

Experiments with Semi Automated Ontology Building using Text Onto Miner

Piotr Gawrysiak, Grzegorz Protaziuk, and Henryk Rybinski

Institute of Computer Science, Warsaw University of Technology

Abstract

This paper presents an overview of Text-Onto-Miner (TOM) – a natural language analysis system, developed by Institute of Computer Science of the Warsaw University of Technology during a research project supported by France Telecom. The main purpose of TOM is testing various algorithms and approaches that could be applied to the problem of building ontologies semi-automatically. It is also an efficient and extensible text mining system, that could be used in many applications related to natural language document processing.

Keywords: ontologies, natural language processing, text mining

1 Introduction

The importance of ontologies, and ontology supported knowledge retrieval systems, is constantly growing. It has been demonstrated that ontologies are a valuable tool in a variety of areas, including document retrieval, natural language processing or knowledge base integration and analysis. It is natural that their usage will be growing, as is growing the number of information repositories – especially repositories storing non textual data, such as photographic or music databases – see e.g. Ras (2007) - that need metadata enrichment and analysis.

While regarded as a very useful tool, ontologies usage is currently quite restricted. The main reason for this is a state of ontology engineering, which is still mostly manual process, very time-consuming, expensive and error prone. Therefore there is a growing need for automated – or at least semi-automated methods, that will be able to leverage the amount of information present in ever growing repositories of text data (e.g. obtainable via the Internet) in order to build useful ontology systems.

One can distinguish two main approaches in extracting semantic information, that could be used for creating an ontology, from corpora – knowledge-rich and knowledge-poor, according to the amount of knowledge they presuppose (Grefenstette, 1995). Knowledge-rich approaches require some sort of previously built semantic information, domain-dependent knowledge structures, semantic tagged training corpora, or semantic dictionaries, thesauri, ontologies, etc. (see e.g. Hamon (1998), Wu (2003)). In most cases, the reported methods refer to knowledge-rich methods, which require deep and specific knowledge "coded" into the algorithms, auxiliary dictionaries and/or thesauri, very much language and domain dependent. Although the methods may bring better results than the knowledge

poor based ones, the requirement for deep and specific knowledge is the main limitation in using these approaches. There is therefore a high demand for finding a knowledge-poor methodology that would give satisfactory results, especially for the cases of limited lexical resources. In this paper we present an overview of a novel approach (and set of tools) of a semi-automated method, that could help build ontologies thanks to the analysis and extraction of semantic information from large text corpora.

In the literature many approaches to building ontologies have been introduced and discussed (see for example Ahonen-Myka (1999), Beil (2002), Byrd (1999), Faure (1998), Fung (2003), Lame (2003), Morin (1999), Velardi (2003)). In Guarino (2002), and Noy (2001) the authors present an opinion that the process of ontology building is not a rigorous engineering discipline. Nevertheless, a tasks that are required in order to create an ontology are quite well define. According to e.g. Noy (2001) these include: (a) defining a domain and scope of an ontology, (b) creating a comprehensive list of concepts (classes) and their hierarchy, (c) defining relations between classes and (d) populating an ontology with instances of classes. Additionally, some auxiliary tasks, such as defining the properties of classes or preserving transitivity of some relations (e.g. taxonomy, `part_of`), while avoiding cycles in ontology, are usually required.

As shown in Rybinski (2007), there are already many publications referring to the research on automatic tools that support ontology building process in various phases. In many of above tasks (e.g. while determining relations between classes) some automatic tools can be used, to a higher or lesser extent. Some of these tasks cannot be even performed manually in a reasonable amount of time. This statement refers specifically to situations, where a huge amount of data should be processed and/or analyzed. With the text mining methods we can provide a support in such cases, however the discovered knowledge always requires human decision and intervention, thus provided tools will be always semi-automatic. The TOM platform system, presented in this paper, has been designed as such semi-automated ontology building system. It is mostly a research tool, used by us for evaluating various approaches to "supported" ontology creation. In its current state it can be however also used as a "production" tool, applied for example for maintenance of existing ontology systems.

This paper is structured as follows – chapter 2 contains an overview of the structure of TOM platform and lists the methods and algorithms that have been implemented. Chapter 3 presents experimental results of using these algorithms (esp. our modification of FIHC algorithm) for semi-automated building of term relation graphs, being the initial stage of ontology construction. Next chapters presents examples of results obtained by applying TOM for automatic synonyms and homonyms discovery. Finally, chapter 5 contains concluding remarks.

