12-2008

Growing Fields of Interest: Using an Expand and Reduce Strategy for Domain Model Extraction

Christopher Thomas

Pankaj Mehra

Roger Brooks

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

Growing Fields of Interest

Using an Expand and Reduce Strategy for Domain Model Extraction

Christopher Thomas[1,2], Pankaj Mehra[1,] Roger Brooks[1] and Amit Sheth[2]

(1)HP Labs, Palo Alto; (2)Kno.e.sis Center, Wright State University, Dayton, OH
{thomas.258, amit.sheth}@wright.edu, {pankaj.mehra, roger.brooks}@hp.com

Abstract

Domain hierarchies are widely used as models underlying information retrieval tasks. Formal ontologies and taxonomies enrich such hierarchies further with properties and relationships associated with concepts and categories but require manual effort; therefore they are costly to maintain, and often stale. Folksonomies and vocabularies lack rich category structure and are almost entirely devoid of properties and relationships. Classification and extraction require the coverage of vocabularies and the alterability of folksonomies and can largely benefit from category relationships and other properties. With Doozer, a program for building conceptual models of information domains, we want to bridge the gap between the vocabularies and Folksonomies on the one side and the rich, expert-designed ontologies and taxonomies on the other. Doozer mines Wikipedia to produce tight domain hierarchies, starting with simple domain descriptions. It also adds relevancy scores for use in automated classification of information. The output model is described as a hierarchy of domain terms that can be used immediately for classifiers and IR systems or as a basis for manual or semi-automatic creation of formal ontologies.

1. Introduction

It is widely agreed on that having a formal representation of domain knowledge can leverage classification, knowledge retrieval and reasoning about domain concepts. Many envisioned applications of AI and the Semantic Web assume vast knowledge repositories of this sort, claiming that upon their availability machines will be able to plan and solve problems for us in ways previously unimaginable [1].

There are some problems with this vision. The massive repositories of formalized knowledge are either not available or do not interoperate well. A reason for this is that rigorous ontology design requires the designer(s) to have extensive domain knowledge and to fully comply with the underlying logical model, e.g. description logics in the case of OWL-DL. It is very difficult to keep a single ontology logically consistent while maintaining high expressiveness and high connectivity, let alone several ontologies designed by different groups.

Another problem is that Ontologies, almost by definition, are static blocks of knowledge that are not supposed to change frequently. The field of ontology was concerned with the essence and categorization of things, not with the things themselves. Our conceptualization of the world and of domains stays relatively stable while the actual things we encounter in the world change rapidly. When looking for information it is mostly these individual things that are of interest to us, not their categories. Keeping up with what is new has become an impossible task. Still, more than ever before we need to keep up with the news that are of interest and importance to us and update our worldview accordingly. One inspiration for this work was given by N.N.Taleb's Bestseller *The Black Swan* [2], a book about the impossibility to predict the future, but the necessity of being prepared for it. The best way to achieve this is to have the best, latest and most appropriate information available at the right time.We want to be the first to know about change, ideally, before it happens, at least shortly thereafter. The *Black Swan* paradigm for information retrieval is thus **"What will you want to know tomorrow?"** Document classification for news delivery needs to take recent changes in domains into account, ideally without the user's interference.

Document classification usually relies on a user-provided, annotated training corpus. Another option is for a system to slowly learn the users' interests from tagged documents. The downside to both methods is that tagging and training is always required. Realizing this shortcoming, we created Doozer, an application that generates restricted hierarchical domain models from readily available conceptual knowledge in the form of the community generated encyclopedia Wikipedia that organizes domain knowledge in a sparsely annotated graph structure. Its category structure resembles the class hierarchy of a formal ontology to some extent, even though many subcategory relationships in Wikipedia are associative

rather than being strict *is_a* relationships; neither are all categorizations of articles strict *type* relationships, nor are all articles representing instances. For this reason we refrain from calling the resulting domain model an ontology. Whereas formal ontologies that are used for reasoning, database integration, etc. need to be logically consistent, well restricted and highly connected to be of any use, domain models for information retrieval can be more loosely connected and even allow for logical inconsistencies. As of today, Wikipedia contains over 2.5 million topic pages organized in a vast category hierarchy. Every day, the number of articles in Wikipedia grows [3] and the quality of older articles increases [4]. In the long run, Wikipedia will likely be a comprehensive Encyclopedia that covers a large number of the concepts known to man. Hence we can assume that most domains of interest are represented as a network of articles on Wikipedia. This makes getting a comprehensive description of a domain a task of carving out a set of Wikipedia articles and categories that are most relevant to the domain. From this set of articles that describe relevant concepts, we can then extract the terms that best describe the concepts and set up Bayesian document classifiers that operate on these terms and the probabilities that these terms unambiguously identify the domain of interest.

