Infobox Suggestion for Wikipedia Entities

ABSTRACT

Given the sheer amount of work and expertise required in authoring Wikipedia articles, automatic tools that help Wikipedia contributors in generating and improving content are valuable. This paper presents our initial step towards building a full-fledged author assistant, particularly for suggesting infobox templates for articles. We build SVM classifiers to suggest infobox template types, among a large number of possible types, to Wikipedia articles without infoboxes. Different from prior works on Wikipedia article classification which deal with only a few label classes for named entity recognition, the much larger 337-class setup in our study is geared towards realistic deployment of infobox suggestion tool. We also emphasize testing on articles without infoboxes, due to that labeled and unlabeled data exhibit different distributions of features, which departs from the typical assumption that they are drawn from the same underlying population.

1. INTRODUCTION

Wikipedia has gained rapid growth and enormous popularity since its inception. The now largest encyclopedia in the world is the product of collective intelligence. In Wikipedia authors collaboratively contribute not only article content but also folksonomy such as infoboxes, categories, and the Wikipedia category hierarchy. Given the sheer amount of work and expertise required in this authoring process, automatic tools that help Wikipedia contributors in generating and improving content are valuable. This paper presents our initial step towards building a full-fledged author assistant, particularly for suggesting infobox templates for articles.

An infobox is a table of attribute-value pairs displayed on the top-right corner of a Wikipedia article. The majority of Wikipedia articles describe real-world *named entities* (in contrast to general concepts). Their infoboxes summarize important facts of corresponding entities. Figure 1 shows the Wikipedia page for Jawed Karim, including its infobox. In addition to improving the quality and readability of articles within Wikipedia, information from Wikipedia infoboxes has also been used in several high-profile applications outside of Wikipedia, including the social database Freebase [3] and Google's Knowledge Graph [1] which directly displays infobox information in Google search results.

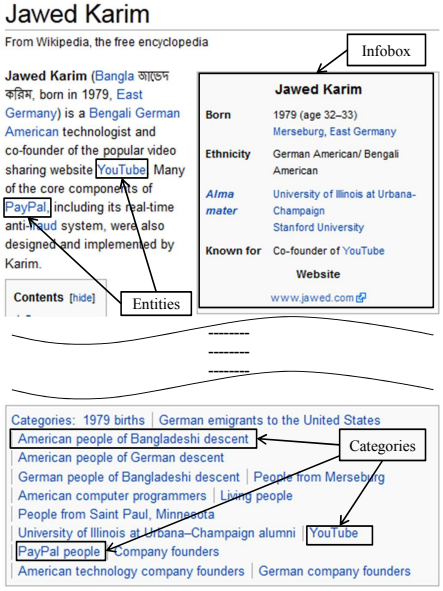

Figure 1: An example Wikipedia article (photo removed due to space limitations).

An *infobox template* contains common attributes shared by entities of the same "type". Figure 2 shows the template of the infobox in Figure 1. Note that PERSON is the name of the template used in this infobox. In other words, the entity belongs to type PERSON. In the 2008-07-24 snapshot of English Wikipedia, there are 1,646 infobox template types. The most common types include SETTLEMENT, FOOTBALL_BIOGRAPHY, FILM, and so on. When authoring the article for an entity, Wikipedia contributors can collectively decide whether to add an infobox and if so, which infobox template to use and which attributes from the template to be included in the infobox. Infobox templates are useful in several ways. They provide convenience to contributors in authoring articles; they effectively enforce a typing system that should be followed within Wikipedia; and they also help users in navigating and exploring articles, e.g., by finding related entities of the same type.

Among about 1.8 million Wikipedia articles in the 2008-07-24 snapshot (excluding disambiguation pages, list pages, and so on), about 55% of the articles do not have infoboxes, especially those that are new and less popular. A tool that can automatically generate infoboxes for articles is thus appealing because such a bootstrapping tool will motivate and facilitate contributors in improving article quality. Given an article and an infobox template, the tool would need to decide which attributes from the template to include

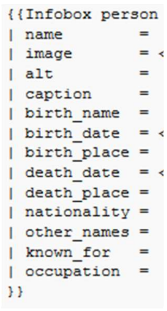

Figure 2: An infobox template.

in the infobox and populate the attributes with values, which can be possibly learned from the content of the article itself. Such is a non-trivial task. Wu et al. [14, 15] have made substantial progress on this line of work.