The research presented in this paper has been partially funded by France Telecom according to the research agreement n. 46 132 897.

2 TOM Platform

The Text-Onto-Miner platform (TOM- platform) is thought as a universal toolkit that should allow easy experimentation with various text mining algorithms that are applied for ontology building. As the main goal of TOM platform is to provide an environment for text mining experimenting and ontologies enrichment. This goal is achieved in TOM through three kinds of functionalities that the system provides, namely: ability to carry out text mining experiments, analysis of experiments results and direct manipulation (creation and editing) of ontology structures.

The system is highly modular (based on plugin architecture), and highly portable as Java has been used as the implementation language. At the top level consists of the following subsystems: *text mining subsystem, analysis support subsystem, ontology subsystem and data storing subsystem.*

The text mining subsystem enables a user to specify both data source of an experiments and a pipeline of a text mining process. Such a pipeline may consists of several steps for text processing, e.g. generation of the bag of words representation of documents, splitting the text into sentences, etc. In addition some text mining algorithms may be used as a step within a pipeline, for example clustering of documents, discovering frequent multiword terms, etc.

The analysis support subsystem is dedicated to working with the results obtained from text mining plan-and-experiment subsystem. For each type of results (clusters, sequences, candidate homonyms, etc.) the dedicated tools supporting basic analysis is available in TOM. Also the viewer of the *ftdoc* type documents is provided.

The ontology subsystem is thought as a tool with convenient graphical interface for working on ontologies, for example enriching ontologies based on results obtained from text-mining experiments. It enables a user to do various operations: such as browsing, annotating or validating ontologies. The subsystem also includes a tool by which a user may generate the owl file with proposals of new entries to ontologies. Such proposals are generated based on results obtained by using the text-mining algorithm implemented in TOM.

The data storing subsystem provides functionality concerning saving and searching for all text data used in the TOM platform. An integral part of this subsystem is an indexer (in TOM the Lucene index is used (Lucene, 2008)). The indexer helps one to create fast-searchable database of text documents backed up with inverted index and vector document representation.

Being a text mining tool, TOM provides a variety of methods of processing textual contents of documents. These include tokenizing text (which includes word and sentence boundary detection), grammatical analysis (such as POS-tagging, stemming, and grammatical relations detection) and document representation processing (including stop-words removal, word clustering and TF-IDF scaling).

The knowledge discovery algorithms that has been implemented as TOM modules include – clustering (FIHC, kMeans, DBScan, AHC, STC, O-Cluster), classification (kNN and Naive Bayes) and a variety of algorithms designed specifically for discovery of relations between words for the purpose of creating a basic ontology skeleton. These include:

- multiword terms (compound terms) and proper nouns detection – for this purpose the T-GSP algorithm has been designed and implemented. Additionally, a simple way of discovering proper names has been implemented and tested (during the last project phase);
- hierarchies (taxonomies) discovery. Proposed method is based on the modified FIHC algorithms;
- discovery of not-taxonomical relations between words. For this purpose two methods have been introduced: The first one requires shallow lexical analysis, and is based on the T GSP algorithm and grammatical patterns; the second one is based on grammatical relations. Its first step consists in performing a deeper lexical analysis;
- discovery of frequent termsets (sets consisting of terms). It is done by using the modified and improved version of the Apriori algorithm;
- homonyms identification. The worked out methods are based on the contexts of a word, which is created from frequent termsets;
- synonyms identification. The introduced method involves frequent termsets and the concise representations.

A more detailed description of these algorithms can be found eg. in Protaziuk (2007), Rybinski (2007) and Gawrysiak (2008).

With the methods mentioned above, the user obtains various results, which then have to be analyzed, and/or compared with existing resources. To this end, TOM offers several analyzing tools that support human decisions:

- Clustering analyzer – it can be used for analyzing clusters obtained from the clustering process. The tool is designed to support some high level operations concerning vocabulary, for example, it enables extracting a common raw dictionary for all the clusters, extracting a common raw dictionary from a selected cluster;
- Hierarchy relations converter – a tool supporting selection of relations obtained from the hierarchy discovering process;
- Sequences analyzer – a tool that supports analyzing the sequences generated by the T GSP component (it is a part of the T-GSP module). It can be used for supporting the analysis of the results obtained from the T-GSP algorithm, oriented on discovering compound terms;
- Relations, synonyms and homonyms analyzers – a tools that supports analyzing the association relations between words.

Except for the clustering analyzer, all the other tools allow the user to generate an xml file in the form of proposals for writing them into an ontology.