The paper is structured as follows. Section 2 discusses related work. In section 3 we describe the model creation process in detail. Section 4 aims at evaluating the resulting models and section 5 finally concludes and gives an outlook toward future work.

2. Related Work

A large body of work is dedicated to the automatic creation of taxonomies or ontologies from text [5]. In [6], no structural knowledge of the domain was available to the system. The resulting hierarchy was generated solely by identifying expressive clusters in a hierarchy that was an artifact of a clustering process. Then, the most salient terms in these clusters were identified and used as labels. Other work has focused on combining linguistic analysis with statistical methods and formal concept analysis, see [7, 8]. The same group also recognized the use of automatically generated ontologies for clustering [9].

Works that have made use of the Wikipedia corpus to infer taxonomic knowledge include [10]. This work takes the category hierarchy and uses heuristics and NLP methods to identify those inter-category relationships that are actually *is_a* relationships.

All domain model generation efforts we are aware of go through the difficult task of analyzing language. Doozer bypasses the problems that arise because of syntactic and semantic ambiguities in free text by taking advantage of a community generated corpus that is free of ambiguities in its graph structure [11].

The question of classification based on a limited set of features has been addressed in [12]. The authors showed that a hierarchically built classifier can achieve high accuracies despite focusing on only a few words. In [13], Wikipedia is used to classify documents into a concept space. Here, we take the reverse direction by building the concept space first and then use it to determine which articles match it.

3. Domain Model Creation Workflow

In this section we describe the different steps involved in getting from a simple query or set of terms to a comprehensive domain model. The overall process follows an *Expand and Reduce* paradigm which allows us to first explore and exploit the concept space before reducing it to those concepts that are closest to our domain of interest. We decided to look at a domain of interest from three different levels that are user inputs to the system.

The focus domain, which is the actual point of interest, e.g. *Web 2.0, Cancer*. In Doozer, the focus domain is given by the user in the form of a *seed description*. The seed description will in most cases be a query, but it can be an initial list of Wikipedia terms.

The broader focus domain, which encompasses concepts that are immediately related to concepts in the focus domain, e.g. Social *Networking, Internet, Oncology*. The user describes the broader focus by a) selecting one or more broader categories of interest and b) optionally entering a second query. If it is not entered, this domain-query or context-query is set to be the same as the seed query. It is used to compute conditional probabilities for reduction.

The World View, which indicates how we look at the domain, whether e.g. the Information Science aspect is important for our interest in Web 2.0 or the social aspect. Hence, for the Web 2.0 example, we could choose the category *Information Science* or *Society* as the broad World View, both of which give different connections. This world view is generated by topologically sorting the categories of Wikipedia with respect to an arbitrarily chosen upper category. The assumption is that during the generation of the topic hierarchy, subcategories that are most important to a category are asserted as closer descendants than subcategories that are only marginally related. For example, both categories *Science* and *Society* are in Wikipedia's Main Topic classification. The category *People* is an immediate subcategory of *Society*, but to get to *People* from *Science*, we have to walk a path through *Social Sciences*, *Sociology* and *Humans*.

The idea behind *Expand and Reduce* is to first collect as many relevant results as possible, then evaluate these results, keep the most promising, categorize them with respect to the *world view* and intersect them with the *broader focus domain*.

Expansion

1) Full-text Search [14]
2) Graph-based expansion [15]
3) Category-growth

Reduction

1) Category-based reduction/intersection
2) Conditional pruning
3) Depth reduction

3.1. Expansion

This subsection describes the expansion steps taken to get from a simple domain description, such as a glossary or simply a seed query to a possibly exhaustive list of terms relevant to the domain (Figure 1). In the expansions steps recall is maximized to allow as many concepts as possible to be taken into account while maintaining a sensible focus on the domain of interest.

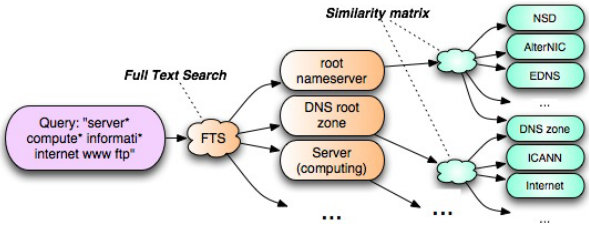

Figure 1: Steps (1) and (2) in the expansion process

3.1.1. Full Text Search – Exploring the knowledge space

Any indexed Wikipedia article that matches a query with a score[1] greater than a given threshold (or smaller than a given maximum rank, depending on user preferences) will be returned, regardless of whether it ultimately matches the desired focus domain or not. However, a carefully stated query will help maintaining the focus even in this early stage. The set of terms returned from this step is described in Formula (1). We chose to give the user the option of scored and ranked search because the Lucene score that is used is not always intuitive, especially when using more involved Boolean queries.

$$\boldsymbol{T}_{search}(\boldsymbol{query}) = \{\boldsymbol{title}(\boldsymbol{article}), \boldsymbol{article} \in \boldsymbol{hits}(\boldsymbol{query}) | \boldsymbol{score}(\boldsymbol{article}) > searchThreshold \vee rank(\boldsymbol{article}) < maxRank\} \quad \textbf{(1)}$$

[1] http://lucene.apache.org/java/docs/scoring.html

3.1.2. Graph-Based Expansion – Exploiting the knowledge space

For the graph based expansion of the initial set of articles, we use a method developed by HP labs Russia. The importance of adjacent articles is measured using a *weighted common neighbors metric* as defined in [15]:

The similarity of two articles in Wikipedia is defined as the sum of weights of their shared neighbors (articles that are linked to or link to the current article), normalized by the node degrees. Let M be the adjacency matrix of Wikipedia, N(a) stands for the neighborhood and *w(a)* stands for the weight of node *a*, and includes all the articles that link to or are linked to *a*. The semantic similarity between nodes *a* and *b* is then defined in formula (2), which is similar to the first iteration of SimRank[16], the only difference being the normalization factor and weights.

$$sim(a,b) = \frac{\sum_{\{i,j\,|\,\rangle N_i(a),N_j(b)\langle \in M\}}^{|N(a)|,|N(b)|} avg\left(w(N_i(a)), w(N_j(b))\right)}{avg\left(\sum_{i}^{|N(a)|} w(N_i(a)), \sum_{j}^{|N(b)|} w(N_j(b))\right)} \quad \textbf{(2)}$$

The weights *w* can vary for different document links considered. Figure 2 shows the different types of links on Wikipedia as described in [15]. The weights for each of these types of links were empirically determined. We emphasized on the *see-also* links because editors add these links usually to refer to highly relevant concepts. *Double* links also indicate that two concepts are mutually important for each other. For the final calculation of the similarity score, only the relative weights of the links are important. We gave the described *see-also* and *double* links double the weights of the other links. Then, the set of articles similar to an article *a* is (Formula 3):

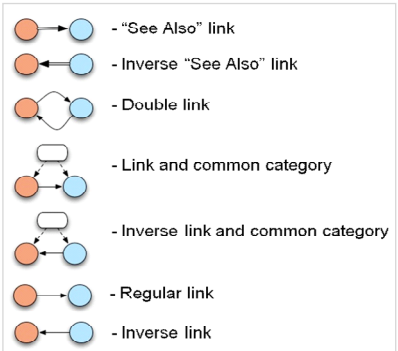

Figure 2: Types of links on Wikipedia

$$\boldsymbol{sim}(\boldsymbol{a}) = \{\boldsymbol{b} \in \boldsymbol{G}_{\boldsymbol{Wiki}} | \boldsymbol{sim}(\boldsymbol{a}, \boldsymbol{b}) > simThreshold\} \quad \textbf{(3)}$$

The final set of terms gained during the expansion steps is the union of the initial search results and their graph-based expansions.

3.1.3. Building a category hierarchy

Building a category hierarchy is an essential step for further pruning. In this step, the *World View* and the *broader focus come* into play. All non-empty categories up to the root category of the *broader focus domain* are incorporated in the initial hierarchy and connected by subcategory relationships with respect to the *World View* taken, not the entire graph structure of Wikipedia

3.2. Reduction

Whereas the expansion steps are used to gather knowledge in a recall oriented way, the reduction steps increase precision and reduce the set of terms to match the focus domain.

