ISSN 1346-5597

National Institute of Informatics

NII Technical Report

Typing Software Articles with Wikipedia Category Structure

Liang Xu, Hideaki Takeda, Masahiro Hamasaki, and Huayu Wu

NII-2010-002E
July 2010

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

Typing Software Articles with Wikipedia Category Structure

Liang Xu[1], Hideaki Takeda[2], Masahiro Hamasaki[3], Huayu Wu[1]

[1] School of Computing, National University of Singapore
[2] National Institute of Informatics (NII)
[3] National Institute of AIST, Japan

Abstract. In this paper we present a low-cost method for typing Named Entities with Wikipedia. Different from other text analysis-based approaches, our approach relies only on the structural features of Wikipidia and the use of external linguistic resources is optional. We perform binary classification of an article by analyzing of the names of its categories as well as the structure. The evaluation shows our method can be successfully applied to the 'software' category (F 80%).

1 Introduction

Named Entity Recognition (NER) is a well-known task in the field of information extraction to classify atomic elements in text to a set of predefined categories such as person, location and organization. The problem we consider in this paper can be seen as a specialization of NER which seeks to classify a named entity as 'software' or not.

Along with the fast development of the IT industry, both software and their associated technological terminologies are constantly evolving. It is expensive for handcrafted databases such as Wordnet[1], which are of high quality but limited coverage, to keep up with the pace. Naturally the knowledge base we choose for our task is Wikipedia[1], the largest encyclopedia available. It is contributed by voluntary users and growing all the time.

Wikipedia differs from unstructured text sources in the set of structural features that it provides[2], including articles, hyperlinks, category structure, infoboxes, etc. By analyzing the articles in Wikipedia, [5][6] have performed NER tasks focusing on persons and locations. Their methods are Wordnet-based and sensitive to the texts in the articles, especially the first sentence which is supposed to define the entity and its type. Hyperlinks have been shown to be useful in measuring the association of articles[3][4]. For the purpose of NER tasks, however, we need relationships that are stronger than association in general. An infobox is a set of relational-style tuples aimed to provide structured summary of an article. Its use and accuracy are however constrained by the limited types of

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

infoboxes available and the knowledge of the contributors. Moreover, the semi-automatic process of generating new templates and infobox migration have been identified for causing inconsistencies in infoboxes[7].

In this work, we tackle the identification of software articles using the category structure in Wikipedia. We classify an article by analyzing the set of categories it has and their positions in the category structure. Our approach differs from Wordnet-based NER[5][6] in the following aspects: (a) our use of external knowledge bases such as Wordnet is optional; (b) articles-based analysis is sensitive to the senses of the keywords considered (In [5], only the first sense listed in Wordnet is used). In our approach, the problem of disambiguation is automatically resolved by analyzing the category structure.

2 Our approach

While it is natural to assume that an article belonging to software should be listed somewhere under 'Category:software'[2], a simple peek into the 'Category:software' convinced us that the kind of information listed under it is very diverse. Besides categories like 'Category:Software by company' and 'Category:Video games developed in Japan' which show strong indication that the articles in them belong to software, it is also easy to find categories like 'Category:Software companies', 'Category:Businesspeople in software' and 'Category:Software industry in China' which probably contain articles that do not. In addition, to classify categories like 'Category:Terminal pagers' and 'Category:Palm webOS' requires specific domain knowledge that we may not possess. To solve this problem, our approach proceeds in the following steps.

2.1 Category decoding

Some common patterns of Wikipedia categories have been identified in[8], which are used for relation extraction. With the task at hand, we are particularly interested in the following patterns:

- **X [VBN IN] Y** Example, Video games developed in Japan
- **X [IN] Y** Example, Software industry in China
- **Y X** Example, Software companies
- **X by Y** Example, Software by company

In all cases, X serves as the dominant constituent which types the category. With the decoding of category names, we can define S-Category which are categories that are highly likely to contain articles belonging to software.

Definition 1 (S-Category). *We say a category is an S-Category if the dominant constituent in its name is a hyponym of software (... is a kind of software) or is highly likely to be a type of software.*

[2] Category:software denotes the category in Wikipedia with name 'category'. Other categories are denoted in the same way.

For example, 'Category:Video games developed in Japan' is an S-Category because its dominant constituent 'video game' is a hyponym of software. In order to recognize S-Categories, we need to compile a list of dominant constituents that satisfy this definition. In our list, there are terms like 'client' and 'library' which have many senses other than being hyponyms of software, and terms like 'components' and 'tools' which are not hyponyms of software. However, our observation shows that the categories having them as main constituents are highly likely to be a kind of software if they are *under the software categories within limited levels.*

Our method to extract the list is based on the Wikipedia category structure. Every category in Wikipedia has a set of parent-categories and a set of child-categories. The Wikipedia category structure is a graph where categories can be considered as points connected by edges to parent-categories and child-categories.