However, even before generating infobox attribute values for an article, we must choose a type (i.e., infobox template). Given the large number of interrelated infobox templates, manual assignment of infobox templates to articles can be time-consuming and error-prone. This factor perhaps contributes to the fact that more than half of Wikipedia articles have no infoboxes.

We build SVM classifiers to suggest infobox template types, among many possible types, to Wikipedia articles without infoboxes. The classifiers use a combination of several intuitive features, including article content, category, and related entities. They together attain better classification accuracy than individual features.

Prior works on classifying Wikipedia articles [12, 13, 4, 8, 9, 6, 11, 10] are for named entity recognition (NER) [7] instead of suggesting infobox template types. The consequence is that they only deal with very small number of classes (between 3 and 18) such as PERSON, ORGANIZATION, and LOCATION, which is also the classic setup in NER-related studies. In contrast, the much larger 337-class setup in our study is geared towards realistic deployment of infobox suggestion tool. Having more classes makes it more challenging to achieve satisfactory classification accuracy, as it is much less possible to hit a correct class accidentally.

The labeled data (articles with infoboxes) and unlabeled data (articles without infoboxes) in our scenario exhibit different distributions of features. This is an interesting departure from the assumption in typical classification problems that labeled and unlabeled data are drawn from the same underlying population. The reason is exactly why unlabeled articles are not labeled (i.e., having no infoboxes). Such articles are less mature due to various reasons (relatively newer, less popular, less experienced writers, or simply less information available). Hence they tend to be shorter, having fewer and less accurate categories, and having no infoboxes. We believe it is important to test on articles without infoboxes, due to two reasons– (1) From practical viewpoint, it is more urgent to assign infoboxes to articles without any than to assign additional infoboxes to articles that already have some. (2) An approach attaining good accuracy on labeled articles does not necessarily achieve equally good accuracy on unlabeled articles, due to their different characteristics mentioned above. This is also verified by our evaluation results. While article categories produce more accurate results on labeled articles, words in article content achieve better accuracy on unlabeled articles.

The work most closely related to ours is [14] as they also predict infobox types for articles. However, their classification is based on a simple rule– an article is assigned to a type if (1) the article is within a Wikipedia list page whose title contains the type name and (2) the article has a category whose name contains the type name. This approach is not applicable on articles that do not satisfy the arguably 2 strong conditions. The approach was tested on 4 classes (COUNTY, AIRLINE, ACTOR, UNIVERSITY), in comparison with the 337 classes in our case. Furthermore, they have only tested on articles with infoboxes (hidden during testing).

2. METHODOLOGY

The majority of Wikipedia articles describe named entities. These named entities are the focus of this work. We will use article and entity interchangeably. There are two kinds of Wikipedia entities– the ones with infoboxes (labeled entities) and the ones without infoboxes (unlabeled entities). The type of the infobox template of an entity is considered the class label of the entity. We consider labeled entities as training examples. An entity may have multiple infoboxes. We only include in training examples those labeled entities with exactly one infobox. We learn classification models based on the training examples and apply the models over unlabeled entities. The predicted class labels are suggested infobox template types for the unlabeled entities.

We use three different kinds of features in classification– words in articles (W), categories of articles (C), and named entities in articles (E). More specifically, given an article, W is the set of words in the article's content, C is the set of Wikipedia categories assigned to the article, and E is the set of named entities hyperlinked from the article's content. Below we provide more details about the features.

Words: Stemming was applied on all training and test articles by using the Porter stemmer and stop words removal was performed by using MySQL full-text stop words list. We apply two improvements over the standard bag-of-words model in constructing article features. First, we use the first k sentences instead of all sentences in an article. This is based on the observation that the first paragraph of an article typically provides a summary of the corresponding entity and the first sentence particularly is often a definition such as "... is a ...". Second, we apply TF-IDF weighting on the features, where TF refers to a token's term frequency and IDF refers to its inverse document frequency, i.e., the number of articles containing the token.

Categories: A Wikipedia article (entity) may be associated with one or more categories. These categories are listed at the bottom of the article. For instance, the categories for the entity in Figure 1 are 1979 births, German emigrants to the United States, and so on. (Figure 1 only highlights some of the categories.) In constructing the features of an article, we use not only its immediate categories but also their direct super-categories based on Wikipedia's category hierarchy.

What is worth noting is that although categorization and classification are intrinsically related, the categories in Wikipedia are much more intense, more detailed, and less organized. An entity may have many categories but belong to only one infobox template (type). Some categories may not be relevant to its type and some may even be inaccurate. For instance, Jawed Karim in Figure 1 has a category YouTube, which is not useful for giving him a type. This problem can be particularly common in lengthy articles which may get hundreds of categories if not assigned carefully. This problem is known as overcategorization or "category clutter" [2].

Entities: The article of an entity may also contain a number of other named entities which are related to the entity and hence can be useful features in classification. We only use the entities in the first k sentences, based on the same intuition applied on word features. The general problem of finding named entities in text documents is the well-studied named entity recognition problem. However, the internal hyperlinks in Wikipedia make it straightforward to identify many important named entities in articles. For instance, in Figure 1 the hyperlinks to Wikipedia entities such as East Germany

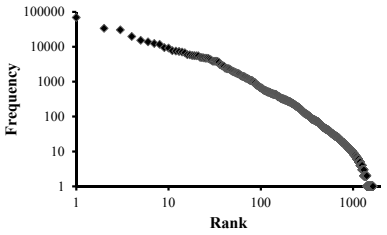

Figure 3: Distribution of articles by infobox template types.

test data	TF (k=1)	TF (k=4)	TF (k=10)	TF (k=∞)
TEST1	80.50%	85.76%	85.70%	85.79%
TEST2	70.5%	70.5%	70.2%	68.3%

Table 1: Micro-averaged F_1 of word-as-feature, varying k.

and Youtube indicate that Jawed Karim is related to these entities.

Voting of the features: To combine the multiple features, we apply a simple voting mechanism. Three classifiers are constructed, by using word-as-feature, category-as-feature, and entity-as-feature. Different from majority-voting, we identify the most effective classifier among the three and follow its vote unless the other two classifiers predict the same class label that disagrees with its vote. Interestingly we find that while category-as-feature produces more accurate results on labeled articles, word-as-feature achieves better accuracy on unlabeled articles.

We employed nonlinear SVM with polynomial kernel in building classifiers. The SVM implementation we used is in Weka [5]. We also applied a naive bayes classifier (NBC). Our results show that SVM consistently outperforms NBC, which is not surprising as SVM has become one of the most effective text classification methods. We do not discuss NBC results due to space limitations.

3. PRELIMINARY EXPERIMENTS

In this section we present the preliminary results of our experiments. In evaluating our method, we used the 2008-07-24 snapshot of English Wikipedia. There are about 1.8 million articles, among which 808,144 articles have infoboxes and the remaining 1 million articles have no infoboxes. There are 1,646 infobox template types in total. Figure 3 shows the distribution of articles by types. The x axis is for ranks of types by frequency and the y axis is for frequencies of types, where the frequency of a type is the number of articles of that type. In this figure, we have only included the 539,468 articles that have exactly one infobox. It is clear that the frequency of infobox template types follows Zipf's law. 336 out of the 1,646 types have at least 100 articles in each type. 515,101 ($\sim$95%) of the 539,468 articles belong to these 336 types.