3.2.1. Probability-based reduction – Conditional Pruning

This reduction step operates on the basis of terms (in this case Wikipedia article titles), not categories. For each term in the list of extracted terms, we compute a relevance probability with respect to the domain of interest. Formula (4) shows this conditional probability computation. A probability of 1.0, for example would indicate that every time the term appears, it is within the domain of interest. Formula (5) shows the inverse: how significant is the term in the domain? Knowing both measures is important for the subsequent use of the created domain model in document classification. However, only the former is used for pruning. If the importance of a term is less than a predefined threshold $\boldsymbol{\varepsilon}$, it is discarded from the set of domain terms (formula 6).

$$\boldsymbol{p(Domain|Article)} = \frac{|\boldsymbol{query(Domain \cap Article)}|}{|\boldsymbol{query(Article)}|} \quad \textbf{(4)}$$

$$\boldsymbol{p(Article|Domain)} = \frac{|\boldsymbol{query(Domain \cap Article)}|}{|\boldsymbol{query(Domain)}|} \quad \textbf{(5)}$$

$$\boldsymbol{T_{final}(Domain)} = \{\boldsymbol{Article} \in \boldsymbol{T}|\boldsymbol{p(Domain|Article)} \geq \boldsymbol{\varepsilon}\} \quad \textbf{(6)}$$

3.2.2. Category-based reduction

After probabilistic pruning, some categories (and their subcategories) will be empty. These can by default be deleted. Furthermore, all categories that do not belong to the chosen *broader focus domain* are deleted immediately. If a term is categorized in more than one category, it is kept in the categories that are part of the broader focus domain, otherwise it is deleted. If the number of terms that remain in a category is below a given threshold, the terms are moved up to the next higher category in the hierarchy. The assumption here is that sparsely populated categories are probably not important for the domain, even though the terms in these categories are.

3.2.3. Depth Reduction

In many cases, after the category-based reduction, deep linear branches of categories remain as artifacts of the category building and deletion tasks. We assume that empty or unbranched category hierarchies can be collapsed without loss of relevant knowledge. This step reduces the depth and increases the fan-out of the domain model. Together with the previous step it reduces the number of resulting categories, which makes the model more manageable.

3.3. Synonym Acquisition

The Wikipedia article names are unambiguous identifiers and as such not necessarily of the form we are used to talking about the concept of the article. A domain model that is used for text classification needs to contain different synonyms for the concept of the article. One good source of synonyms is WordNet[17], but it requires to first unambiguously identify a match between a Wikipedia article name and a WordNet synset, which adds another level of uncertainty. Hence we decided to stay within the Wikipedia corpus and analyze the anchor texts that link to the respective pages. The probability that a term is a synonym of an Article name is given by formula 7); the conditional probability that a term links to an article:

$$\boldsymbol{p_{syn}(term, Article)} = \frac{|\boldsymbol{links\_to(term, Article)}|}{\sum_{\boldsymbol{a \in AllArticles}} |\boldsymbol{links\_to(term, a)}|} \quad \textbf{(7)}$$

The impact a synonym has on the probability that a Wikipedia article name is indicative of a domain is given by formula (8).

$$\boldsymbol{p(Domain|term)} = \max_{\boldsymbol{Article \in syn(term)}} \boldsymbol{p(Domain|Article)} * \boldsymbol{p_{syn}(term, Article)} \quad \textbf{(8)}$$

3.4. Serialization

The resulting domain model is serialized as an OWL file, which greatly facilitates visualization and further modification. We are aware of the fact that it does not meet the formal standards of OWL; for example, the Wikipedia category hierarchy is often associative rather than expressing formal *is_a* relationships. However, knowing about the limitations of the generated models, OWL as the W3-recommended ontology language seems the best way to make these models more easily accessible.

4. Experiments and Evaluation

The generated topic hierarchies can be evaluated in different ways. Subjectively, we can look at the hierarchies and term lists and get a feel for the coverage of the domain. Ideally we would evaluate the quality and utility of the generated topic hierarchies or ontologies by using the terms in a classifier and measuring its precision and recall and measure it with respect to a baseline classifier. Future work will

evaluate the classification accuracy of different domain models.