3 Ontology Hierarchy Building

The main objective of experiments that have been carried out with TOM, was to evaluate the approach for building an ontology from an unstructured text repositories. In the experiments we have not decided to build a complete domain specific ontology. Instead, we have tested the particular steps, all based on the same repository, in order to see how particular elements of the ontology can be obtained, and

how they can be incorporated into the ontology structure. In the research a FAO¹ document repository was used. The full repository consists of more than 20000 national legislation documents concerning legal issues for food and agriculture. The repository is provided in various languages. We have selected 5658 documents written in English documents. Of it, we were able to extract 546.617 paragraphs and then 1.296.929 sentences.

The first step of the experiment was to extract raw text from MSWord type files. To eliminate documents written in other languages than English, for each document we have calculated the volume of words included in English dictionary (in %). If the value was less than 90% a document was discarded from farther analysis, otherwise the document was subject to preprocessing. The documents were split into paragraphs, sentences and words, i.e. in a given document paragraphs were determined, then in each paragraph sentences were detected, and finally, words were extracted and annotated by appropriate part of speech tags (POS tag). In the next step the stop words were removed and the resulting text was analyzed in order to identify keywords ("top words") that could be used for ontology building.

For determining "top terms" specific for a domain of interest the k-means algorithm was used. We performed several experiments applying two approaches:

- Building hierarchical groups by repeating splitting groups into two clusters i.e. at the beginning two groups of documents were created from the entire repository, then each group was divided into two new clusters, and so on. The process was continued, until a satisfactory consistency of the clusters was achieved (measured as an average distance between documents and a cluster mean).
- Direct division into a given number of clusters. In the experiments we have evaluated clustering for 4, 8, 10, 16 and 50 groups.

Finally, for creating vocabulary we have chosen a direct division into 10 clusters. The list of "top words" was created as follows:

1. For each clusters a set of words which occur in at least 50% of documents belonging to the considered cluster was created.
2. The XOR operation was perform on the sets of words created in the previous step. It allowed us to obtain the list of words which were specific for clusters.

Additionally we have created a list of words common for the groups. To this end, we have applied the procedure, as described above, with the following differences:

- The value 20% was used instead of 50%.
- The AND operation was performed instead of the XOR operation.

We were interested only in nouns, so the lists were filtered out from words which are not nouns. Then, the lists were pruned by a researcher (e.g. removing words erroneously annotated as noun).

Finally, after merging the sets, we have obtained the list containing 127 words.

For obtaining the list of proper nouns and compound terms the T-GSP algorithm was used. We applied the following grammatical patterns:

¹Food and Agriculture Organization

- $gp1 = [\langle \{ \text{noun} \}, (\text{true}, 1, 1) \rangle, \langle \{ \text{noun} \}, (\text{true}, 1, 1) \rangle, \{0\}]$
- $gp2 = [\langle \{ \text{noun} \}, (\text{true}, 1, 1) \rangle, \langle \{ \text{noun} \}, (\text{true}, 1, 1) \rangle, \{0, 0\}]$
- $gp3 = [\langle \{ \text{noun} \}, (\text{true}, 1, 1) \rangle, \langle \{ \text{preposition} \}, (\text{true}, 1, 1) \rangle, \{0, 1\}]$

In the experiments we looked for sequences in entire documents. The minimal support was set to 40. It means that a given sequence was considered as interesting if it occurred in at least 40 documents. After merging the two result lists, we have obtained 458 proper nouns and 61 compound terms. This list was then used for building ontology hierarchy manually and with the modified FIHC algorithm. The results obtained from the FAO repository were not fully satisfactory, because FIHC produced wide but rather shallow tree (like 3 levels in depth). Among many relations only few were sensible. Moreover, there were many nodes that were not connected to each other. So, almost all useful relations were simple, separate, parent-child *isa* relations, which makes the results difficult to be used as a core for ontology. The other problem is that FIHC was not able to catch all important terminology from the domain repository. It is caused by the fact, that only frequent words are considered by the algorithm. Since the domain covered by the FAO repository is rather wide, the frequency of important words is low and similar to common words. This is also the reason for quite large presence of noise in the results. Finally, the depth of a tree is determined by the length of the frequent termsets found in the repository.

