Harvesting Domain-Specific Terms using Wikipedia

Su Nam Kim
Department of Computer Science and Software Engineering
The University of Melbourne
VIC, Australia
snkim@csse.unimelb.edu.au

Lawrence Cavedon
School of CS and IT
RMIT University
VIC, Australia
lawrence.cavedon@rmit.edu.au

Timothy Baldwin
Department of Computer Science and Software Engineering
The University of Melbourne
VIC, Australia
tb@ldwin.net

Abstract *We present a simple but effective method of automatically extracting domain-specific terms using Wikipedia as training data (i.e. self-supervised learning). Our first goal is to show, using human judgments, that Wikipedia categories are domain-specific and thus can replace manually annotated terms. Second, we show that identifying such terms using harvested Wikipedia categories and entities as seeds is reliable when compared to the use of dictionary terms. Our technique facilitates the construction of large semantic resources in multiple domains without requiring manually annotated training data.*

Keywords Domain-specific Terms, Self Training, Wikipedia

1 Introduction

The identification and use of **domain-specific terms**—i.e., terms that have significance or special meaning in a domain—have been leveraged in a range of NLP and related tasks (e.g., query expansion and cross-lingual text categorization [7], and named entity recognition [2]). However, construction of knowledge bases of such terms (e.g. *Agrivoc*, *UMLS*) has generally been manual, requiring high cost and time. Even hand-crafted resources often have limitations in quality, are typically small, and have the problem of needing to be maintained as new words are added or become significant within the domain.

Most previous work on automatically identifying and categorizing domain-specific terms has used supervised techniques (e.g. [1, 7, 5, 3]). This has the disadvantage of requiring a significant amount of hand-crafted training data; this in itself can require a sizeable investment for any new domain. Much less work has used unsupervised approaches (e.g. [6, 4, 8]).

Our primary aim is to define an unsupervised method to acquire domain-specific terms, allowing us to harvest a large number of domain-specific terms, and making the technique more practical. To achieve this, we leverage the Wikipedia folksonomy, i.e. a large set of user-contributed web documents hand-tagged with categories. Following Milne et al. [5], we contend that **Wikipedia categories** are domain-specific, and thus, Wikipedia categories and entries can replace manually-annotated domain-specific terms to enable self-supervised learning. Wikipedia has previously been shown to be effective training data for self-supervised learning in information extraction tasks, as reported by Wu and Weld [9]. Further, we can use the harvested domain-specific terms and subcategories as seeds/a training corpus to expand the set of domain-specific terms.

Vivaldi and Rodríguez [8] have previously made use of Wikipedia categories, page content and the category hierarchy to retrieve Wikipedia entries as domain-specific terms. Despite similarity between the approaches, our main goal is to evaluate the usability of automatically extracted Wikipedia terms as a self-training corpus. Also, as shown in previous work, supervised methods (based on contextual semantic similarity) generally outperform unsupervised ones. Thus, we focus on extracting a smaller number of high quality Wikipedia terms for training supervised methods, rather than aiming at large-scale unsupervised term extraction. Another contribution of this paper is to evaluate the domain-specificity of Wikipedia. Despite simple overlap checking between Wikipedia

Proceedings of the 16th Australasian Document Computing Symposium, Canberra, Australia, 2 December 2011.

and *Agrivoc* by Milne and Witten [5], it is not clear how well Wikipedia categories correlate with domain-specificity, especially in terms of the category hierarchy. Thus, we test this using human annotators.

One challenge with Wikipedia is that categories become highly specialised and fragmented, and paradoxically less domain-specific [5], as we move further away from the top of the category hierarchy. To combat this, we restrict our selection of subdomains to the top-N categories of the hierarchy, to maximize utility and coverage. To validate the usability of Wikipedia information, we used human annotators to verify the domain-specificity of selected subcategories. Second, to build an inventory of domain-specific terms, we project the task as a classification task. That is, we classify target terms with their domain-specificity relative to the target domain using context-based semantic similarity. To do so, we use extra features from Wikipedia; in particular, we use Wikipedia category entries plus the words contained in the document's first-sentence to describe entries. In evaluation, we test the performance when using of first-sentence words from Wikipedia articles by comparing to the use of snippets.