In our classification task, we considered 337 classes– the 336 most frequent types and OTHERS, which is the combination of all other infrequent types. Our training set had 21,905 articles, consisting of 65 articles for each of the 337 classes. These articles were randomly chosen from the 539,468 articles with exactly one infobox. We used two test sets. The first test set (TEST1) had 3,370 articles– 10 random articles for each class. The second test set (TEST2) had 1,000 articles that were randomly sampled from the 1 million articles without infobox. During the random sampling, we discarded articles that do not describe named entities. Hence the 1,000 test articles are all named entities and thus are reasonable to be assigned infobox template types. Since these 1,000 articles do not have infobox, we manually labeled them to form the corresponding ground truth. We used these 2 test sets due to the aforementioned different characteristics of articles with and without infoboxes. In TEST1, we made the sizes of all classes equal so that we can test on all classes. In TEST2, we did not guarantee that, for capturing realistic distribution of articles in different classes and for coping with overhead in manual labeling.

test data	TF ($k = 4$)	TF-IDF ($k = 4$)
TEST1	85.76%	86.77%
TEST2	70.5%	70.8%

Table 2: Micro-averaged F_1 of word-as-feature, TF vs. TF-IDF, varying k.

test data	L1	L1+L2
TEST1	79.29%	88.22%
TEST2	34.2%	37.2%

Table 3: Micro-averaged F_1 of category-as-feature, L1 vs. L1+L2.

In our following discussion, we present the performance of classifiers constructed by various features and their combinations. All classifiers were tested on both test sets. For each experiment, we report *accuracy*, i.e., the percentage of correctly classified articles. Note that in this case micro-averaged F_1, micro-averaged precision, micro-averaged recall, and accuracy are the same, because we perform *one-of* classification, in which each article is in exactly one class and a classifier assigns exactly one class to each article.

Word-as-feature: Table 1 and 2 show the results of SVM classifiers using words as features. We use TF to denote a classifier if only term frequency is applied and TF-IDF if inverse document frequency is also applied. We tested the performance of TF under different k values, in which the classifier used the first k sentences of an article and discarded the rest. We use k=∞ to represent the case where all sentences are exploited.

On TEST1 (test articles with infoboxes), using the first sentence of an article achieved 80% accuracy and using the first 4 sentences further substantially improved the accuracy to 85.76%. The diminishing return came quickly after the first several sentences, as further enlarging k did not bring clear improvement in accuracy. This verifies the intuition of using only first several sentences.

On TEST2 (test articles without infoboxes), using first sentences already achieved the best accuracy. Furthermore, the accuracy on TEST2 is significantly lower than that on TEST1. Both observations on TEST2 show the differences between the two test sets, as discussed in Section 1. They indicate that articles without infoboxes are naturally shorter and perhaps have lower quality than TEST1. Using all sentences in this case actually downgraded performance.

Table 2 compares the accuracy of TF and TF-IDF, under k=4. We observe that TF-IDF attained marginal improvement on both TEST1 and TEST2.

Category-as-feature: Table 3 shows the results of SVM classifiers using categories as features. We experimented with using immediate categories of articles (L1) and using both immediate categories and their direct super-categories (L1+L2). Since categories do not appear multiple times on an article, we did not consider frequency of features. We did not consider inverse document frequency (of categories) either, since the cardinality of categories is much smaller than that of words.

On TEST1, we observed a substantial accuracy improvement from L1 to L1+L2, indicting the effectiveness of using super-categories. We also note that L1+L2 achieved better accuracy than word-as-feature. This suggests that categories are more reliable than words in predicting classes for TEST1. On the other hand, unlike word-as-feature, category-as-feature performed poorly on

test data	TF-IDF (k=4)	L1+L2	Entity (k=4)	W+C+E (favor W)	W+C+E (favor C)
TEST1	86.77%	88.22%	68.64%	86.80%	92.03%
TEST2	70.8%	37.2%	26.4%	71.7%	37.3%

Table 5: Micro-averaged F_1 of the voting scheme vs. individual features.

test data	Entity (k=1)	Entity (k=4)	Entity (k=∞)
TEST1	61.45%	68.64%	65.28%
TEST2	21%	26.4%	24.8%

Table 4: Micro-averaged F_1 of entity-as-feature, varying k.

TEST2. This can be explained by that articles without infoboxes may not be well-categorized.

Entity-as-feature: Table 4 shows the results of SVM classifiers using entities as features. We observed that, when entities are features, the first 4 sentences are more effective than just the first sentence. This is consistent with the observation made from Table 1. However, using all instead of the first 4 sentences worsened the accuracy on TEST1, which is different from Table 1. This suggests that the relevance of entities in later sentences downgrades more than that of words. We also observed that entity-as-feature performed poorly on TEST2. The gap between TEST1 and TEST2 was about 15% when using words as features and became a much wider 40% under entity-as-feature. This suggests that articles without infoboxes may have fewer and less relevant entities, since they are less mature than articles with infoboxes.

