SEMI-AUTOMATIC DATA-DRIVEN ONTOLOGY CONSTRUCTION SYSTEM

Blaž Fortuna, Marko Grobelnik, Dunja Mladenić
Department of Knowledge Technologies, Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 477 3127; fax: +386 1 477 3315
e-mail: blaz.fortuna@ijs.si

ABSTRACT

In this paper we present a new version of OntoGen system for semi-automatic data-driven ontology construction. The system is based on a novel ontology learning framework which formalizes and extends the role of machine learning and text mining algorithms used in the previous version. List of new features includes extended number of supported ontology formats (RDFS and OWL), supervised methods for concept discovery (based on Active Learning), adding of new instances to ontology and improved user interface (based on comments from the users).

1 INTRODUCTION

In [Fortuna05a] we introduce a semi-automatic, data-driven system for constructing topic ontologies called **OntoGen**. The phrases "semi-automatic" and "data-driven" stand for:

- **Semi-Automatic** – The system is an interactive tool that aids the user during the ontology construction process. The system suggests concepts, relations and their names, automatically assigns instances to concepts and provides a good overview of the ontology to the user trough concept browsing and visualization. At the same time the user can fully adjust all the properties of the ontology by manually adding or deleting concepts, relations and reassigning instances.
- **Data-Driven** – Most of the aid provided by the system (concept, relation suggestion, etc.) is based on some underlying data provided by the user at the beginning of the ontology construction. The data reflects the domain for which the user is building an ontology. Instance and instance co-occurrences are extracted from the data together with their profiles. Representation of profiles will be discussed later.

The system is used in the EU project SEKT as well as in several other smaller projects. We got very informative feedback from the users which we took very seriously when developing the new version.

Besides improvements based on the users feedback, we also continued research in the direction of improving and generalizing the system. The new functionality included in the system is based on machine learning and text mining methods such as *simultaneous ontologies* [4], *active learning* [10], *automatic ontology population* [6], *text corpora visualization* [5] and *semi-automatic ontology construction* [3].

The rest of this report is organized as follows. In the next section we give a short overview of the previous version of the system and analysis of the users feedback. We also give a short description of methods from previous deliverables which are included in the new version of the system. Section 3 describes a theoretical framework on which the new version of the system is based while Section 4 demonstrates its implementation. We conclude this report with future work directions and final conclusions.

2 RELATED WORK

Here we give a short description of the previous version of the system together with a list of most notable user comments about it. Following that are descriptions of the machine learning methods which we integrated into the new version presented in this paper.

2.1 OntoGen v1.0

In [3] we introduced a system called OntoGen for semi-automatic construction of topic ontologies. Topic ontology consists of a set of topics (or concepts) and a set of relations between the topics which best describe the data. The OntoGen system helps the user by discovering possible concepts and basic relations between them within the data.

For the representation of documents we use the well established bag-of-words document representation, where each document is encoded as a vector of term frequencies and the similarity of a pair of documents is calculated by the number and the weights of the words that these two documents share.

The central parts of OntoGen are the methods for discovering concepts from a collection of documents. OntoGen uses Latent Semantic Indexing (LSI) [2] and k-means clustering [7]. LSI is a method for linear dimensionality reduction by learning an optimal sub-basis for approximating documents' bag-of-words vectors. The sub-basis vectors are treated as topics. k-means clustering is used to discover topics by clustering the documents' bag-of-words vectors into k clusters where each cluster is treated as a topic.

The user interaction with the system is via a graphical user interface (GUI). When the user selects a topic, the system automatically suggests its potential subtopics. This is done by LSI or k-means algorithms only on the documents from the selected topic. The number of suggested topics is supervised by the user. User then selects the subtopics s/he finds reasonable and the system adds them to the ontology as subtopics of the selected topic.
The system also has two methods for extracting the main keywords which help the user to understand and name the topics: keyword extraction using centroid vectors (descriptive keywords) and keyword extraction using Support Vector Machine (SVM) [8] (distinctive keywords).

2.2 Active Learning

Active learning is a generic term describing a special interactive kind of learning process. In contrast to the usual (passive) learning where the student is presented with a static set of examples that are then used to construct a model, the active learning paradigm means that the student can 'ask' the 'oracle' (eg., a domain expert, the user, …) for a label of an example (see Figure 1). Here we use the SVM based method originally proposed in [10].