2 Wikipedia Categories and Domain-Specificity

2.1 Building a Category Hierarchy

Milne and Witten [5] demonstrated that Wikipedia entries contain a significant number of domain-specific terms found in *Agrivoc*, a hand-created domain resource. The authors also showed that the overlap between Wikipedia entries and *Agrivoc* increases with higher degree of domain-specificity. However, due to different degrees of domain-specificity of categories, their technique needs to be modified to construct a proper category hierarchy due to cyclic links. Further, in lower levels of the category hierarchy, the degree of domain-specificity decreases. We thus model domain-specificity in terms of the levels of hierarchy in the category tree. As an illustration of the domain-specificity of Wikipedia entries from specific categories, see the Wikipedia category hierarchy for *Internet* and its subcategories in Figure 1.[1]

In the category hierarchy, nodes indicate categories, and links are between categories. The category hierarchy was mined using breadth-first search, starting from the root category, *Internet*, then recursively adding subcategories linked from the current category set. During this process, we found two issues: (1) a category may have multiple super-categories, and (2) a category may have no super-category. The first problem was addressed by not linking super-categories of a node that already had a super-category in the current category set: e.g. *Cat_C* is linked as a super-category for *Cat_A* but *Cat_B* is not. For the second problem, we observed that when a category with no super-category is linked to sub-categories, these sub-categories generally have an alternate super-category; hence, we can disconnect the original sub-category: e.g., if we disconnect the link between *Cat_E* and *Cat_D*, *Cat_D* is still in the hierarchy. By iterating this process, we constructed a tree with 28 hierarchical levels and 478,332 unique categories, starting with *Internet* as the root.

Level	$[0,1)$	$[1,2)$	$[2,3)$	$[3,4)$	4
L1	0	1	0	19	3
L2	0	0	28	114	0
L3	2	10	78	193	0
L4	3	32	196	165	0
All	5	43	302	491	3

Table 1: Scores of sub-categories for degree of domain-specificity with respect to the *Internet* domain

In the Wikipedia category hierarchy, we found two issues related to the measure of domain-specificity. First, even if categories seem related, they may be placed at different levels: e.g. *Web 1.0* and *Web 2.0* are placed at different levels. Currently, this problem is addressed manually by the Wikipedia community and we do not discuss it here. Second, the degree of domain-specificity may vary even at the same level: e.g. *Web service specification* and *Acid tests* are both located under *Web standards* but receive different domain-specificity annotations from the human annotators. We decided to consider categories down to 5 levels in the tree as domain-specific: this was selected empirically. We leave as future work the relation between tree-depth and domain-specificity.

2.2 Verifying Domain-Specificity of Category

To verify that Wikipedia sub-categories are domain-specific w.r.t. a target domain, we asked three human annotators to score the domain-specificity between a given sub-category and the main category, i.e., *Internet*. The scores are between 0 and 4, where 0 means no relatedness and 4 means domain relatedness with high confidence. We assume that if the score of a sub-category is at least 2, then the sub-category is domain-specific relative to *Internet*. Prior to the actual annotation, the human annotators were trained using other data. In our case, we trained them over the domain *Disease* and its sub-categories. Table 2.2 shows the distribution of average scores over *all* selected sub-categories (*LAll*) and selected sub-categories at each level (*L1-4*). 58.53% of sub-categories have scores of at least 3, and 94.31% of sub-categories have scores of at least 2. The agreement between judges was $R = 0.52$ using Spearman rank correlation. We conclude that identifying domain-specific sub-categories using our method is reliable, though with decreasing reliability with increasing depth in the hierarchy.

[1] Constructed using the Nov 2009 version of Wikipedia.