Voting of the features: Table 5 shows the results of the simple voting mechanism, in comparison with individual features. From word/category/entity-as-feature, we chose TF-IDF (k=4), L1+L2, and Entity (k=4), respectively, because they achieved almost the best performance in their own type and used only few features (the first 4 sentences). We used each of these 3 individual classifiers as a voter. We considered 2 different voting schemes in combing these classifiers, represented as W+C+E, i.e., word+category+entity. The one favoring W (word) follows the vote from TF-IDF (k=4) unless the other two classifiers predict the same class label that contradicts with the vote from TF-IDF (k=4). Similarly, the one favoring C (category) follows L1+L2 unless the other two classifiers disagree.

The interesting observation from Table 5 is that these two schemes performed inconsistently on TEST1 and TEST2. While favoring W was more effective on TEST2, favoring C was more effective on TEST1. Since category-as-feature has better performance on TEST1 than word-as-feature and entity-as-feature, favoring C gave us the best accuracy (92.03%) in all experiments. It indicates that W and E together corrected some mistakes made by C. Even though E had worse accuracy than W and C, it helped. On TEST2, favoring W is the better choice since word-as-feature has the best individual performance. The improvement was marginal though, from 70.8% to 71.7%. Since both C and E have poor accuracy on TEST2, they together could not correct many mistakes made by W.

4. CONCLUSION

This paper presents our work in progress towards building a full-fledged tool that assists Wikipedia contributors in authoring articles, particularly for suggesting infobox templates to articles. The preliminary results suggest several directions towards our goal. We will apply our approach over the full set of Wikipedia articles–training on all articles with infoboxes and testing on all articles without infoboxes. Such large scale evaluation would require a parallel framework such as MapReduce. We also plan to apply more principled feature selection in SVM, although our choice of using the first several sentences is a form of rudimentary feature selection. Finally, we will incorporate infobox template suggestion with the automatic infobox completion techniques developed in [14], to deploy a more complete author assistant tool.

5. REFERENCES

[1] Introducing the knowledge graph. `http://www.google.com/insidesearch/features/search/knowledge.html`.

[2] Wikipedia:overcategorization. `http://en.wikipedia.org/wiki/Wikipedia:Overcategorization`.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[4] W. Dakka and S. Cucerzan. Augmenting Wikipedia with named entity tags. In *IJCNLP*, pages 545–552, 2008.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[6] R. Kaptein and J. Kamps. Using links to classify Wikipedia pages. *Advances in Focused Retrieval*, pages 432–435, 2009.

[7] Nadeau, David, Sekine, and Satoshi. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007.

[8] J. Nothman, J. R. Curran, and T. Murphy. Transforming Wikipedia into named entity training data. In *Proceedings of the Australian Language Technology Workshop*, pages 124–132, 2008.

[9] A. E. Richman, P. Schone, and F. G. G. Meade. Mining Wiki resources for multilingual named entity recognition. In *ACL*, pages 1–9, 2008.

[10] I. Saleh, K. Darwish, and A. Fahmy. Classifying wikipedia articles into ne's using svm's with threshold adjustment. In *Proceedings of the 2010 Named Entities Workshop*, NEWS '10, pages 85–92, 2010.

[11] M. Tkatchenko, A. Ulanov, and A. Simanovsky. Classifying wikipedia entities into fine-grained classes. In *2011 IEEE 27th International Conference on Data Engineering Workshops (ICDEW)*, pages 212–217, april 2011.

[12] A. Toral and R. Munoz. A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In *Proceedings of the EACL Workshop on New Text*, pages 56–61, 2006.

[13] Y. Watanabe, M. Asahara, and Y. Matsumoto. A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In *EMNLP-CoNLL*, pages 649–657, 2007.

[14] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *CIKM*, pages 41–50, 2007.

[15] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *WWW*, pages 635–644, 2008.