Definition 2 (Parent Edge). *A directed edge from Category:X to Category:Y is a parent edge if and only if Category:Y is in the set of Category:X's parent categories.*

Definition 3 (Ancestor Path). *A directed path from Category:X to Category:Y is an ancestor path if and only if the path only contains parent edges.*

We use the set of articles with software infobox as positive instances (Some manual filtering is required because infobox is not perfectly accurate. For example, some software infoboxes are used in articles about programming languages and software companies). For every category of each positive instance, we extract the set of ancestor paths from that category to Category:software. The main constituents of the categories along the paths are then extracted and added to our list with record of their numbers of occurrences. The main constituents with low occurrences are removed from the list.

2.2 Category evaluation

Our classification of Wikipedia articles are based on the evaluation of their categories. Given an article to be classified, we first assign a score to each of its categories by applying Algorithm 1.

If no ancestor paths can be found from this category to Category:Software within a level limit, a score 0 is assigned to it. In our experiment, we used level 10. If such paths can be found and the category is an S-Category, it is given a score 1.0. Otherwise, the score of the category is the maximum score among the scores of the ancestor paths found. The score of an ancestor path is calculated by dividing the number of S-Categories along the paths by the number of categories. The heuristic we apply here is that, the more S-Categories a path has, the more likely that it is a IS-A paths, and the higher score it will get for containing articles belonging to software.

The final score we assign to an article is the sum of its categories' scores divided by its number of categories. It is a score between 0 and 1. The higher the score, the more likely this article belongs to software.

Algorithm 1: AssignScore(Category:X)

Data: Category:X is a category in Wikipedia
Result: Score of Cateogry:X

```
1  if no ancestor path can be found from Category:X to Category:software within a
   level limit then
2      return 0;
3  end
4  else if Category:X is a S-Category then
5      return 1.0;
6  end
7  else
8      double maxScore=0.0;
9      for each path in the set of ancestor paths do
10         double score=number of S-Categories in the path / number of categories in the path;
11         if maxScore<score then
12             maxScore=score;
13         end
14     end
15 end
16 return maxScore;
```

3 Experiments and results

Our experiment used the Wikipedia dump created at the time of 2007 Feb 6th. Parsing and accessing the Wikipedia dump is based on the Java library introduced in [9].

We evaluated our classification scheme on the set of articles with software infobox. Out of the 3239 articles with software infobox, 2800 have score greater than or equal to 0.5. This conforms to our intuition the software infobox is a strong indication of articles belonging to software. There are a number of reasons why an article with software infobox gets a low score: (a) the article in fact does not belong to software. For example, articles belong to programming languages were commonly misconceived as software and assigned software infoboxes. Their Wikipedia categories commonly have 'programming languages' as their dominating constituents, which are not in our list for identifying S-Categories. Their articles received low scores as a result; (b) the article does belong to software but its categories do not provide sufficient information. For example, we have found an article with software infobox but only two administrative categories; (c) the article does belong to software. However, its categories are given low scores, which indicates the limitation of our list for identifying S-Categories.

In addition, we used a random set of articles from Wikipedia containing 1000 articles for testing. There are 46 articles with scores greater than 0. A manual evaluation was conducted on them. 0.5 was used as the threshold for decision and we have precision 69.57%, recall 94.12% and F-measure 80%.

4 Conclusion

With Wikipedia becoming the largest encyclopedia, how to extract high quality data from it has attracted a lot of research attention. Methods that rely on handcrafted high quality linguistic sources are constrained by their coverage. This paper presents a unique approach that makes use of the structural features available in Wikipedia only. We believe our methods are promising with the constant evolvement of the category structures in Wikipedia. We are extending our method for NER of other categories and tuning our scoring schemes for higher accuracy.

References

1. A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM 38,* 11, 39–41.
2. O. Medelyan, D.Milne, C.Legg, and Witten. I. H. 2009. Mining meaning from Wikipedia. *Int. J. Hum.-Comput. Stud 67,* 9, 716–754.
3. D.Milne and Witten. I. H. 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *AAAI*.
4. Lev Muchnik and Royi Itzhack and Sorin Solomon and Yoram Louzoun. 2007. Self-emergence of knowledge trees: Extraction of the Wikipedia hierarchies. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics*.
5. A. Toral, and R. Munoz 2006. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. *EACL 2006*.
6. D. Buscaldi and P. Rosso 2007. A comparison of methods for the automatic identification of locations in wikipedia. *GIR '07: Proceedings of the 4th ACM workshop on Geographical information retrieval*.
7. Wu, F. and Weld, D. S 2008. Automatically refining the wikipedia infobox ontology. *the 17th international Conference on World Wide Web*.
8. V. Nastase and M. Strube 2008. Decoding Wikipedia Categories for Knowledge Acquisition. *Proceedings of the 23 AAAI Conference on Artificial Intelligence*.
9. T. Zesch, C. Mller and I. Gurevych 2008. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.