Guarino [18] suggests to compare a new ontology to a canonized domain conceptualization and then measure precision and recall with respect to the coverage of the ontology. We follow this route, but acknowledge some of the problems that occur, because (a) often we do not have such a high-quality domain conceptualization, and (b) the problem of mapping between concepts in both descriptions has to be resolved. We encounter these problems in our evaluation.

4.1. Comparison to related services

We created domain taxonomies and compared them with tools specializing in mining Wikipedia and human-composed glossaries.

Sets by Google Labs [19]: The service allows the user to input between one and five example terms that it expands to a longer list of related terms.

Grokker by Groxis, Inc. [20] allows the user to find and organize related concepts, and can be constrained to return only Wikipedia concepts.

PowerSet [21]: is a service to mine Wikipedia using either simple queries or natural language questions.

As a baseline, we compared against results obtained using Wikimedia search, available as the "search" button in Wikipedia.

4.2. Quantitative comparison of competing tools against a reference taxonomy

In the analysis, we used a glossary [22] of financial terms which has been pre-categorized into domains. In particular, we utilized the list of terms in the "federal reserve" and "mortgage" domains. The tools from Google, Grokker, Powerset, and Wikimedia as well as Doozer were queried with these two seeds to produce two domain lists per tool. In order to reduce the terms from the glossary down to only the ones found in a search for the respective seed topics in Wikipedia, we produced the reference list as the intersection of the respective glossary and Wikipedia search results. These reference lists then contain terms that the author of the glossary would consider relevant to the respective domains and that are also present in the corpus upon which the taxonomies are built.

The values of the F-measure were then computed with equal weight for precision and recall for the lists generated by the tools. The results are illustrated in Figure 3. Doozer's results are at least a factor of two improvement over those of the other tools. This difference in performance can be attributed to the amount of noise in the topic search results of Wikipedia.

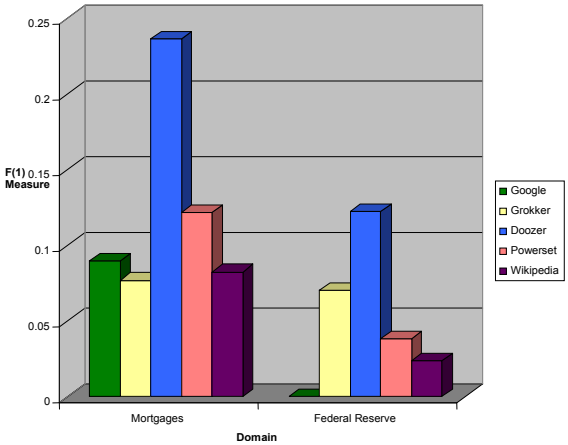

Figure 3: F-measures, computed against a reduced glossary, for the lists of terms generated by various mining tools

The results in Figure 3 provide evidence that blind use of topic search results of Wikipedia will have high rates of both false positives and false negatives if they are used as the sole basis for taxonomies. As described above, the approach reported in this work increases the recall primarily by exploiting the link structure of Wikipedia to find additional topics that are similar to an initial set of topics. Furthermore, we use domain relevancy statistics (weights and conditional probabilities) to prune intermediate lists, thereby increasing the precision of Doozer's results, as evidenced by the results in Figure 3.

4.3. Comparison against MeSH

Comparing the generated list of domain terms to a Gold Standard such as MeSH opens a new can of worms. Domain ontologies and glossaries usually contain terms for immediate domain concepts rather than terms that are highly indicative of a domain. The term *cancer*, for example is very important for, but not highly indicative of the oncology field. The content of the created domain models are meant to be used in retrieval and classification tasks. Nevertheless, in order to have a numerical evaluation of the model creation process, an automated Gold-Standard evaluation [5] is performed. We extracted all MeSH terms in the Neoplasms sub-tree to compare them against the terms in an automatically generated Neoplasms domain model.

Just like Wikipedia, MeSH is constantly evolving. In order to show how useful an automatic extraction of a domain model can be to stay up-to-date without investing human effort, the extracted domain model is compared against the Neoplasm sub-trees of 2004 and 2008.

Alignment of Wikipedia and MeSH is not in the scope of this work. Therefore the terms in the generated domain model are matched against two subsets of both MeSH Neoplasms versions using simple string matching techniques. Subset (1) is the full set of terms, (2) is the subset of terms that can actually be found in the Wikipedia titles and their synonyms and is thus the maximum number of matches we can possibly achieve with the current evaluation method (see Table 1). The comparison to the restricted set of MeSH terms accounts for the limitations that are imposed on Doozer by the underlying knowledge repository.

	MeSH Neoplasms (1)	Matches Wiki term (2)	Percent in Wikipedia
2004	405	147	36.3
2008	636	227	35.7

Table 1: Terms in the Neoplasm sub-tree of MeSH

We performed the comparison with the Neoplasms sub-tree gradually shifting from less restrictive to more restrictive settings, moving from recall- to precision-oriented runs. Table 2 shows these experimental settings. By changing the thresholds in various steps in the algorithm, we achieve more expansion or more reduction. The recall oriented runs had a low search threshold (more initial results), a low expansion threshold (more similar nodes) and a low domain-importance threshold ε (fewer nodes deleted because of conditional probability). In the precision oriented runs, higher thresholds were set.

For the experiment we chose biology as a world view, oncology as broader focus and this seed query:

Seed query and Domain query: (Adenoma Carcinoma Vipoma Fibroma Glucagonoma Glioblastoma Leukemia Lymphoma Melanoma Myoma Neoplasm Papilloma)

	Search Results	Expansion threshold	min p(Domain|Article)
1	40	0.5	0.1
2	40	0.5	0.4
3	25	0.8	0.5

Table 2: Experimental settings

Figure 4 shows the evaluation of these runs with respect to the MeSH versions of 2004 and 2008. \We achieve a precision of up to 48% wrt. the 2008 MeSH version and a recall of up to 78% wrt. MeSH 2004. One reason for not achieving higher scores is the different scopes of Wikipedia and MeSH, another is the different goals that MeSH and our generated domain models have. Looking at the resulting domain model of the high-precision run, out of a total of 222 extracted instances, 135 belong to the category *types of cancer*. Other categories and terms that are relevant to

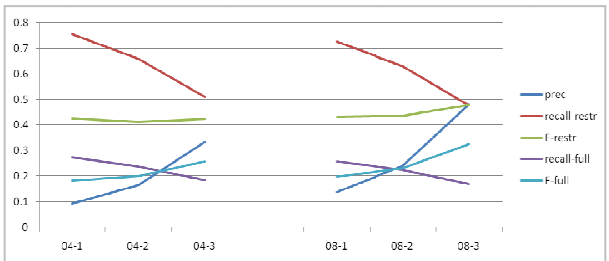

Figure 4: Evaluation wrt. full set of terms and Wiki subset of terms in MeSH versions of 2004 (04) and 2008 (08)

the neoplasms domain can also be found, such as *radiobiology* and *therapy* which amongst others share the instance *Radiation Therapy* as well as the terms *chemotherapeutic agents, Tumor suppressor gene* or *carcinogens*[2]. These additional, non-MeSH finds show that, Doozer's domain models and MeSH have different scopes. Whereas the MeSH C04 sub-tree restricts itself to listing different types of neoplasms, Doozer discovers many related concepts that are important for classification, but do not denote types of neoplasms. We take this as a strong indication that Doozer performs well in the task of producing domain hierarchies even in such a specialized domain as the neoplasms field.

5. Conclusion and Future Work

We presented the creation of topic hierarchies using a corpus of community-created, loosely structured knowledge. We showed that using a minimum set of input keywords, an intended focus domain and a world view, we can expand these to a sufficiently large set of domain terms categorized in a domain hierarchy. Our evaluation showed that an expand and reduce strategy on a well linked corpus such as Wikipedia results in domain models that are superior to the relevant terms extracted by Google sets and the Wikipedia specific search engines. A comparison with the widely accepted MeSH taxonomy showed that Doozer's domain models achieve high recall while maintaining sufficient focus even in highly specialized domains. That in mind, we believe that Doozer has the potential to facilitate the creation of comprehensive, formally rigorous domain ontologies and even mostly automate the creation of domain models used for information retrieval and classification.

[2] Due to the limited space, we cannot give a full explanation of the created domain models. A more in-depth discussion and the different models that were created can be found here:http://knoesis.wright.edu/research/semweb/projects/knowledge-extraction/

With respect to the use of the domain terms, future work will focus on the tight integration of the generated domain models with classifiers. For the generation of domain models, we are looking at using more background knowledge such as named relationships that are available on DBPedia [23] as well as building a system that automatically identifies named relationships between domain concepts. This will also lead to improving the quality of the category hierarchy. Analysis of text as described by Hearst [24] and by Ponzetto and Strube [10] can lead to identification of actual *is_a* relationships between concepts. Recent work in the area of relationship extraction will allow us to enrich the current hierarchies with binary relationships between the instances. The ambitious goal of this work will eventually be automated acquisition of domain ontologies that are formally more rigorous than what we can achieve today (using automated methods) and will require little or no further human involvement after the initial creation of the background knowledge on Wikipedia.

6. References

1. Berners-Lee, T., J. Hendler, and O. Lassila, *The Semantic Web*. Scientific American, 2001.
2. Taleb, N., *The Black Swan: The Impact of the Highly Improbable*. 2007: {Random House}.
3. Voss, J. *Measuring Wikipedia*. in *Proceedings International Conference of the International Society for Scientometrics and Informetrics*. 2005.
4. Thomas, C. and A. Sheth, *Semantic Convergence of Wikipedia Articles*, in *Web Intelligence*. 2007: Freemon, CA.
5. Brank, J., M. Grobelnik, and D. Mladenić, *A Survey of Ontology Evaluation Techniques*. 2005.
6. Kashyap, V., et al., *TaxaMiner: an experimentation framework for automated taxonomy bootstrapping*. International Journal of Web and Grid Services, 2005. **1**(2): p. 240-266.
7. Cimiano, P., A. Hotho, and S. Staab. *Comparing Conceptual, Divise and Agglomerative Clustering for Learning Taxonomies from Text*. in *ECAI*. 2004.
8. Cimiano, P., A. Hotho, and S. Staab, *Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis*. J. Artif. Intell. Res. (JAIR), 2005. **24**: p. 305-339.
9. Bloehdorn, S., P. Cimiano, and A. Hotho. *Learning Ontologies to Improve Text Clustering and Classification*. in *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the German Classification Society (GfKl 2005), Magdeburg, Germany, March 9-11, 2005*. 2006: Springer.
10. Ponzetto and M. Strube. *Deriving a Large Scale Taxonomy from Wikipedia*. in *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*. 2007.
11. Hepp, M., K. Siorpaes, and D. Bachlechner, *Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management*. IEEE Internet Computing, 2007. **11**(5): p. 54-65.
12. Koller, D. and M. Sahami, *Hierarchically classifying documents using very few words*. 1997.
13. Gabrilovich, E. and S. Markovitch, *Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis*. Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007: p. 6-12.
14. Banerjee, S., K. Ramanathan, and A. Gupta. *Clustering short texts using wikipedia*. in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 2007: ACM.
15. Turdakov, D., *HP Labs Summer Internship Report*. 2007. p. 14.
16. Jeh, G. and J. Widom. *SimRank: a measure of structural-context similarity*. in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002: ACM Press.
17. Fellbaum, C., *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. 1998: The MIT Press.
18. Sure, Y., et al., *Why Evaluate Ontology Technologies? Because It Works!* IEEE Intelligent Systems, 2004. **19**(4): p. 74-81.
19. Google-Labs. *Google Sets: Automatically create sets of items from a few examples*. 2008 [cited; Available from: http://labs.google.com/sets.
20. Groxis, I. *Grokker Features Overview*. 2008 [cited; Available from: http://www.groxis.com/grokker/pdfs/grokker_features_ENG.pdf.
21. Powerset, I. *PowerLabs Wikipedia search*. 2008 [cited 2008; Available from: http://labs.powerset.com.
22. Wheeler, A. and L.Wheeler. *Knowledge, Internet, Payment, and Security References*. [cited; Available from: http://www.garlic.com/~lynn/.
23. Auer, S., et al. *DBpedia: A Nucleus for a Web of Open Data*. in *Proceedings of ISWC 2007 (To Appear)*. 2007.
24. Hearst, M. *Automatic acquisition of hyponyms from large text corpora*. in *Proceedings of the 14th conference on Computational linguistics*. 1992: Association for Computational Linguistics.