Text mining tool for ontology engineering based on use of product taxonomy and web directory

Jan Nemrava and Vojtěch Svátek

Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
{nemrava, svatek}@vse.cz

Abstract. This paper presents our attempt to build a text mining tool for collecting specific words – verbs in our case – that usually occur together with particular product category as support for ontology designers. As the ontologies are headstone for the success of the semantic web, our effort is focused on building small and specialized ontologies concerning one product category and describing its frequent relations in common text. We describe the way we use web directories to obtain suitable information about the products from UNSPSC taxonomy and we propose the method how the extracted information could be further processed.

1 Introduction

Information Extraction (IE) and Ontology (OL) learning are frequently discussed issues in the field of Semantic Web. The problems of information extraction using hand-crafted patterns have been addressed in many papers and it is obvious that the most promising way is automated or semi-automated ontology-based extraction of information. Since the results of IE from rigidly structured and semi-structured texts are already quite satisfying, the problems remain in field of unstructured free text processing. Large amount of knowledge-sparse text with full linguistic analysis would be too demanding. Shallow linguistic methods typically rely on POS tagging and/or shallow parsing. In our work we focus on finding verbs as simple POS category (in [4] called "indicator terms") that usually occur with some product selected from The United Nations Standard Products and Services Code[1] (UNSPSC) *product catalogue* so that we can:

- construct *ontologies* containing relations labeled with extracted verbs
- use these verbs for *extracting* further product categories from web pages

Web directory hierarchies (e.g. DMOZ[2]) are sometimes mistaken for ontologies; however, as already observed by Uschold [11], they are rarely valid taxonomies. It is easy to see that subheadings are often not specializations of headings; some of them are even not concepts (names of entities) but properties that implicitly restrict the

[1] http://www.unspsc.org

[2] http://www.dmoz.org

K. Richta, V. Snášel, J. Pokorný (Eds.): Dateso 2005, pp. 94–102, ISBN 80-01-03204-3.

extension of a preceding concept in the hierarchy. Consider for example .../Industries/Construction and Maintenance/Materials and Supplies/ /Masonry_and_Stone/Natural Stone/International Sources/Mexico.
Semantic interpretation of a sample of DMOZ paths revealed that:

– Terms in the headings belong to quite a small set of classes, such as 'Object' (i.e. product such as 'Car'), 'Subject' (e.g. 'Manufacturer' or 'Dealer'), 'Domain' (of competence of company, such as 'Transport' or 'Insurance'), 'Location' (e.g. 'Mexico') etc.

– Surface 'parent-child' arrangement of headings belonging to particular classes corresponds (with some ambiguity) to 'deep' ontological relations.

The idea of closed loop between IE and OL bootstrapped with web directory headings was first formulated in [4]: by matching headings (mostly corresponding to generic names of products, services, or domains of competence of companies) with full texts of pages, we can obtain content of these fulltext and use it for data extraction.

The reason why we use UNSPSC is that we would like to join this taxonomy and list of products with content of company websites to gain valuable information about verbs that usually occur in one sentence with some product category from the taxonomy. We build a tool that collects these verbs from given web pages. Presented text mining tool is based on combination of catalogue and fulltext search engine. Our approach is exploits redundancy of data on large data repositories like World Wide Web. We are exploiting the knowledge stored in hand classified web directories like DMOZ and we use their ability to provide web sites relevant to term we have chosen. The problem that had been already discussed in [10] is that the first website page does mostly does not contain much or even any text. When it does, it hardly ever describes the product or the offered services. This led us to use the fulltext search engines with restriction to particular website to ensure that we discover all term occurrences in content of whole company's website. As UNSPSC is freely available in standard ontology format from Protégé[3] website, it contains 16.000 unique products and has unambiguous structure, it is suitable for use in this field.

In this paper, we first describe the reason why UNSPSC was chosen, and why we use directories as source for our data. In next section we introduce our method to identify verbs related to products and in third section we describe experiments and the results. At the end of the paper related work and our future plans are discussed.

2 Proposed method description

2.1 Finding UNSPSC leaves in DMOZ directory

As suggested in [4] use of UNSPSC could be good technique how to overcome web directories problem with their structure and overlapping categories which describe more products. UNSPSC contain 16 000 specialized terms each describing particular

[3] http://protege.stanford.edu/

product category which can hardly be further divided. On the other hand this raise problem that UNSPSC tree leaves (product categories) varies from the directory headings in commonly used directory structure including DMOZ and Google directory. At current time there aren't any tools to automate the process of assigning right UNSPSC category to relevant DMOZ category so it must be done manually by choosing product from taxonomy and then finding appropriate category in directory. There are either a lot of categories describing our term or none. In the first case we focus on Business branch where we expect that the manufacturers and the company offering our products will be stated. The latter case – where no category is found – is worse and we have to find similar category, or find similar product. These two issues disallow this part of our work to be done automatically. In our test we found 7 nodes corresponding to the same number of products from UNSPSC from “Material handling” field.