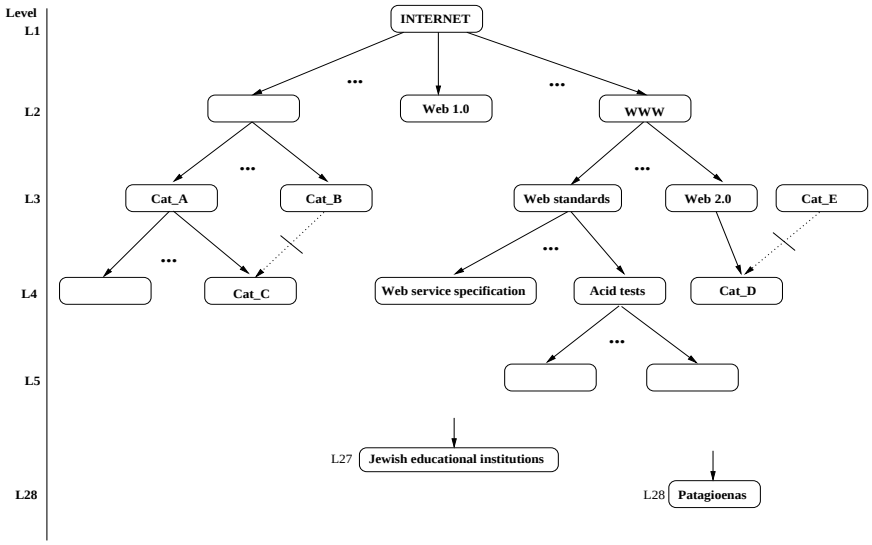

Figure 1: An example of the Wikipedia category hierarchy for *Internet*

3 Domain-Specific Term Identification

3.1 Contextual Semantic Similarity

Our main focus here is to determine whether the use of Wikipedia categories and entries can replace manually-crafted training data for building an inventory of domain-specific terms. We also investigate whether the first sentence of Wikipedia entries can be used as reliable context in identifying domain specificity, much like snippets. To test this second hypothesis, we use a semantic similarity metric using context words in an N-word window, as domain-specific terms often share similar co-occurring words.

To measure semantic similarity among domain-specific terms and targets, snippets are often used as context (e.g. [7]) due to their ease of acquisition and their relatively large size. However, retrieving snippets for large data sets can be time-consuming; snippets may also contain significant noise. On the other hand, we observed that dictionary definitions of words often contain frequent words in the context; however, the coverage of existing dictionaries is relatively low. As an alternative, we follow Kazama and Torisawa [2], who showed that the first sentence of a Wikipedia page is descriptive and can be used as a definition for a Wikipedia entry. For example, the definition of *Microsoft FrontPage* is *A WYSIWYG HTML editor and web site administration tool from Microsoft....* Words such as *HTML, editor, web* often co-occur with context words related to *DreamWeaver*.

3.2 Experiments and Results

In evaluating domain-specific term identification, our interests are in measuring: (1) the quality of Wikipedia terms as training data by comparing them to the use of hand-crafted dictionary terms; and (2) the utility of first-sentence words from Wikipedia in comparison to the use of snippets.

To determine the domain-specificity of categories, we first constructed a category hierarchy starting at *Internet*, and chose to explore categories found down to the 5th level (selected categories), resulting in 19,460 entries over 844 categories. To collect gold-standard data, we took a portion of Wikipedia entries which were already identified as being domain-specific to *Internet*: i.e., we used Wikipedia entries containing up to 3 *Internet*-related document categories. Further, we chose to use only Wikipedia entries that were composed exclusively of ASCII characters and numerals (but not numerals only) as our data, to extract clean snippets as instances. We separately collected an equivalent number of Wikipedia entries from *non-Internet* categories as test and training data using the same selection criteria. We restricted all *non-Internet* entries to have only one category, and every such term to belong to a different category. Finally, we used 20% and 80% of terms for test and training data, respectively. For the test data, we had two human annotators verify the domain-specificity of each entry to *Internet* or *non-Internet*; the inter-annotator agreement was $\kappa = 0.97$.

For features for the test data, we used snippets as a source of context words, since the target words do not necessarily occur in Wikipedia. As features of Wikipedia terms in the training data, we tested snippets and first sentence words from Wikipedia. To collect snippets, we used the top-1000 retrieved snippets based on querying the terms in the test and training data using `Yahoo!BOSS`,[2] then filtered out stop words. The use of first-sentence words as training data significantly reduces the time required to collect

[2] `http://developer.yahoo.com/search/boss/`

Feature I/N	Value	Frequency F5/F1	F10/F2	F15/F3
Training Source : Dictionaries				
D/S	DF	56.73%	57.35%	55.23%
	TF	**70.65%**	70.45%	67.75%
D/F	DF	60.87%	51.19%	49.48%
	TF	**77.23%**	77.02%	56.63%
D/C	DF	52.43%	57.56%	55.85%
	TF	73.29%	**73.65%**	73.29%
Training Source : Wikipedia				
S/S	DF	84.99%	**85.14%**	83.00%
	TF	77.69%	74.73%	71.85%
F/F	DF	70.08%	52.28%	51.24%
	TF	**85.20%**	70.34%	64.18%
C/C	DF	71.33%	73.29%	77.17%
	TF	**84.78%**	84.42%	84.11%

Table 2: **Accuracy of Term Identification**: *I/N* in the Feature column indicates of **I**nternet and **N**on-Internet terms, where *D,S,F,C* indicate **D**ictionary definition, **S**nippets, **F**irst-sentence words, and **C**omplete Wikipedia pages, resp.; *DF/TF* refers to the feature representation (document or term frequency); *FN* (e.g., *F1* and *F5*) indicate that the minimum term frequency is N.

context words. For comparison, we also tested use of the complete Wikipedia page as context. Note that due to the availability of snippets and first sentences from Wikipedia pages, the test data consists of 975 *Internet* terms and 977 *non-Internet* terms; the training data is made up of 3,906 terms for each of *Internet* and *non-Internet*.

For comparison, we collected 4,369 dictionary glossaries for *Internet* training data from 4 different sources.[3] After filtering duplicated terms and 19 terms that were also in the set of test terms, we obtained 2,972 terms as our hand-crafted *Internet* training data. As features for dictionary terms, we used dictionary definitions. For *non-Internet* training data, we used the same Wikipedia terms.

Table 2 shows the performance of our domain-specific term identification experiments using a linear-kernel support vector machine with various inputs. For comparison, a majority-class baseline (50.05%) and accuracies using the dictionaries are shown, since there are no other comparator systems due to the paucity of comparable unsupervised approaches and the unavailability of manually-crafted training data.

Compared with best performance using dictionaries (77.23%), our method achieved 85.20% accuracy at its best. This suggests that our method performs better than existing dictionaries at identifying domain-specific terms. The errors occur for the following reasons: (1) our training data contains noise; (2) the domain-specificity of the training data varies; (3) the size of the training data is relatively small; and (4) the semantic similarity metric can introduce error. Likewise, errors introduced in the approach using dictionaries are observed to be similar. We also note that the lower performance using dictionary terms is due to the smaller number of training terms; dictionary terms also seem to contain only more general words that occur frequently in *Internet*. We further found that using first-sentence words as context performs as well as using snippets — *85.20%* for first-sentence words vs. *85.14%* for snippets — and even outperformed the use of complete Wikipedia pages. Considering that acquiring snippets for large training terms is time-consuming, using first-sentence words not only produced the higher accuracy but also has the benefit of efficiency.

[3] `www.webopedia.com/Top_Category.asp`, `www.cnet.com/Resources/Info/Glossary/index.html`, `www.sharpened.net/glossary/`, `www.matisse.net/files/glossary.html`

4 Conclusions

In this paper, we explored techniques for extracting domain-specific terms from Wikipedia, and used these as seed/training data to predict the domain-specificity of unseen terms. We also showed that first-sentence words from Wikipedia entries are a useful feature for measuring semantic similarity among terms. In evaluation, we manually verified the domain-specificity of selected sub-categories and demonstrated the technique by identifying domain-specific terms over the Wikipedia *Internet* category. We believe that this method can be used to efficiently harvest large collections of domain-specific terms in a range of domains, significantly reducing time and cost for this task.

References

[1] Patrick Drouin. Detection of domain specific terminology using corpora comparison. In *Proceedings of the fourth international Conference on Language Resources and Evaluation*, pages 79–82, Lisbon, Portugal, 2004.

[2] Jun'ichi Kazama and Kentaro Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.

[3] Mitsuhiro Kida, Masatsugu Tonoike, Takehito Utsuro and Satoshi Sato. Domain classification of technical terms using the web. *Systems and Computers*, Volume 38, Number 14, pages 2470–2482, 2007.

[4] Su Nam Kim, Timothy Baldwin and Min-Yen Kan. Extracting domain-specific words — a statistical approach. In *Proceedings of the Australasian Language Technology Workshop 2009 (ALTW 2009)*, pages 94–98, Sydney, Australia, 2009.

[5] David Milne, Olena Medelyan and Ian H. Witten. Mining domain-specific thesauri from wikipedia : A case study.

In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 442–448, Washington, USA, 2006.

[6] Youngja Park, Siddharth Patwardhan, Karhik Visweswariah and Stephen C. Gates. An empirical analysis of word error rate and keyword error rate. In *Proceedings of International Conference on Spoken Language Processing*, Brisbane, Australia, 2008.

[7] Leonardo Rigutini, B. Liu and Marco Magnini. An em based training algorithm for cross-language text categorization. In *Proceedings of the Web Intelligence Conference (WI)*, pages 529–535, Compiegne, France, 2005.

[8] Jorge Vivaldi and Horacio Rodríguez. Finding domain terms using wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Valetta, Malta, 2010.

[9] Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the ACL (ACL 2010)*, pages 118–127, Uppsala, Sweden, 2010.