DOMAIN ONTOLOGY: AUTOMATICALLY EXTRACTING AND STRUCTURING COMMUNITY LANGUAGE FROM TEXTS

Kurt Englmeier♬, Fionn Murtagh[μ], Josiane Mothe[✗]
♬ *LemonLabs GmbH, Germany KurtEnglmeier@computer.org*
[✗] *Institut de Recherche en Informatique de Toulouse, 118 Route de Narbonne 31400 Toulouse France mothe@irit.fr*
[μ] *Department of Computer Science, Royal Holloway, University of London*

ABSTRACT

We present a method that automatically builds a domain concept hierarchy from a textual corpus. This concept hierarchy aims at helping ontology building and for semantic indexing of document collections. The approach we implemented aims at being automatic, as independent as possible of the language and can be applied to any field. We evaluate the different steps of the method using a Wikipedia sub-collection. We clearly show that our method can be of great help when building domain ontologies. The results also show the benefits of using C-value to select candidate terms.

KEYWORDS

Building ontology, text analysis, C-Value, term selection, concept hierarchy

1. INTRODUCTION

The aim of the semantic web is to make web information or resources comprehensible and useful not only for human but also for machines, advanced services and search engines. Searching for information on the semantic web refers to various points of views (Corby et al. 2004): that of the design of ontologies which aims at representing a field of knowledge, that of the semantic indexing of the resources starting from these ontologies, finally, that of the access to resources for a user or an engine. Each of these aspects has its specific problems to solve but it is necessary to consider these three aspects to lead to operational systems on the semantic Web. We focus here on the building of ontologies even though we have developed an engine that automatically indexes texts from a lightweight ontology (Hernandez and Mothe, 2004). This latter indexing engine is based on semantic indexing that associates concepts to texts and weight these associations according to the occurrences of concept labels in texts and according to subsuming and non-taxonomical relationships that exist in the ontology.

Knowledge acquisition is a key issue when building ontologies that represent a domain. Different methodologies have been proposed to help manual building; however, manually building an ontology is time and effort consuming and cannot be considered to build all the domains of the world. Exploring texts with some statistical or semantics based tool is an efficient way to design terminological structures, and particularly ontologies (Maeche, 2003). This trend has recently benefited from the synergy between research in various disciplines such as text mining, knowledge acquisition from texts, natural language processing (NLP), corpus linguistics or even terminology (Aussenac and Mothe, 2004). TERMINAE (Aussenac et al. 2000) and TEXT-TO-ONTO (Maedche and Staab, 2001) are examples of such tools that rely on linguistics techniques, but human intervention at different phases of the ontology development is still necessary in these approaches. Text mining approaches combined with information extraction techniques can help the ontology designer by automatically determining what the important terms are and what the relationships between terms are. When based on NLP, these techniques are language dependent. Even if tools have been developed for different common languages, developing a tool to analyse a new language is time consuming and much effort is required. For this reason, we promote an approach which is as independent of the language as possible. The aim is to automatically extract terms from a set of representative documents and automatically structure this vocabulary. Regarding terms, (Seretan et al., 2004) distinguishes compound words that are units of lexical category from collocations that are conventional associations of words.

Our approach is based on n-gram detection to extract term candidates; two types of weight are combined in order to select good candidates. Then, selected terms are organised in order to derive a concept hierarchy.

This paper is organized as follows: in section 2, we first explain how term candidates are extracted and selected. Our approach is based on n-gram detection. In an information retrieval system, ideal candidates would serve to discriminate between relevant and non relevant documents when the collection is queried, but good candidates for a domain ontology should also be terms that are frequent because these frequent terms are generally representative of a domain. In our approach, we consider these two aspects to select good candidates. In section 3 we explain how the selected terms are organised in order to derive a concept hierarchy. The mechanism is two-fold: firstly the longest terms (longest n-grams in terms of words) are considered as leaves of the hierarchy and their hyperonyms are either automatically built from the n-gram components or selected from the other extracted n-grams. Then the upper level of the hierarchy is automatically extracted from a general resource (WordNet). Related works are described in each of these two sections. Section 4 is devoted to examples of results and evaluation based on a Wikipedia sub-collection.

2. TERM EXTRACTION AND SELECTION

2.1 Related works

Different automatic extraction techniques have been proposed in the literature and implemented in several contexts. Some are based on the statistical analysis of terms in documents (Church and Hanks, 1990) (Feng and Croft, 2001) while others are based on syntactical features. Many combine the two types of approaches. (Frantzi et al. 2000) exploits both linguistic and statistical features to extract phrases. The former features are used to filter word sequences (noun and adjective phrases). The latter exploits the occurrences and emphasis terms that may appear as nested within longer terms. The authors show that combining these two approaches is efficient on different text collections. (Song et al., 2003) uses the information gain measure to select the best candidate terms that are first extracted using the tf.idf and distance features. (Wu and Agogino 2004) implements a genetic algorithm to find the optimal set of key-phrases that display the lowest dispersion level. Candidates correspond to noun phrases extracted from sentences. In GenEx (Turney, 2000), key-phrases are extracted using a set of parameterized rules. The optimal set of parameters is learnt from a training set using a genetic algorithm that optimizes the number of automatically extracted phrases as compared to the known phrases). The FipsCo system (Goldman et al., 2001) extracts bi-grams that respect predefined syntactic rules (e.g. noun-adjective, adjective-noun) from a parsed text using a POS tagging. (Seretan et al., 2004) relies on a similar approach to extract multi-word collocations. (Feng and Croft, 2001) uses a Markov model approach to extract noun phrases. Candidates are matched to a dictionary to train the Markov Model. (Berman, 2005) developed a method to extract candidate terms based on the hypothesis that knowledge domain is composed entirely of word combinations found in other terms from the same knowledge domain.

2.2 Principle

Candidate terms are words or list of n words that are automatically extracted from a corpus of resources (documents, service descriptions, etc.). They are called candidates because not all of these elements will be part of the concept hierarchy or ontology: only selected terms will, depending on their calculated importance.

The terms are extracted in two complementary ways. The first one consists in their extraction according to a dictionary the user provides, while the second one resides on a statistical analysis of the collection. The extraction by dictionary compares each n-grams extracted with the dictionary entries and select only the candidates that match an entry. Considering only this method would make the hypothesis that a complete dictionary exists and that only necessary phase is the selection phase. For this reason, the process is completed by the statistical analysis method. It extracts collocations that correspond to candidates. Candidates are n-gram automatically extracted from texts. Ideal candidates serve to discriminate between relevant and non relevant documents when the collection is queried, but good candidates are also terms that are frequent because these frequent terms are generally representative of a domain. Two complementary weights are thus used to select the extracted candidates.

Figure 1 describes the algorithm we use to extract and select terms that will occur in the knowledge domain resource built. In our approach, the value of n, that is to say the length of the extracted n-grams is a parameter. Section 4 presents a comparative study of the influence of this parameter. The code associated to our method is composed of about 130 lines of Python code (without the document parsing and term lemmatization part). It takes about 30 min to extract 4-grams in a collection that contains 75000 single terms when using a simple PC.

Figure 1: Candidate extraction and selection - Algorithm

```
Stem all the single terms apart from stop-words
/* extract n-grams n from 2 to n */
for i in [2, n]:
|  repeat from the beginning of the collection (first term from the first document)
|  |    consider a i word-window (i.e. i-gram)
|  |    if i-gram is valid (more than 2 non-stop-words, no stop word at the beginning or
|  |    end)
|  |      Select the i-gram as a candidate
|  |      Calculate the i-gram total frequency
|  |    move the window by 1 term
|  until end of the collection
endfor
for each candidates
|  if weight < weight_threshold: remove
|  compute the C-value
   if C-value > probability threshold: add n-gram to dictionary
endfor
add all 1-gram to dictionary
```

2.3 The role of stop words

Stop words are generally eliminated when texts are analysed in order to build text description. However, stop words play important linguistic roles and are part of many phrases (examples: “part of speech” “collection of texts”). For these reasons, when extracting phrases, they are not necessarily eliminated. As a result, extracted n-grams will contain stop words. However, because candidates are combinations of words that occur in texts, it can happen that a stop word occurs at the end or at the beginning of a sequence; in that case it is deleted to obtain a real candidate.

2.4 Stemming

Stemming is used in order to conflate terms in a common form. For example “representations” and “representation” will be stemmed into “represent”. In our approach, we stem each single word before extracting n-grams from texts. This leads to n-grams and they equivalent terms. The stemmed n-gram will be considered as the label of the concept in the ontology while variants of this n-gram correspond to the associated labels. For example, to the concept *knowledg represent* will be associated two additional labels: *knowledge representation*, *knowledge representations*.

2.5 Selecting candidates

All candidates will not be part of the built ontology. We consider two types of weights that will be used to select the candidates.

Firstly, candidates should get a weight higher than a threshold. This weight is computed using the following function which is based on so called “tf.idf” weighting scheme in information retrieval:

$$Weight(t) = \max(tf_{doc} \log \frac{N}{n_t} + 1)$$

Where tf_{doc} is the term frequency in the document doc, N is the total number of documents and n_t is the number of documents containing the term.

Second, they should have a strong C-Value. C-Value enhances the usual statistical measure of frequency of term occurrence, making it sensitive to the nested terms (Frantzi et al., 2000). C-Value is defined as follows:

$$C-value(c) = \log_2|c| \cdot (f(c)) - \frac{1}{nbc}\sum_{Tc\in} f(Tc)$$

Where c is a candidate, f(c) its frequency, Tc is the set of candidates that contain c, nbc is the number of candidates.

Section 4 provides details on the influence of these parameters on a document collection in English.

3. STRUCTURING THE VOCABULARY

3.1. Related work

Relation extractors usually are based on linguistic patterns such as Prométhée (Morin, 1999) or Caméléon developed in our team (Séguéla, 1999). Patterns are applied on tagged text files to find out phrases where the lexical relation appears. Most of these systems are based on M. Hearst principles: patterns and relations may be either general or domain dependant (Hearst, 1992). The use of these tools may require some linguistic skills but it gives significant information for structuring domain knowledge. (Chen et al., 2005) presents a way to automatically extract term relationship to build a thesaurus. Their method relies on conditional probabilities and takes into account both the occurrence context and the content context of terms.

More complex platforms include several of these tools and support a modelling process for ontology engineering. KAON is a standard workbench for the analysis of texts in German and English (Maeche and Staab, 2003). Terminae is a general workbench to build up ontologies from large text corpora in French or English (Aussenac et al., 2000). The Terminae methodology suggest steps and heuristic principles to apply linguistic tools (Syntex and Caméléon), to explore their results, to identify and structure knowledge in an ontology. Validation is made through a formal representation in a logic description.

3.2. Building a concept hierarchy

Figure 2 provides the algorithm we use. The associated code is composed of about 200 lines of Python code. It takes about 25 sec. (using a Athlon XP 2500+ with 512Mo RAM) to structure a dictionary composed of 15863 n-grams. In the algorithm, thresholds are used in order to decide what terms are useful. Section 4 gives elements on the evaluation.

Figure2: Structuring the vocabulary to a concept hierarchy - Algorithm

```
Select discriminant terms (T):
for each term in the candidate list:
   compute the relevance weight
   if weight > weight threshold: count the number of documents in which it occurs.
For each so called discriminant term:
   search recursively for a sub-expression in the candidate list:
        get the sub-expression having the greater probability (C-value)
   if a sub-expression is in the candidate list use the sub-expression as parent
   else use the first n-1 words of the expression as parent
```

where $rel\_weight(c) = f(c) \cdot (\log(D/Dc)+1)$, *and D is the number of documents ; Dc, those that contains c*

Using this method to structure a vocabulary leads to a large number of concepts at the higher level. This is damageable for indexing because general vocabulary (not specific of a domain) may be lost. To solve this problem, we use a general resource (in our case WordNet) in order to had more generic terms to the ontology. The all process is described in (Hernandez, 2006).

Basically, the principal is to select the terms that are at the higher level of our ontology and retrieve them in WordNet by a simple term matching. Then, matching Synsets are disambiguated and we add the associated hyperonym from WordNet into our ontology.

4. A SUB-COLLECTION OF WIKIPEDIA TO EVALUATE THE PROCESS

The results presented in this section have been obtained using a collection from the English Wikipedia section about artificial intelligence. It is composed of about 30 documents corresponding to a collection of about 75000 words long.

4.1. Example of results

Figure 3 presents two extracts of the resulting concept hierarchy. The concepts are represented in bold. They are followed by their different labels as they occur in texts or may occur in future texts related to the field.

Figure3: Resulting concept hierarchy – Extracts

```
fuzzi logic: fuzzy logic
      fuzzi logic rule: fuzzy logic rules, fuzzy logic rules
            write fuzzi logic rule: writing fuzzy logic rules, writing fuzzy
            logic rule, write fuzzy logic rules, write fuzzy logic rule
fuzzi set: fuzzy sets, fuzzy set
      yuan fuzzi set: yuan fuzzy sets, yuan fuzzy set
fuzzi truth: fuzzy truth
      fuzzi truth repres: fuzzy truth represents, fuzzy truth represent
fuzzi truth repres membership: fuzzy truth represents membership
fuzzi variabl: fuzzy variables, fuzzy variable
[….]
knowledg repres: knowledge represented
knowledg represent: knowledge representation, knowledge representations
      web knowledg represent: web knowledge representation, web knowledge
representations
      web knowledg represent view: web knowledge representation views, web
knowledge representation views
knowledg sourc: knowledge source, knowledge source
      knowledg sourc integr: knowledge source integration
knowledg structur: knowledge structures, knowledge structure
```

4.2. Weight and C-value of the candidates

Candidates are selected according to different weights.

4.2.1 Weight of the candidates

Figure 4: Sample of the extracted terms, their weight and relevance (3-grams)

Weight	n-grams	Statut
1	Network	Relevant
0,982598475	expert system	Relevant
0,858552512	neural network	Relevant
0,787006469	genet algorithm	Relevant
…	….	….
0,252510893	Emptyrow	Non Relevant B
0,252510893	Cyc	Non Relevant B
0,246666808	semant network	Relevant

0,242346306	User	Non Relevant B

Non Relevant A is used when the candidate is not relevant for the ontology but is part of the interface or the language of the system, here Wikipedia (examples: copyright permission, upload file). *Non Relevant B* is used for non relevant terms, that is to say collocations with no semantics (example: Emplyrow).

Considering the top 50th terms, 31 are relevant to be inserted in the ontology, 5 correspond to truncated phrases, 3 are phrases from the system and 9 are non relevant. Figure 4 corresponds to an extract of the results.

4.2.2 C-Values of the candidates

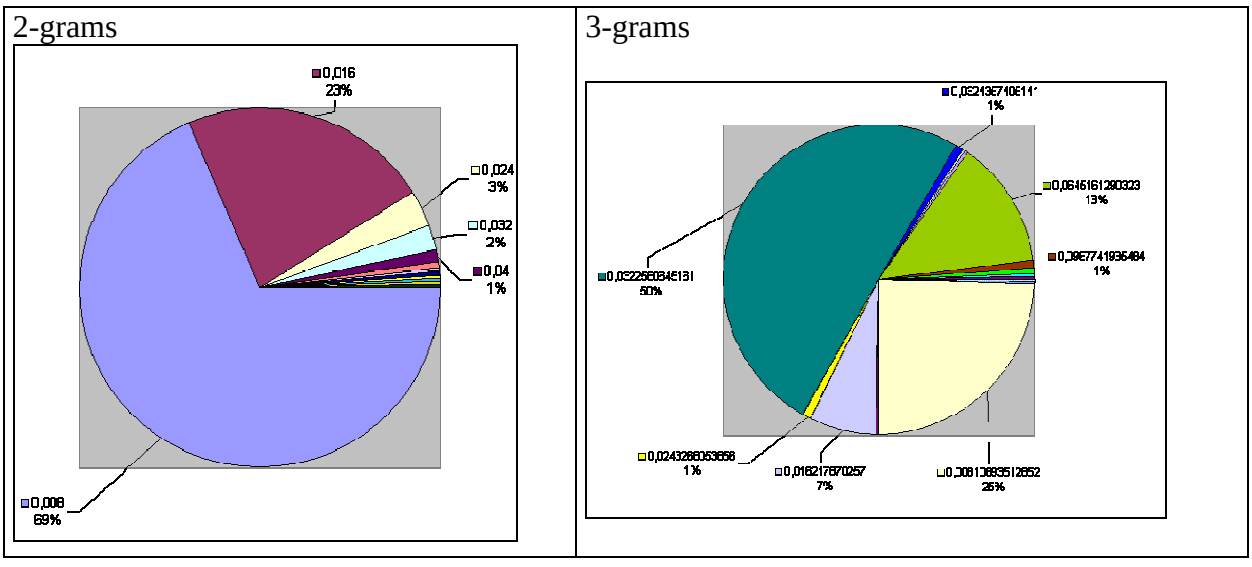
In this study, we consider all the candidates extracted, whatever their frequency and c-values. We considered the extracted terms and decide manually of their relevance. Figure 5 is a sample of the results when considering 3-grams. Considering the 50 top terms, 22 are relevant to be inserted in the ontology, 19 are part of the system navigation, 3 are truncated terms and 6 are non relevant. Considering a larger sample of randomly selected 290 terms, 138 are correct, 90 are just collocation but without semantic, 32 are truncated phrases and 30 are correct but correspond to phrases used in the interface.

Figure 5: Sample of the extracted terms, c-values and their relevance (3-grams)

C_Value	n-grams	Statut
1	Copyright for detail	Non Relevant A
1	edit this page	Non Relevant A
1	neural network	Relevant
1	free document licens	Non Relevant A
0,98118	genet algorithm	Relevant
0,89589	artifici intellig	Relevant
….	…..	…..
0,03225	term the train	Non Relevant B
0,03225	term was coin	Non Relevant B

The study clearly shows that n-gram candidates that get the higher C-Values are either highly relevant terms of the domain (e.g. genet algorithm) or terms used in the interface of the web site. (e.g. copyright for detail). Term C-Value is a parameter of our approach. To be part of the ontology, a candidate should reach a certain C-Value. Figure 6 indicates the repartition of the C-Value according to n.

Figure 6: Distribution of C-Values according to n ; Wikipedia English



Most of the 3-grams (50%) get a C-Value of 0.03. This value is much higher than the C-Value of the majority of the 2-grams: 0.008 (for 69% of the 2-grams). The repartition is also different. When considering 4 and 5-grams, the phenomenon is similar. The proportion of terms that have a C-value of 0.03 increase (60% of 4-grams and 52% of 5-grams).

4.3. Hierarchy

Figure 7: Distribution of C-Values according to n ; Wikipedia English

network
(weight : 1.000 , c-value : -1.000); networks ; networking ; networked
- **bayesian network**
  (weight : 0.190 , c-value : 0.173); bayesian networks
  - **includ bayesian network** :
    (weight : 0.012 , likelihood : 0.097); including bayesian networks ; include bayesian networks
- **feedforward network**
  (weight : 0.059, c-value : 0.033); feedforward networks
- **hopfield network**
  (weight : 0.074, c-value : 0.041);

In the sample of the hierarchy presented figure 7, were weights and C-values are indicated, we can see that it is possible to filter candidates so that the best one are kept. Notice that when a negative value is indicated that means that the term was not extracted from the text collection but automatically generated in order to build a upper level in the hierarchy.

5. CONCLUSION

The method we present in this paper can be used to help building a domain ontology. Because it is statistically based, it can be applied to any language even those for which no POS have been developed. However, when a POS is available in a language, it can help to extract candidates. Our weighting scheme as well as the term structuring can still be applied. Future work will evaluate the weighting scheme when POS is used.

Building ontology, even restricted to the “is-a” relationship between concepts is an issue not only for the semantic web but for many applications such as text classification, information retrieval or text summarization. Building such resources on a domain is time consuming and helping designers is of interest.

In some applications, the ontology should be totally correct. This is the case when the ontology is part of the browsing facilities (e.g. to access associated documents). Nevertheless, in some applications, ontology does not need necessarily to be completely correct. This is in particular the case when ontology is used internally by the system to represent information. In this case, ontology can be used to enrich the representations, even if some are distorted. We can parallel this fact with the indexing methods in information retrieval. It has been shown that indexing by stems is effective, even if it implies a certain lost information. However, this type of indexing is not possible if it is part of the interface provided to the users for whom stems (rather than phrases or terms) would not have any meaning.

ACKNOWLEDGEMENT

Development presented in this paper is part of the WS-Talk project supported by the European Commission under the 6th FP (COOP-006026). However views expressed herein are ours and do not necessarily correspond to the WS-Talk consortium. We also thank Nathalie Hernandez and Yannick Loiseau for their valuable comments.

REFERENCES

Aussenac-Gilles N., Biébow B., Szulman N., 2000, Revisiting Ontology Design: a method based on corpus analysis, Knowledge engineering and knowledge management: methods, models and tools, Int. Conf. on Knowledge Engineering and Knowledge Management. LNAI Vol 1937, Springer Verlag, pp 172-188.

Aussenac-Gilles N., Maedche A., 2002, Machine Learning and Natural Language Processing for Ontology Engineering, *workshop at ECAI 2002*.

