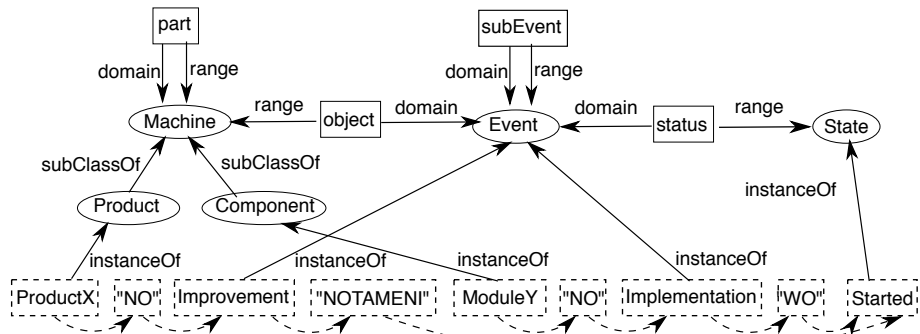


Fig. 2. An example of an ontology and the result of NLP

As shown in Fig. 1, the pipeline consists of several processes.

The process *NLP* constructs syntactic trees from texts. It consists of three subprocesses: sentence breaking, looking up words in the dictionary, and dependency parsing using CaboCha [2]. The lower part of Fig. 2 shows an example of the output of the NLP process. Each dashed box shows a Japanese word. For explanation, some words are translated into English. The original Japanese sentence is “ProductX NO KAIZEN(improvement) NOTAMENI ModuleY NO JISSOU(implementation) WO KAISHISHITA(started).” The meaning is “In or-

der to improve ProductX, we started the implementation of ModuleY.” The Japanese words “NO,” “NOTAMENI,” and “WO” are function words that indicate the syntactic roles of phrases; for example, the word “WO” indicates that the previous phrase is the object of the following verb. The dashed curved arrows show the dependency structure of the sentence.

The process *Creating instances* creates instances for words and annotates them with their classes. For example, the word “Improvement” in Fig. 2 indicates an instance of the “Event” class.

The process *Metadata extraction* extracts the title, filename, creation date, authors, and keywords as metadata from each internal document. The metadata are encoded to instances and relations directly.

The process *Creating relations* creates relations between instances. When a sentence has multiple instances and the types of the instances satisfy a domain/range restriction of a particular property, the system creates a provisional relation; this is similar to the approach in [5]. The system evaluates the certainty of the relations by using a one-class classifier of machine learning. The features for machine learning are created by encoding paths between words on the dependency structure. Currently, we deploy libsvm [1] for machine learning.

The process *Linking instances* completes semantic networks by creating sameAs relations over sentences. The conditions for creating the relation are as follows:

- 1) the labels of two instances are the same or similar, or
- 2) most of the properties between the two instances are the same or similar.

Condition 1) is used to identify persons, organizations, products, or components, and condition 2) is used for events. In our case, most of the names of events consist of common nouns; therefore, we could not use condition 1) for identifying events.

The process *Visualization* creates Semantic MediaWiki pages from the semantic networks. The system creates the pages for the instances in the semantic networks. The pages include the hyperlinks of the relations with some visualization methods, i.e., an infobox, a table, text, or a diagram.

3 A Case Study

We prepared 408 sentences of 30 weekly reports about a specific product in a production design department. The system constructed semantic networks that included 1351 instances and 667 relations.

Fig. 3 shows a screenshot of the Semantic MediaWiki site from the reports. The system provides some different viewpoints for the development events. The left window describes the state of an event. The infobox on the right side of the window shows the properties of the state and the event. The upper-right window shows a page for one product. The table in the window shows the events associated with the product and their latest statuses. The lower-right window shows the relations between two organizations as a semantic network. These Semantic MediaWiki pages also include hyperlinks to internal documents. Users can access detailed information about events by clicking the hyperlinks.

The screenshot displays a web interface for a Semantic MediaWiki site. On the left, there is a navigation sidebar with links like 'Main page', 'Recent changes', and 'Help'. The main content area shows a page titled 'クエリ拡張モジュールの実装を開始 150401'. Below the title, there is a summary box with details: '対象: IRespond, クエリ拡張モジュール', '関係者: FXDMS, INC', '関連技術: Java', '行為内容: 実装', '行為: 開始', '状態: 開始', '目的: 改善', '時期: 2015-04-01', 'プロジェクト: AutoAnswer'. To the right, a table lists related events:

時期	行為内容	行為	状態
15 April 2015	共通モジュール化	実装	完了
8 April 2015	IRespondの実装中 150408	実装	実装中
1 April 2015	クエリ拡張モジュールの実装を開始 150401	実装	開始
15 April 2015	プロジェクトの休止を含めた対応を検討中 150415	休止	検討 実装中

Below the table, there is a category '開発物' and a network diagram showing relationships between nodes like 'MO社', 'INC', and 'IRespondとINCの関係'.

Fig. 3. User interface of the system

4 Conclusion and Further Work

We have described a method for constructing semantic networks and Semantic MediaWiki sites of development activities using information extraction techniques. The system provides developers with access to events and internal documents with Semantic MediaWiki sites from some perspectives, for example, products, events, and relations between organizations.

In future work, the usefulness of this system in practical contexts in particular development departments will be assessed.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27:1–27:27 (2011)
2. Kudo, T.: CaboCha : Yet another japanese dependency structure analyzer. <http://chasen.naist.jp/chaki/t/2005-08-29/doc/CaboCha%20Yet%20Another%20Japanese%20Dependency%20Structure%20Analyzer.htm> (2004)
3. Saggion, H., Funk, A., Maynard, D., Bontcheva, K.: Ontology-based information extraction for business intelligence. In: *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*. pp. 843–856. *ISWC'07/ASWC'07*, Springer-Verlag (2007)
4. Szekely, P., Knoblock, C.A., Slepicka, J., Philpot, A., Singh, A., Yin, C., Kapoor, D., Natarajan, P., Marcu, D., Knight, K., Stallard, D., Karunamoorthy, S.S., Bojanapalli, R., Minton, S., Amanatullah, B., Hughes, T., Tamayo, M., Flynt, D., Artiss, R., Chang, S.F., Chen, T., Hiebel, G., Ferreira, L.: Building and using a knowledge graph to combat human trafficking. In: *The Semantic Web - ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*. pp. 205–221. Springer-Verlag (2015)
5. Taheriyani, M., Knoblock, C.A., Szekely, P., Ambite, J.L.: A graph-based approach to learn semantic descriptions of data sources. In: *The Semantic Web – ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*. pp. 607–623. Springer-Verlag (2013)