2.2 Obtaining verbs from relevant web sites

We take advantage of human-classified web page links stored in web directories. As stated above their structure is not always valid taxonomy. Subheadings are often not specialization of headings; some of them are even not concepts (names of entities) but properties that implicitly restrict the extension of a preceding concept in the hierarchy. This is reason why we make use of UNSPSC classification in our paper. We would like to obtain so called „indicator verbs” that characterize particular term (product category in our case) in UNSPSC. Particular terms will be then generalized and may mine verbs that are indicative for the upper level of these terms. The trial was only made on one category and several terms, which limits the representativeness of results. Only several common verbs were obtained and they had to be classified manually, as we don't have any other categories to be compared with results from this. Next paragraph describes the text mining tool that collects data from selected directory category.

Table 1. Task sequence decomposition

1) Input: URL of DMOZ directory containing companies that manufacture desired product.
Output: List of URL of companies.
2) Input: URL of company website
Output: List of web pages containing the target term.
3) Input: Web page containing the term
Output: File with extracted sentences containing the term
4) Input: Sentence with term.
Output: Extracted verb.

Table 1 depicts sub-tasks of the tool. The input data for this tool are the *URL of directory in DMOZ* containing links relevant to chosen term and the *product category* chosen from UNSPSC. When we have chosen the right category the script can be run. The first part uses *link extractor* to obtain all company's web sites URLs. The list of extracted links is stored in file for further processing. Every URL from the list is then inserted into Yahoo Search Engine with the term we are currently exploiting and the parameter “site” is added. This ensures that the particular term is only searched on the selected web site. This process is repeated until all URLs from the list have been processed. We only store first 10 links from every domain, but it is only matter of setting of script and here we see a possibility of extracting more data. Up to 100 links from every company URL can be stored.

Now we have several hundreds links (depending on number of links on list) to sites where our desired term occurs in the page full text. Next part task is to extract every sentence from this set of links where the terms occur. The task is carried out by means of regular expressions and finding occurrence of the term in set of documents. As the sentences are discovered and saved into file we need to carry out some syntactical analysis to discriminate verbs from other lexical units. It is done by Adwait Ratnaparkhi's Java based Maximum Entropy POS Tagger (MXPOST) [6]. The extracted verbs are then compared with each other to find similar verbs, and number of occurrences is counted. Using WordNet[4] database and its ability to discover word stem from any word form we assure that neither text parser nor MxPost made mistake by during assigning verbs. If mistakes were made WordNet discovers them and it also provides lemma for each word inserted, which makes storing of verbs much easier. The Table 2 lists first 10 verbs given by our script for term ”hoists” where word in the 9th row (i.e. “products”) was incorrectly labeled by MxPost tagger as verb.

[4] http://wordnet.princeton.edu/

Table 2. Example of extracted verbs and their lemmas

word	lemma	next occurrences
include	*include*	include, includes, includes, includes, includes, includes, include, include, include, include, included, included, include, include, include, include, include, included, include
announced	*announce*	announced, announced, announced, announced, announce, announce, announced, announced, announced, announced
are	*be*	are, are
providing	*provide*	providing, providing, providing, providing, providing, providing, providing
feature	*feature*	featured, features, features, features, features, features, features, features, features, feature, feature
following	*follow*	following, following, following, following, following, following, following
leading	*lead*	leading, leading, leading, leading, leading, leading, leading, leading, leading
products	products is not a verb	products, products, products, products, products, products
including	*include*	including, including, including, including, including, including

We store all extracted verbs as matrix in relational database, where the discovered verbs represents rows and the desired terms represents columns. The intersection of row and column is the number of verb occurrences in all web pages of the directory category. We can use these sets of verbs and number of their occurrences for further examination on how the verbs characterize some broader term and if a small ontology could be build for every term. The goal of effort is to build separate lists of verbs such that some only characterize specific product types, while some other characterize whole product areas. We started with node *Handling materials and products* from this category.

3 Experiments

In extracting selected information we currently face the problem of how to set proper amount of web sites to be analyzed and what is the right amount of words taken in surroundings of the word. As stated above in this particular experiment, 10 web pages from one company website and one sentence from each web page containing examined word is taken. Our research shown that this can be sufficient amount of data for extraction of verbs for some common product types, but other product types from the same category of UNSPSC suffer from lack of data for extraction because they don't have appropriate category (node) in DMOZ.