Aussenac-Gilles N., Mothe J. (2004) Ontologies as Background Knowledge to Explore Document Collections, RIAO 2004, Coupling approaches, coupling media and coupling languages for information retrieval, pp 129-142.

Aussenac-Gilles N., Soergel D., 2005, Text Analysis for Ontology and Terminology Engineering, *In Applied Ontology*, IOS PressAmsterdam, Vol. 1, N° 1, pp 35 - 46.

Berman J., Automatic extraction of candidate nomenclature terms using the doublet method, 2005, BMC Medical Informatics and Decision Making, Vol. 5, N° 1, record 35, http://www.biomedcentral.com/1472-6947/5/35.

Chen L., Fankhauser P., Thiel U., Kamps T., 2005, Statistical relationship determination in automatic thesaurus construction, Conference on Information and Knowledge Management, pp 267-268.

Church K. and Hanks P., 1990, Word Association Norms, Mutual Information, and Lexicography, *Computational Linguistics*, Vol. 16, N°. 1, pp 22-29.

Corby O., Dieng-Kuntz R., Faron-Zucker C., 2004, Querying the Semantic Web with Corese Search Engine, *European conference on artificial intelligence*, pp 705-709.

Dkaki T., Mothe J., 2004, Combining Positive and Negative Query Feedback in Passage Retrieval, *In Proceedings of RIAO, Coupling approaches, coupling media and coupling languages for information retrieval*, pp 661-672.

Feng F., Croft W.B., 2001, Probabilistic Techniques for Phrase Extraction, *Information Processing Management*, Vol. 37, pp 199-220.

Frantzi K, Ananiadou S., Mima H., 2000, Automatic recognition of multi-word terms: the C-value/NC-value method, *International Journal on Digital Libraries*, Vol. 3, pp 115-130.

Goldman J-P., Nerima L., Wehrli E., 2001, collocation Extraction using a Syntatic Parser, *In Proceedings of the ACL'01 Workshop on collocation*, pp 61-66.

Hernandez N., Mothe J., 2004, An approach to evaluate existing ontologies for indexing a document corpus, *International Conference on Artificial Intelligence: Methodology, Systems, Applications*, Semantic Web Challenges, pp 11-21.

Hernandez, 2006, Ontologies de domaine pour la modélisation du contexte en Recherche d'information, PhD manuscript, Université Paul Sabatier, Toulouse, France.

Hearst M.A., 1992, Automatic Acquisition of Hyponyms from large Text Corpora, *Int. Conf on Computational Linguistic (COLING)*.

Maedche A., Staab S., 2001, Ontology Learning for the Semantic Web, *IEEE intelligent systems*. Vol. 16, N°2, pp. 72-79.

Maedche, A., Staab, S., 2003, Ontology Learning, In S. Staab & R. Studer (eds.) *Handbook on Ontologies in Information Systems*. Springer.

Morin E., Acquisition de patrons lexico-syntaxiques caractéristiques d'une relation sémantique, 1999, *TAL (Traitement Automatique des Langues)*, Vol. 40, N°1, pp 143-166.

Sanderson M., Croft B., 1999, Deriving concept hierarchies from texts, ***In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieva,l*** pp 206-213.

Séguéla P., 1999, Adaptation semi-automatique d'une base de marqueurs de relations sémantiques sur des corpus spécialisés, *Terminologies Nouvelles n°19*, pp 52-60.

Seretan V., Nerima L., Wehrli E., 2004, Multi-word collocation extraction by syntactic composition of collocation bigrams. *Recent Advances in Natural Language Processing III*, ed. by Nicolas Nicolov et al., pp 91-100.

Song M., Song I., Hu X., 2003, KP Spotter: A Flexible Information Gain-based Keyphrase Extraction System. *In Proceedings of the fifth ACM International Workshop on Web Information and Data Management*, pp 50-53.

Turney P.D., 2000, Learning Algorithms for Keyphrase Extraction, *Information Retrieval*, Vol. 2, pp 303-336.

Wang Y., Völker J., Haase P., to appear 2006, Towards Semi-automatic Ontology Building Supported by Large-scale Knowledge Acquisition, *In AAAI Fall Symposium On Semantic Web for Collaborative Knowledge Acquisition*. AAAI Press, Arlington, VA, USA.

Wu J-L. and Agogino A.M., 2004, Automatic Keyphrase Extraction with Multi-Objective Genetic Algorithms, *In Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pp 40104.3.