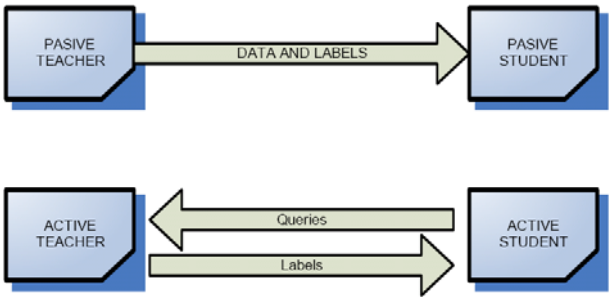

Figure 1: *Passive vs. Active Learning.*

2.3 Simultaneous Ontologies

The topic suggestion methods presented above heavily rely on the weights associated with the words – the higher the weight of a specific word the more probable that two documents are similar if they share this word. The weights of the words are commonly calculated by the so called TFIDF weighting [9].
In [4] we argue that this provides just one of the possible views on the data and propose an alternative word weighting that also takes into account the domain knowledge which provides the user's view on the documents. We integrated this method into data loading functions of the system.

2.4 Text Corpora Visualization

In [5] we presented a system for visualizing larger collection of documents. This system is now loosely integrated into OntoGen system to aid the user at comprehending and understanding the topics covered by the instances inside a specific concept. This is done by visualization of the instances using the Document Atlas tool [5].
Document Atlas is a tool for creating, showing and exploring visualizations of text corpora. The documents are presented as points on a map and the density is shown as a texture in the background. Most common keywords are shown for each area of the map. When the user moves the mouse around the map a set of the most common keywords is shown for the area around the mouse (the area is marked with a transparent circle). The user can also zoom-in to see specific areas in more details.

2.5 Ontology Population

In order to support addition of new instances to the ontology (ontology population) we use the approach proposed in [6], but instead of using k-nearest neighbors classifier in each of the concepts we use the concept's SVM linear model for classification of new instances into the existing ontology. The system shows to the user all the concepts that the instance belongs to together with the level of certainty for instance belonging to the concept (see Figure 5). Note that a new instance can be classified into more then one leaf concept.

3 USERS FEEDBACK

The topic ontology construction system was used in several projects, most notable being SEKT Case Studies *Decision Support for Legal Professionals* and *BT Digital Library*. We gathered the feedback from the users and used it as a guide when deciding what features to develop in the new version of the system.
Here we give a list of the main suggestions from the users, together with the related changes in the new version of OntoGen:

- Concept learning:
  - *"More details about the suggested concepts"* – the new version has extended keyword list describing suggested concepts
  - *"Generate suggestions only when explicitly asked"* – now the user must click a button to generate a suggestion list
  - *"I know what sub-concept to add but the system does not suggest it"* – now the user can generate concept suggestions by providing a query (for this task we used active learning)
- Concept management
  - *"How can I move a sub-concept?"* – this was already possible in the previous version by adding and removing relations; this is greatly simplified in the new version
  - *"System suggests a sub-concept which is not related to the selected concept"* – we added option to prune the suggested sub-concept which also removes related documents from the selected concept
- Ontology management

- *"Can I add new documents to the existing OntoGen ontology (e.g., to support online learning of digital library knowledge spaces)?"* – we added support for including new documents to the already built ontology

4 SYSTEM IMPLEMENTATION

4.1 Overview

The main window (Figure 2) is divided into three main areas. The largest part of the windows is dedicated to ontology visualization and document management part (the right side of the window). On the upper left side is the concept tree showing all the concepts from ontology and on the bottom left side is the area where the user can check details and manage properties of the selected concept and get suggestions for its sub-concepts.

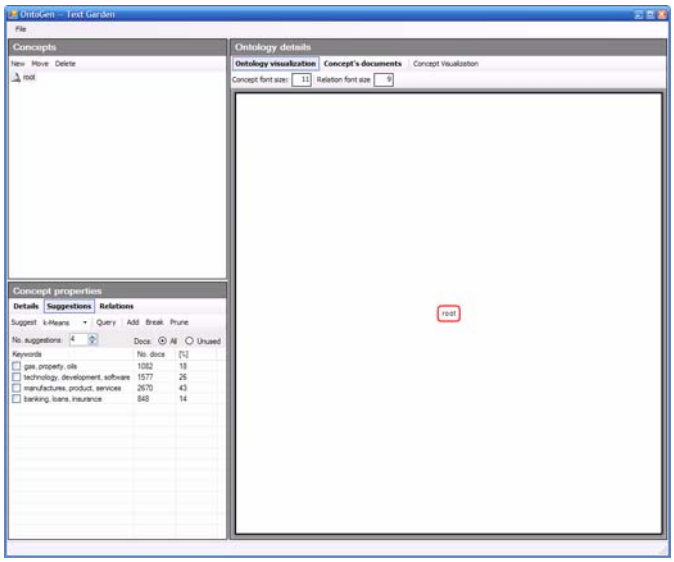

Figure 2: *The main window of the system.*

OntoGen supports several input formats for text instances and support for proprietary Text Garden format Bag-Of-Words. If the instances already have assigned some preliminary labels in the input data, then OntoGen automatically asks if it should apply SVM word weighting method [4]. Otherwise the TFIDF word weighting is used by default. Ontologies created in OntoGen can be saved as Proton Topic Ontology (also available in the previous version), RDF Schema or OWL ontology. OntoGen is also integrated into OntoStudio as a plug-in. The user can use it for creating initial version of ontology which he can then further refine inside OntoStudio.

4.2 Concept Suggestion

One of the main parts of the system is concept learning. There are two different approaches implemented for concept learning, supervised and unsupervised. In the unsupervised approach the system provides suggestions for possible sub-concepts of the selected concept and this was already implemented in the previous version of the system.

Sometimes the system identifies a sub-concept for which the user thinks that should not be part of the concept. The user can decide to prune the suggested sub-concept from the selected concept which effectively removes suggested sub-concept's instances from the selected concept. The prune feature is new.

A new feature in OntoGen is a supervised method for adding concepts. In the supervised approach the user has an initial idea of what a sub-concept should be about and enters it into the system as a query. Implementation is based on active learning method described in Section 2.2. The querying and active learning is only applied to the instances from the selected concept.

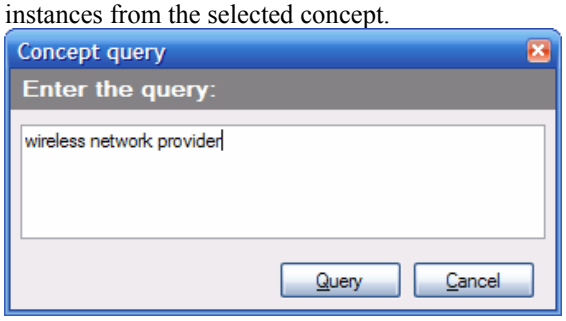

Figure 3: *The main window of the system.*

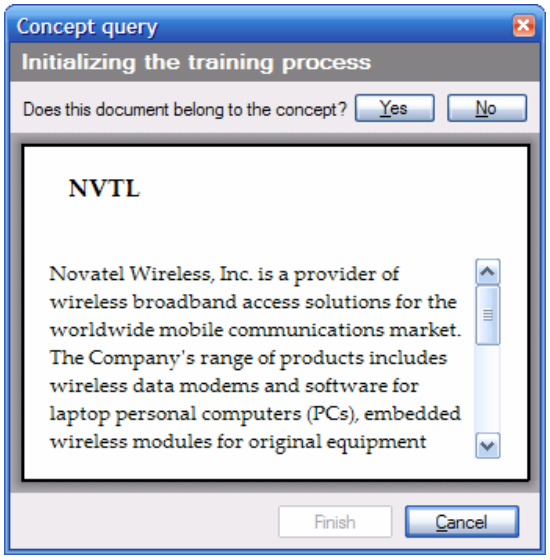

Figure 4: *The main window of the system.*

The user can start this method by clicking "Query" button. The system then launches a dialog that takes the query from the user (Figure 3). After the user enters a query the active learning system starts asking questions and labeling the instances (Figure 4). On each step the system asks if a particular instance belongs to the concept and the user can select Yes or No.

Questions are selected so that the most information about the desired concept is retrieved from the user. After some initial labeled sample is collected from the user the system displays some additional information about the concept. It displays the current size (number of documents positively classified into the concept) and most important keywords

for the concept (using SVM keyword extraction). The user can continue answering the questions or finish by clicking on the Finish button. The more questions that the user answers the more correct assignment of instances in the final concept are. After the concept is constructed it is added to the ontology as a sub-concept of the selected concept.

Unsupervised vs. Supervised: There is a fundamental difference between the unsupervised and supervised methods. The main advantage of unsupervised methods is that it requires very little input from the user. The unsupervised methods provide well balanced suggestions for sub-concepts based on the instances and are also good for exploring the data. The supervised method on the other hand requires more input. The user has to first figure out what should the sub-concept be, he has to describe the sub-concept trough a query and go trough the sequence of questions to clarify the query. This is intended for the cases where the user has a clear idea of the sub-concept he wants to add to the ontology but the unsupervised methods do not discover it.

4.2 New Instance Importing

The new version of OntoGen also enables the user to add new instances to an existing ontology. Ontology population described in Section 2.5 is used for this.

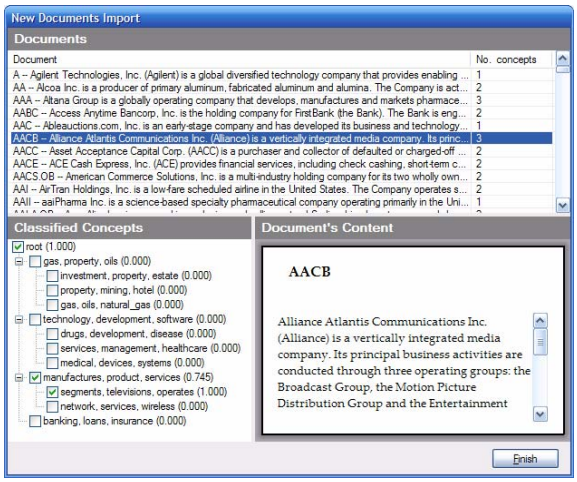

Figure 5: *Classification of new instances into the ontology.*

First the user loads new instances into the system. In the next step the system trains SVM classifiers on the instances already arranged into ontology and uses them to classify the new instances.

In the next step the OntoGen presents to the user a list of all the newly imported instances and their classification results (Figure 5). User can check and correct classifications for each of the instances by first selecting the instance from the list and then checking the appropriate concepts in the concept tree. Preview of the selected instance is also displayed to aid the user. The instances are automatically added to the ontology after the user clicks Finish.

4 CONCLUSIONS

In this paper we presented integration of various machine learning and text mining algorithms in a novel software tool for semi-automatic data-driven ontology construction. The system builds on top of our previous form [Fortuna05a] and includes new features based on users feedback and other research results from machine learning and text mining field.

As part of the future work we are planning to fully integrate relation learning into the system and to perform evaluation of the system based on ontology evaluation methods presented in [1].

OntoGen system is available as a free download from `http://ontogen.ijs.si/`.

Acknowledgement

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under SEKT (IST-1-506826-IP), and PASCAL (IST-2002-506778).

References

[1] Brank, J., Grobelnik, M., Mladenić, D. A Survey of Ontology Evaluation Techniques. Conference on Data Mining and Data Warehouses (SiKDD 2005), Ljubljana, Slovenia, 2005.

[2] Deerwester S., Dumais S., Furnas G., Landuer T. & Harshman R. Indexing by Latent Semantic Analysis. J. of the American Society of Information Science, vol. 41/6, 391-407, 1990.

[3] Fortuna, B., Grobelnik, M. Mladenić, D. *Semi-automatic construction of topic ontology*. Proceedings of the ECML/PKDD KDO'05 Workshop.

[4] Fortuna, B., Grobelnik, M., Mladenić, D. *Background Knowledge for Ontology Construction*. WWW 2006, May 23.26, 2006, Edinburgh, Scotland.

[5] Fortuna, B., Grobelnik, M. Mladenić, D. *Visualization of Text Document Corpus*. Informatica 29 (2005), 497-502.

[6] Grobelnik M., Mladenik D. *Simple classification into large topic ontology of Web documents*. In Proceedings: 27th International Conference on Information Tech-nology Interfaces, 20-24 June, Cavtat, Croatia, 2005.

[7] Jain, A. K., Murty M. N., & Flynn P. J. *Data Clustering: A Review*. ACM Computing Surveys, vol 31/3, 264-323, 1999.

[8] Joachims, T. *Making large-scale svm learning practical*. In B. Scholkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods, Support Vector Learning, MIT-Press, 1999.

[9] Salton, G. *Developments in Automatic Text Retrieval*. Science, Vol 253, 974-979, 1991.

[10] Tong, S., Koller, D. *Support Vector Machine Active Learning with Applications to Text Classification*. In Proceedings of 17th International Conference on Machine Learning (ICML), 2000.