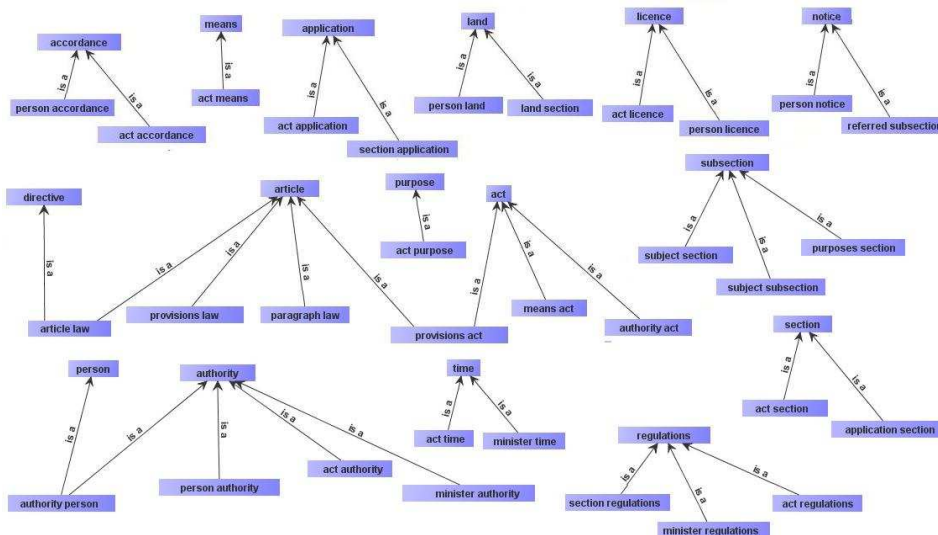


FIGURE 1: An ontology skeleton created automatically using FIHC

4 Discovering term meanings

The experiments for identifying synonyms and homonyms have been performed for the whole domain dictionary. All experiments were carried on about 200 000 sentences.

Synonyms

The algorithm for discovering synonymy and similarities has identified 131487 pairs composed of 1117 terms (nouns). This list of pairs has been filtered out by setting $ASIM \geq 0,18$ and $CSIM \geq 0,4$, which gave 2916 candidate pairs. Having sorted the pairs by the first term (alphabetically) and by ASIM, one can easily review the list and identify the acceptable pairs.

The most spectacular results were with finding some names of the "fish" species. The list is presented below in the table below:

fish	acanthias, alalunga, albacore, anchovies, bonito, brama, bream, brine, cancer, clupea, coalfish, cod, crabs, crangon, dentex, dogfish, engraulis, euthynnus, gadus, hake, halibut, herring, hippoglossoides, katsuwonus, lepidorhombus, lesser, ling, lobsters, lophius, mackerel, megrim, melanogrammus, merlangus, merluccius, molva, monkfish, morhua, nephrops, pagellus, pelamis, plaice, platessa, platichthys, pleuronectes, pollachius, redfish, reinhardtius, sardina, sardines, scomber, sebastes, shrimps, skipjack, solea, squalus, thunnus, tunas, virens,
------	---

For this group the precision reached some 80%. Another part of the resulting table has shown a pair of real synonyms:

Term1	Term2	ASIM	CSIM	Cosinus
thunnus	tunas	0,1818	1	1

Also the part shown below does show a very high precision result²:

dock	bull	yellow0,5	0,4	0,7746
dock	jetty	0,3333	1	1
dock	wharf	0,3333	1	1

Homonyms

For discovering homonyms we carried out series of experiments and applied a method based on closed sets (Rybinski 2008). The minimal support varied from 30 to 200. The words with the support higher than 20000 were not considered as interesting. In the experiments, for each candidate complete distinctive context were constructed. For the graph method the minimal support varied from 30 to 200. The words with the support higher than 20000 where not considered as interesting. In the experiments, for each candidate a complete graph was constructed, additionally all context had to be disjunctive, i.e. maximal number of links between the contexts has been set to 0.

²True eels (Anguilliformes) are an order of fish, which consists of 4 suborders, 19 families, 110 genera and 400 species. Most eels are predators (from Wikipedia).

After pruning the results by using the list of words created in Step we obtained 91 candidates for homonyms and then we selected 58 as homonyms. Below we present the exemplary results:

Term: agriculture; homonyms: {*conduct inquiries; development industries ministry; body director; control supervision; national service; animal bird; operations production*}

In this example some meanings should be discarded (especially the first meaning "conduct inquiries as it occurs in almost every other context graph). However, we still can find some interesting ones, e.g. "animal bird", "food product", "forestry lands", "environment pollution", "department planning" or "commission fisheries".

Term: port; homonyms: {*conduct inquiries; owner service; equipment master vessel; fisheries vessels; cargo ship*}

Here interesting meanings are: "owner service", "fisheries vessels", "cargo ship". The first one represents port as a vendor of particular services, whereas the two last ones are highly interesting as they represent two types of a port - small one for fishers and a cargo port.

5 Conclusion

The research briefly described in this paper is far from being complete. While the results of our experiments with applying knowledge-discovery algorithms to text corpora show, that it is indeed possible to extract automatically information about term relationships, the nature of extracted data is still not entirely satisfactory. The manual phase of cleaning the results, in order to obtain structures ready to incorporate into ontology, remains the crucial part of ontology building process. Therefore we are planning to continue research related to improving these methods.

Additionally, while the initial project, that has been commissioned by France Telecom, has been completed successfully (i.e. the TOM platform can be readily used in order to speed up ontology building and to improve the quality of resulting ontologies) we will be extending the TOM platform system, with special emphasis on ontology merging and introduction of more knowledge-rich methods into the platform.

References

- Helen AHONEN-MYKA (1999), Finding all frequent maximal sequences in text, in *Proceedings of the 16th International Conference on Machine Learning ICML 1999*, pp. 11–17.
- Florian BEIL and Martin ESTER and Xiaowei XU (2002), Frequent term-based text clustering, in *Proceedings of KDD 2002*, pp. 436–442.
- Roy BYRD and Yael RAVIN (1999), Identifying and extracting relations from text, in *Proceedings of NLDB 1999*.
- David FAURE and Claire NEDELLEC (1998), A corpus-based conceptual clustering method for verb frames and ontology acquisition, in *LREC Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications 1998*.

- Benjamin C. M. FUNG and Ke WANG and Martin ESTER (2003), Hierarchical document clustering using frequent itemsets, in *Proceedings of SDM 2003*.
- Piotr GAWRYSIAK and Henryk RYBINSKI and Grzegorz PROTAZIUK (2008), Text-Onto-Miner - a semi automated ontology building system, in *Proceedings of 17th International Symposium on Intelligent Systems 2008*, to appear.
- Gregory GREFENSTETTE (1995), Evaluation techniques for automatic semantic extraction: Comparing syntactic and window based approaches, *Corpus processing for Lexical Acquisition*, pp. 205–216.
- Nicola GUARINO and Christopher WELTY (2002), Evaluating ontological decisions with OntoClean, *Communications of ACM*, pp. 61–65.
- Thierry HAMON and Adeline NAZARENKO and Cécile GROS (1998), A step towards the detection of semantic variants of terms in technical documents, in *Proceedings of 36th Ann. meeting of ACL 1998*, 498–504.
- Guiraude LAME (2003), Using text analysis techniques to identify legal ontologie’s components, in *Proceedings of ICAIL 2003 Workshop on Legal Ontologies & Web Based Legal Information Management 2003*.
- Emmanuel MORIN (1999), Automatic acquisition of semantic relations between terms from technical corpora, in *Proceedings of the Fifth International Congress on Terminology and Knowledge Engineering 1999*.
- Natalya F. NOY and Deborah L. MCGUINNESS (2001), Ontology development 101: A guide to creating your first ontology, *Technical report, Stanford Knowledge Systems Laboratory*.
- Grzegorz PROTAZIUK, *et al.* (2007), Discovering compound and proper nouns, in *Proceedings of Rough Sets and Intelligent Systems Paradigms 2007*, pp 505–515.
- Zbigniew RAS and Xin ZHANG and Rory LEWIS (2007), MRAI - multi hierarchical, FS-tree based music information retrieval system, in *Proceedings of Rough Sets and Intelligent Systems Paradigms 2007*, pp 80–89.
- Henryk RYBINSKI, *et al.* (2007), Discovering synonyms based on frequent termsets, in *Proceedings of Rough Sets and Intelligent Systems Paradigms 2007*, pp 516–525.
- Henryk RYBINSKI (2007), State of the art on ontology and vocabulary building & maintenance research and applications, *Institute of Computer Science Warsaw University of Technology*.
- Henryk RYBINSKI, *et al.* (2008), Discovering word meanings based on frequent termsets, in *Proceedings of MCD 2007, LNAI 4944 2007*, pp 82–92.
- Paola VELARDI and Paolo FABRIANI (2001), Using text processing techniques to automatically enrich a domain ontology, in *Proceedings of FOIS 2001*.
- Hua WU and Ming ZHOU (2003), Optimizing synonym extraction using monolingual and bilingual resources, in *Proceedings of of the 2nd Int’l workshop on Paraphrasing 2003*, pp. 72–79.