There are two possible ways to overcome this problem. The first one is to concede the given constrains and allow our script to crawl more pages from one website and also allow to extract more sentences from one page. The second approach we have on mind is take advantage of some news resources like Google News service as there might appear verbs that characterize some product category. But from previous experience [3] we know, that the language used in non-official texts could contain misguiding verbs that have loose connection to term's denotation because of author's will to attract the readers attention by lot of ambiguous verbs. What is necessary is to restrict the domain of search, e.g. technical innovation, or technical news.

Using the above described tool we have built a database containing 303 verbs for 7 product categories from *handling material* category. These are only words that have appropriate category in DMOZ and therefore our approach could be used for their extraction. These verbs occurred 7300 times near the selected terms.

Our goal is to find some method that would enable us to categorize verbs as either:

- *common* for most products.
- *characterizing* one branch of products
- *specific* for small group of products, or even only *one product.*

Even from seven product categories – as expected – some verbs are obvious to be entirely neutral and do not characterize the products at all. According to three methods described later, verbs *be, have, provide* and *use* are common for all sentences describing any product. Then we have verbs describing activities connected with manufacturing of any types of products e.g. *design, require, offer, make, contact, manufacture, develop, supply,* etc. More specific for our branch might be verbs describing activities related to manipulating with material. They are *handle, lift, install* and *move*.

We experimented with three different measures that could separate specific verbs from more general ones. First and second are normalizations of frequencies to eliminate the influence of very frequent verbs. Normalization based on proportions of product categories in collection is the first, **Croft's normalization** using elimination of high-frequency terms with a specific constant is the second and **TF/IDF** [8] which relies on indirect relation between verb occurrences in its importance for product category is the last. We also tried **Lift measure [**2] but it didn't provide satisfactory results for aggregate values. We plan to use it for individual product category in future as it measures how many times more often occurs one verb with one term together than expected if they where statistically independent.

We tried these three methods to class verbs to their corresponding groups of verbs. All methods provided quite similar results. The first is *normalization* described by formula (1), where *Fij* is normalized frequency, f_{ij} is the frequency of verb *j* in product category *i*, V_{tj} is sum of all occurrences of product category *i* in collection and V is total number of collected verbs. Then V_{tj} / V represents how many per cent has product category *i* in collection. We recalculate whole matrix to get numbers ranging from 0 to 43 representing the normalized frequencies showing that the verbs with high value (30-43 in our case) are independent on the product category and thus they can be considered as common one. Verbs with values from 10 to 30 are not so often and they could be used as branch descriptors. The rest are with frequency lower than 10 are out of our interest for this moment.

$$F_{ij} = f_{ij} * (V_{tj} / V) . \qquad (\mathbf{1})$$

Croft’s normalization (2) moderates the effect of high-frequency verbs, where cf_{ij} is Croft's normalized frequency, f_{ij} is the frequency of verb *j* in product category *i*, m_i is the maximum frequency of any verb in product category *i*, K is a constant between 0 and 1 that is adjusted for the collection. K should be set to higher value (higher than 0.5) for collections with short documents. We used 0,3 as there are no different between 0.3 and 0.5 in our table. With this formula we get sum values for every verb ranging from 2.1 (7 product category × 0.3 for zero occurrences) for no occurrences of verb in our database to 8.58 for the most often verbs. Verbs with number above 5 normalized occurrences are significant for us as the common indicator while verbs between 3 and 5 normalized occurrences could be taken as the products representing verbs. The rest, with 3 and lower occurrences is for us as in previous method uninteresting.

$$cf = K + (1 - K) * f_{ij} / m_{ij} . \qquad (\mathbf{2})$$

TF/IDF (term frequency / inverse document frequency) (3), where w_{ij} is a weight of verb in product category *i*, f_{ij} is the frequency of verb *j* in product category *i*, *N* is number of all verbs in collection and *n* is sum of verb *j* occurring in all product categories. TF/IDF is technique that gives verb a high rank in a document if the verb appears frequently in a document or the verb does not appear frequently in other product categories. In other words a verb that occurs in a few product categories is likely to be a better discriminator than a verb that appears in most or all categories. As a result in this test we got values from 0 to 1350. Where as usual, the highest values between 1000 and 1350 are verbs that occur independently on selected product category and we consider them as common verbs. We are much more interested in verbs with value starting around 300 and ending at 1000. As stated above, these could be used as identifiers of the product category.

$$w_{ij} = f_{ij} * \log_2(N / n) \qquad (\mathbf{3})$$

In our trial we only examined 7 product categories from one UNSPSC node and hence we are not able to classify verbs into four categories as we suggested in part 1. We only classified them on common and specialized verbs. The first 15 results with values from each of described method are shown in Table 3.

The reason why this approach cannot be automatically run are mainly the non-corresponding items from product taxonomy to categories in widely-spread product catalogues. Our plans and intentions for the development of this tool are stated in future work section.

Table 3. Comparison of three methods

	lemma	Per cent	lemma	croft	lemma	TFIDF
1	have	43,01	have	8,58	have	1 318,40
2	provide	40,38	provide	7,41	provide	1 164,76
3	design	39,36	design	7,14	design	1 119,10
4	use	37,29	use	6,38	use	1 028,17
5	lift	26,47	require	5,32	require	802,81
6	require	26,43	handle	4,70	lift	703,11
7	handle	19,81	lift	4,70	handle	676,10
8	mount	17,75	offer	4,68	offer	648,62
9	operate	17,66	allow	4,31	allow	596,96
10	truck	17,61	include	4,30	contact	587,38
11	allow	17,25	please	4,29	move	582,57
12	contact	16,37	make	4,18	please	582,57
13	offer	15,99	contact	4,15	include	572,89
14	meet	15,91	need	4,06	meet	538,52
15	include	15,49	install	4,06	make	538,52

4 Related Work

The idea of combination information extraction with ontology learning has been described by Maedche in [5]. The idea of using identified words to extract more words was in [7] called *mutual bootstrapping*. This paper follows up with work [3] as it brought to this field use of universal product taxonomy and web directories and firstly suggested UNSPSC as possible way to obtain relevant data from given branch for a given product category. While Brin [1] uses fulltext search engines to obtain data from arbitrary sources we only use search engines for obtaining full text from the websites we have previously identified by another method, because our data are less structured and can be mistaken easily by ambiguous meanings of terms.

5 Future Work

As described in this paper, there are currently some limitations of this approach; they are mainly caused by lack of data to be mined from websites for some specialized terms. We proposed some techniques how to overcome these limitations. One of them is relaxing restrictions of fulltext search engines and the second is searching in all subdirectories for given terms in all whole DMOZ branch tree structure. All our plans for future stem from our effort to obtain as much data as possible and also better automation of whole process. We recently discovered a tool that could help us to use fulltext search on selected nodes from DMOZ web directory. As soon as we obtain verbs for more branches we could try to classify the verbs into four categories as proposed in section 3 and use them for creating ontologies with relations labeled with extracted verbs.

6 Acknowledgements

The authors would like to thank to Martin Labský and Martin Kavalec for their comments and help.

The research has been partially supported by the grant no. 201/03/1318 of the Grant Agency of the Czech Republic „Intelligent analysis of the WWW content and structure“.

7 References

1. Brin,S.: Extracting Patterns and Relations from the World Wide Web. In WebDB Workshop at EDBT’98
2. Brin S, Motwani R., Ullman J, Tsur S.: Dynamic itemset counting and implication rules for market basket data. In SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, pages 255-264, Tucson, Arizona, USA, May 1997.
3. Kavalec M., Maedche A., Svátek V.: Discovery of Lexical Entries for Non-Taxonomic Relations in Ontology Learning. In: SOFSEM – Theory and Practice of Computer Science, Springer LNCS 2932, 2004
4. Kavalec M., Svátek V.: Information Extraction and Ontology Learning Guided by Web Directory. In: ECAI Workshop on NLP and ML for Ontology engineering (OLT-02). Lyon, 2002.
5. Maedche A., Neumann G., Staab S.: Bootstrapping an Ontology Based Information Extraction System. Studies in Fuzziness and Soft Computing, editor Kacprzyk J., Intelligent exploration of the web, Springer 2002/01/01
6. Ratnaparkhi A.: Adwait Ratnaparkhi's Research Interests, [online], http://www.cis.upenn.edu/~adwait/statnlp.html
7. Riloff E., Jones R..: Learning Dictionaries for Information Extraction by Mulit-Level Bootstrapping, Proceedings of the Sixteenth National Conference on Artificial Intelligence, P 474-479, 1999
8. Salton G., Buckley C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24 (5):513--523, 1988.
9. Salton, G. and Buckley, C. (1988d). : Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24:513-523.
10. Svátek V., Berka P., Kavalec M., Kosek J., Vávra V.: Discovering company descriptions on the web by multiway analysis. In: New Trends in Intelligent Information Processing and Web Mining (IIPWM'03), Zakopane 2003. Springer-Verlag, 'Advances in Soft Computing' series, 2003
11. Uschold M. , Jasper R.: *A Framework for Understanding and Classifying Ontology Applications*. In: Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends.