Wiki3C: Exploiting Wikipedia for Context-aware Concept Categorization

Peng Jiang[†], Huiman Hou[‡], Lijiang Chen[†], Shimin Chen[†], Conglei Yao[§], Chengkai Li[¶], Min Wang[†]
[†]HP Labs China, Beijing, China
[‡]Baidu Inc., Beijing, China
[§]Tencent Inc., Beijing, China
[¶]University of Texas at Arlington, Arlington, TX, USA
[†]{pengj, lijiang.chen, shimin.chen, min.wang6}@hp.com
[‡]humanhou@hotmail.com, [§]ycl.pku@gmail.com, [¶]cli@cse.uta.edu

ABSTRACT
Wikipedia is an important human generated knowledge base containing over 21 million articles organized by millions of categories. In this paper, we exploit Wikipedia for a new task of text mining: Context-aware Concept Categorization. In the task, we focus on categorizing concepts according to their context. We exploit article link feature and category structure in Wikipedia, followed by introducing Wiki3C, an unsupervised and domain independent concept categorization approach based on context. In the approach, we investigate two strategies to select and filter Wikipedia articles for the category representation. Besides, a probabilistic model is employed to compute the semantic relatedness between two concepts in Wikipedia. Experimental evaluation using manually labeled ground truth shows that our proposed Wiki3C can achieve a noticeable improvement over the baselines without considering contextual information.

Categories and Subject Descriptors
I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text analysis.*

General Terms
Algorithms, Experimentation.

Keywords
Context-aware concept categorization, Wikipedia, text mining.

1. INTRODUCTION
Wikipedia is a rich human-generated knowledge base containing over 21 million articles organized into millions of categories. A number of prior studies utilize category information in Wikipedia to enrich the representation of text in various text mining tasks, including text classification [1, 2], text clustering [3, 4], word sense disambiguation [5, 6], information retrieval [7, 8], similarity computing [9, 10], taxonomy building [11], and so on. The main methodology of text enrichment in these works is to map text tokens to Wikipedia concepts and further to Wikipedia categories. However, a concept in Wikipedia usually has many categories. Some categories are irrelevant to the text from which a concept is extracted. Most previous works treat all Wikipedia categories equally without considering their relative importance in different specific contexts.

In this paper, we introduce and study a new task to solve the above problem: context-aware concept categorization by Wikipedia. In this task, we are interested in ranking categories for a concept to determine which categories describe it better with respect to a particular textual context. It leads us to a more fine-grained understanding of concepts in their contexts. The understanding can help enrich unstructured text by its relevant semantic information.

Figure 1 illustrates an example of the task of context-aware concept categorization. There are two paragraphs of text, from which concepts are extracted by an existing tool [12] and are underlined. For instance, both paragraphs in Figure 1 contain the concept "*Iron Man*". Each extracted concept corresponds to a Wikipedia article. The top portion of Figure 1 shows that the Wikipedia article for "*Iron Man*" belongs to 22 Wikipedia categories. [1] These categories bear different importance in different contexts. For instance, consider two particular categories "*Film characters*" and "*Characters created by Stan Lee*" of "*Iron Man*", which are in boldface in Figure 1. Between these two categories, "*Film characters*" is the more relevant one in the context of Paragraph 1, whereas "*Characters created by Stan Lee*" is more relevant with regard to Paragraph 2. Our task is to rank all categories of "*Iron Man*" according to their relevance to the contextual information where the concept is extracted.

Context-aware concept categorization provides a fine-grained understanding of extracted concepts. As shown in the above example, the same concept is captured by quite different categories in different contexts. In addition, context-aware categorization is a form of in-depth understanding of important aspects of concepts in particular textual contexts. Through this task, one can derive context-aware important categories that are at higher level of abstraction than individual concepts.

WSDM'13, Feb 4–8, 2013. Rome, Italy.

[1] Articles in Wikipedia are assigned to multiple categories, which represent major topics of the articles. Each category may contain a number of child articles.

"Iron Man" belongs to 22 categories in Wikipedia:

1968 comic debuts | 1996 comic debuts | 1998 comic debuts | 2005 comic debuts | 2008 comic debuts | 2011 comic debuts | Comics characters introduced in 1963 | Characters created by Don Heck | Characters created by Jack Kirby | **Characters created by Stan Lee** | Comics adapted into films | Fictional business executives | Fictional characters from New York | Fictional cyborgs | Fictional engineers | Fictional inventors | Fictional scientists | Fictional socialites | **Film characters** | Iron Man | Marvel Cinematic Universe characters | Marvel Comics titles

Paragraph 1 that contains "Iron Man":

In The Avengers, Nick Fury, director of the peacekeeping organization S.H.I.E.L.D., recruits **Iron Man**, Captain America, the Hulk and Thor to form a team that must stop Thor's adoptive brother Loki from enslaving the human race.

Category: Film characters, ……

Paragraph 2 that contains "Iron Man":

He co-created Spider-Man, the Hulk, the X-Men, the Fantastic Four, **Iron Man**, Thor, and many other fictional characters, introducing complex, naturalistic characters and a thoroughly shared universe into superhero comic books.

Category: Characters created by Stan Lee, ……

Figure 1. An example of context-aware concept categorization. Different categories should be chosen for the same concept "Iron Man" in different paragraphs.

Previous studies have attained considerable success in exploiting Wikipedia for many text analysis tasks by identifying concepts in text and linking them to Wikipedia articles [12, 13]. The task of context-aware concept categorization proposed in this paper moves one step further. It is worth mentioning that, while Wikipedia provides a gold standard of ground truth for the task of concept extraction (i.e., the hyperlinks in Wikipedia articles to other Wikipedia articles representing such ground truth), it does not provide ground truth for context-aware concept categorization. Therefore, although supervised or semi-supervised techniques can be effectively applied in concept extraction [12, 14, 15], we design an unsupervised and domain-independent approach for concept categorization.

Context-aware concept categorization is related to but different from text categorization (a.k.a. document classification). While the latter converts text into vector, and aims to label it with one or more predefined categories, concept categorization requires a finer-grained analysis to rank categories in the granularity of concept. Moreover, context-aware concept categorization is different from word sense disambiguation task. As shown in Figure 1, "*Iron Man*" in the two paragraphs refers to a same concept without ambiguity. In this work, we focus on compute the different relevance of categories for a given concept according to context. Being lack of description in Wikipedia Category, we represent category for ranking by two kinds of article sets: child article and split article. The set of child article includes Wikipedia articles under a given category, while the set of split article includes articles that are the sub-components of the given category' name. By leveraging the relatedness between each article in the two article sets and concepts in context, we combine the contextual relevance to rank categories for a concept. The relatedness measuring is based on the comparison of link structures[2] in Wikipedia articles. In order to address the bias problem caused by the incompleteness in Wikipedia, we propose a probabilistic model. In the model, any unseen link in a concept can have a probability of occurrence which is proportional to that of the link given by the concept's category.

The main contributions of this paper are two folds. First, to the best of our knowledge, we are the first to introduce and study the problem of context-aware concept categorization. Second, we present an unsupervised and domain-independent approach for this task. Experimental evaluation using manually labeled ground truth shows the effectiveness of our approach.

The remainder of the paper is organized as follows: Section 2 describes our approach to the task of context-aware concept categorization. Evaluation of both accuracy and efficiency of our solution is presented in Section 3. In Section 4, we make a brief discussion of related work. Finally, Section 5 concludes the paper and discusses future work.

Table 1. Symbols used in this paper.

Symbols	Description
a	An article in Wikipedia
$In(a)$	Link set in a
T	Set of concepts extracted from a piece of text
t	A concept in Wikipedia
$r(t_i, t_j)$	Semantic relatedness of concept t_i and t_j
C_i	Set of categories of concept t_i in Wikipedia
c_{ii}	A category of concept t_i
$ch(c)$	Set of child articles in category c
$ch'(c)$	Set of filtered child articles in category c
$sp(c)$	Set of split articles for category c
$R(t, c)$	Relevance of category c to concept t
$T_{context_i}$	Set of contextual concepts of t_i
D	Window size for contextual concepts
μ	Dirichlet smoothing parameter
α	Weight parameter of two category representations
β	Weight Parameter to control the influence of context
K	Pseudo size of $ch'(c)$

[2] Link structure is an important feature of Wikipedia. Underlinked words in a Wikipedia article are typically linked to another relevant Wikipedia page.

2. WIKI3C: CONTEXT-AWARE CONCEPT CATEGORIZATION

In the task of context-aware concept categorization, we are given 1) a set of concepts extracted from an input text, and 2) a list of categories defined in Wikipedia for each concept. We aim at ranking the categories for each concept according to their relevance to the particular textual context surrounding the concept.

In this section, we first present the basic idea of our solution, and then describe each component of our solution in detail. Table 1 summarizes some symbols that are frequently used in this paper.

2.1 Overview of Our Approach

Given a target concept with the surrounding textual context, and the concept's corresponding categories, the task is to rank these categories by exploring the surrounding textual context. For example, in paragraph 1 of Figure 1, the context of the target concept "*Iron Man*" contains several concepts surrounding it, such as "*The Avengers*", "*Nick Fury*", "*S.H.I.E.L.D*", "*Captain America*", etc. Similarly, in Paragraph 2, the context of "*Iron Man*" includes "*fictional characters*", "*naturalistic*", "*shared universe*", etc. The number of contextual concepts is determined by the window size parameter D, provided that the contextual concepts must be in the same paragraph as the target concept.

Figure 2 illustrates the basic idea of our approach. The rectangular box T represents the set of concepts extracted from the input text. $t_i \in T$ is a target concept, represented by "*Iron Man*" in Figure 2. $T_{context_i} \subset T$ is the set of contextual concepts for t_i. In this case, the window size D is set to 2. We denote C_i as the set of categories of t_i. $c_{ij} \in C_i$ is one category of t_i, such as "*Film characters*". In Wikipedia, each category has little description. In order to rank the categories according to their relevance to the context, we investigate two types of Wikipedia article sets to represent a category. The first set is $ch(c_{ij})$ that contains all the child articles of category c_{ij} in Wikipedia. The second set is denoted as $sp(c_{ij})$ with all articles generated directly from the name of category c_{ij}. We will describe the two types of article sets and their usage in the following two subsections.

2.2 Child Article Selection

To represent category c_{ij}, we can select all the child articles $ch(c_{ij})$ in c_{ij}. $ch(c_{ij})$ can be obtained from the category page in Wikipedia. In this way, the relevance between c_{ij} and a contextual concept in $T_{context_i}$ can be measured by the average relatedness between each article in $ch(c_{ij})$ and the concept in $T_{context_i}$.

However, there are biases in Wikipedia. For example, given a concept "*Physics*" in $T_{context_i}$, we consider two categories "*Swiss physicists*" and "*American physicists*". Intuitively the two categories should have a similar relevance to "*Physics*". However category "*Swiss physicists*" is more relevant to "*Physics*" than category "*American physicists*". What happens is that "*Swiss physicists*" has 44 child articles, while "*American physicists*" has 1206 child articles in Wikipedia. Most of the 44 child articles of "*Swiss physicists*" contain long descriptions and have a large number of links with the article corresponding to "*Physics*". Therefore they can achieve higher relatedness with "*Physics*". On the other hand, most of the 1206 child articles of "*American physicists*" describe less famous American physicists, and are very short with few links shared with "*Physics*". Therefore, they are considered as irrelevant to "*Physics*". As such, the average relevance between all articles in category "*American physicists*" and "*Physics*" is significantly biased. It shows that categories that contain few but long articles tend to have higher relevance to a given concept. This seems unfair.

To resolve the above problem, we use K articles with the highest relatedness with other articles in the category to represent the category regardless of the actual number of articles. K is the pseudo size of category. This will remove "unpopular" or "uncompleted" child articles to maintain the size of $ch(c)$ to K. Another benefit of category filtering is that it can shorten the time of relatedness computation. Algorithm 1 selects child articles in $ch(c)$ by the following steps:

1) If the count of links of an article in $ch(c)$ is less than *Threshold*, delete the article.
2) If the size of $ch(c)$ is greater than K, calculate the total relatedness between each article and all the other articles in $ch(c)$. Then delete the article with the smallest total relatedness.
3) Repeat step 2) until the size of $ch(c)$ is equal to K.

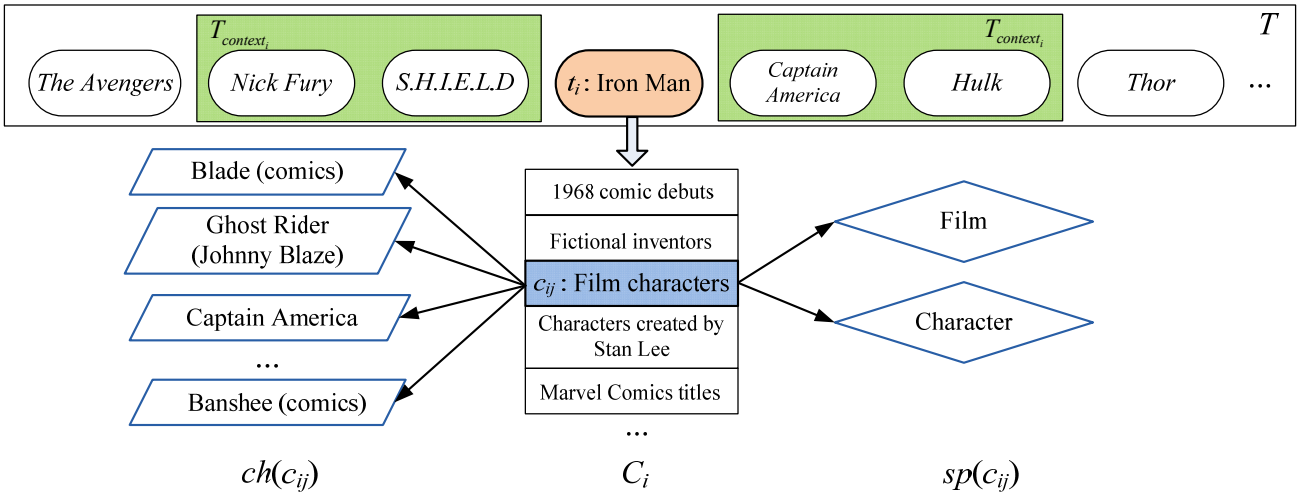


Figure 2. An illustration of the computation of relevance for a category using two article sets.

Algorithm 1: Child articles filtering

Input: $ch(c)$
Output: filtered $ch(c)$
Const Threshold, K;
foreach $a \in ch(c)$ **do**
  if $|In(a)| <$ Threshold **then**
    $ch(c).\text{delete}(a)$;
  end
end
while $|ch(c)| > K$ **do**
  $\min \leftarrow |ch(c)| \times 1.0$;
  foreach $a_i \in ch(c)$ **do**
    $total \leftarrow 0$;
    foreach $a_j \in ch(c)$ **and** $i \neq j$ **do**
      $total \leftarrow total + r(a_i, a_j)$;
    end
    if $total < \min$ **then**
      $\min \leftarrow total$;
      $a_{\min} \leftarrow a_i$;
    end
  end
  $ch(c).\text{delete}(a_{\min})$;
end
return $ch(c)$

Algorithm 1 first deletes noise in a category. The threshold parameter is set to 3 in our experiment. Then Algorithm 1 finds and deletes the article that is most irrelevant to others in a category until the size of $ch(c)$ is equal to K. The relatedness computation will be discussed in Section 2.4. The filtering algorithm does not require any contextual information. Therefore, we perform it offline. We denote the filtered $ch(c)$ as $ch'(c)$ in the rest of the paper.

If a category has less than K articles before filtering, we can use the article with minimum relatedness as supplements. This will be discussed in detail in Section 2.4.3. Figure 2 shows several child articles for category "*Film characters*", including "*Blade (comics)*", "*Ghost Rider (Johnny Blaze)*", "*Captain America*", "*Banshee (comics)*", etc.

2.3 Split Article Selection

Another way to represent a category is to use variants of the category's name. For the example of category "*Swiss physicists*" and "*American physicists*", it is natural to expect the two categories to have similar relevance to the concept "*Physics*" since their names contain a common concept "*physicists*". Another example is "*American physicists*" and "*American chemists*" with the common concept "*American*" in their names. Therefore it is natural to expect them to have similar relevance when their context contains concept such as "*America*". To support this intuition, we split the name of a category into many sub-components if they are valid Wikipedia articles. Compared to articles in $ch(c_{ij})$, the concepts corresponding to split articles are more general. We denote the set of split articles for category c_{ij} as $sp(c_{ij})$ and expect it to serve as a complement for $ch(c_{ij})$.

To measure the relevance between concept t and category c_{ij}, we compute the relatedness between each article in $sp(c_{ij})$ and t. Unlike relatedness computation using child articles in Section 2.4.3, we select the maximum relatedness rather than average relatedness. That means we select the split article with the maximum relatedness with contextual concept to represent c_{ij}. It is because the relevance of each member in $sp(c_{ij})$ may vary greatly. Therefore, it is not appropriate to use all split articles to represent c and compute the average relatedness as the relevance of c_{ij}. For example, "*America*" is a contextual concept and "*American physicists*" is a category for a target concept. The relatedness between the concept "*America*" and a split article "*American*" of category "*American physicists*" is 1, while the relatedness between "*America*" and the other split article "*Physicist*" will be much smaller. Intuitively, it is better to use "*America*" to represent category "*American physicists*" in this context. Computing maximum relatedness matches this intuition. Figure 2 shows two split articles "*Films*" and "*Character*" for category "*Film Characters*".

Now we have two approaches to representing a category in Wikipedia. In the next section, we present our approach to ranking categories using these two representations.

2.4 Category Ranking

Before presenting the category ranking function, we first introduce the computation of semantic relatedness between concepts. We use link structure to represent concepts in our approach. There are two kinds of links in Wikipedia: inlink (incoming link) is the article with links toward the target article while outlink (outgoing link) is the article referred by the target article. Inlink and outlinks behave similar and both are good indicators of relevance [16].

2.4.1 Basic Model

In this model, we represent concept t by an link set $In(a)$, where a is the corresponding article of t in Wikipedia. According to the representation, we can compute the semantic relatedness between two concepts t_i and t_j by the following formula:

$$r(t_i, t_j) = \frac{|In(a_i) \cap In(a_j)|}{|In(a_i) \cup In(a_j)|} \quad (1)$$

where a_i and a_j are the corresponding Wikipedia articles for concept t_i and t_j respectively. The above formula employs the link structure in Wikipedia to measure the relatedness between two articles. It assumes that the more links shared by the two concepts, the more related they are. In addition, the number of link shared is normalized by the size of $In(a_i) \cup In(a_j)$. This model is simple and intuitive, but has the following weakness. In some cases, two articles in the same category have no shared links. The relatedness between them would be zero, even though they belong to the same category. This is because Wikipedia is human generated knowledge base and some articles corresponding to "unpopular" concepts may be quite short and incomplete. In order to solve this problem, we introduce the following probabilistic model.

2.4.2 Probabilistic Model

In the probabilistic model, we represent a concept t as a probability distribution over links. Different from the basic model, we assume an unseen link (outlink) in t to have a probability of occurrence. The probability is proportional to that of the link given by all categories that t belongs to. The probabilistic model θ_t for t can be estimated as follows:

$$p(link \mid \theta_t) = \frac{n(link; t) + \mu p(link \mid C)}{|t| + \mu} \quad (2)$$

where $n(link; t)$ is the number of times *link* appears in the article corresponding to *t*. $|t|$ is the number of links in *t*. μ is a Dirichlet parameter and will be determined in experiment. *C* is the set categories that *t* belongs to. $p(link|C)$ can be estimated as follows:

$$p(link|C) = \frac{\sum_{c \in C} \sum_{a \in c} |n(link; a)|}{\sum_{c \in C} \sum_{a \in c} |a|} \quad (3)$$

where *c* is a category of *t* in *C*. *a* is an article belongs to *c*. $|a|$ is the number of links in *a*. According to Formula 2, each concept in *c* shares all links of *c* with the probabilistic related to the frequency of the link occurring in *c*. The semantic relatedness between two concepts t_i and t_j can be measured as follows:

$$r(t_i, t_j) = -D(\theta_i \| \theta_j) - D(\theta_j \| \theta_i) \quad (4)$$

where $D(\theta_i \| \theta_j)$ is the KL-divergence of θ_i and θ_j:

$$D(\theta_i \| \theta_j) = \sum_{link} p(link|\theta_i) \log \frac{p(link|\theta_i)}{p(link|\theta_j)} \quad (5)$$

The more t_i relates to t_j, the smaller $D(\theta_i \| \theta_j)$ is. If t_i and t_j are the same concept, $D(\theta_i \| \theta_j)$ equals 0. Therefore, we use the negative KL-divergence to measure the relatedness.

2.4.3 Category Ranking Approach

Based on the models introduced in the above section, the relevance between a category *c* and a concept *t* can be computed as follows:

$$\begin{aligned} R(t,c) &= \alpha R(t, ch'(c)) + (1-\alpha) R(t, sp(c)) \\ &= \alpha \frac{1}{K} \sum_{t_i \in ch'(c)} r(t,t_i) + (1-\alpha) \max_{t_i \in sp(c)} r(t,t_i) \end{aligned} \quad (6)$$

where $R(t, ch'(c))$ is the relatedness between *t* and filtered child articles $ch'(c)$. $R(t, sp(c))$ is the relatedness between *t* and split articles $sp(c)$. The reason to use the maximum relatedness instead of average relatedness can be found in Section 2.3. α in Formula (6) is a parameter used to control the influence weight of two category representations. *K* is the pseudo size of each category. If the size of $ch'(c)$ is less than *K*, we can use the concept with minimum relatedness with *t* as supplement:

$$t_{\min} = \arg\min_{t_i} r(t,t_i) \quad (7)$$

Thus $R(t, ch'(c))$ in Formula (6) can be rewritten as:

$$R(t, ch'(c)) = \frac{1}{K} \left(\sum_{i=1}^{n'} r(t,t_i) + (K-n') r(t,t_{\min}) \right) \quad (8)$$

where n' is actual size of $ch'(c)$. We use the article in $ch'(c)$ with the minimum relatedness as supplement and keep the size of each category to *K*, so that each child article has the same contribution to the relevance of different-sized categories. For example, if in category A, the relatedness of two articles is 0.8 and 0.2 respectively; whereas the relatedness of three articles in category *B* is 0.8, 0.3 and 0.3 respectively. Obviously, *B* should be more relevant than *A*. But if we use the actual size to average the impact of each article, category *A* would rank higher than category *B*.

Given *n* concepts $T = \{t_1, t_2, \ldots, t_n\}$ extracted from a paragraph, we use $T_{Context_i} = \{t_{i-D}, \ldots, t_{i-1}, t_{i+1}, \ldots, t_{i+D}\}$ to represent all contextual concepts for a target concept t_i. *D* is the contextual window size which determines how many concepts surrounding t_i will be considered as contextual concepts. Suppose there are *m* predefined categories $C_i = \{c_{i1}, c_{i2}, \ldots, c_{ij}, \ldots, c_{im}\}$ that t_i belongs to. The scoring function for category is defined as follows:

$$score(t_i, c_{ij}) \overset{\text{rank}}{=} \frac{\beta}{|T_{context_i}|} \sum_{t' \in T_{context_i}} R(t', c_{ij}) + (1-\beta) R(t_i, c_{ij}) \quad (9)$$

where $R(t', c_{ij})$ is the relevance between a contextual concept t' and a category c_{ij} of the target concept t_i. On the other hand, $R(t_i, c_{ij})$ is the relevance between the target concept t_i and its category c_{ij} without considering the context. We observe that the relevance values of a target concept to its own categories may vary. For example, for the concept "*Iron Man*", its category "*Marvel Comics title*" is more relevant than its category "*Fictional socialites*" without considering the context. Formula (9) takes into account the impact of the relevance without the context in category ranking. $R(t_i, c_{ij})$ and $R(t', c_{ij})$ can be computed by Formula (6). β is a parameter used to control the influence weight of context. If $\beta = 0$, the approach will be simplified to use only the information of the target concept without using the context of the target concept. Finally, we compute the ranking score for every category c_{ij} in C_i, and then rank the categories in descending order of the score.

3. EXPERIMENTS

This paper aims at exploiting Wikipedia for context-aware concept categorization. In line with this, we select two baselines in our experiments for comparative studies: (1) Ranking randomly for categories (Random). (2) Approach only using target concept but not the context surrounding it (Without context). Note that baseline 1 only utilize the basic knowledge structures in Wikipedia, and are just used to prove the intention of concept categorization. Baseline 2 is used to evaluate the contextual impact on the task of concept categorization.

We perform experiments for the two strategies of article selection discussed in Section 2.2 and 2.3, and compare our basic model and probabilistic model mentioned in Section 2.4.

3.1 Data Set and Evaluation

Currently, there is no available dataset in the community for the evaluation of concept categorization task. Therefore, we manually label a test set as the ground truth. We randomly select 100 English articles from Wikipedia for fine-grained labeling. The concepts in the articles have already been labeled by Wikipedia, and are used as given knowledge in our labeling process of the experiment. Our objective is to select relevant categories among all categories for each target concept according to the context.

We develop a category labeling tool. In our tool, each evaluation user can browse each article in Wikipedia with concepts marked in a different color from the main text. The reason to select Wikipedia article is that all concepts are linked unambiguously by editors. He/she can click on each concept and get all candidate categories of the clicked concept. Evaluation users can select the relevant categories according to textual context. We merge their labeled result together to obtain a final ground truth: only if it is checked by more than one user, a category is selected for the ground truth. In this way, the ground truth enables us to know whether or not a category is relevant to the concept given its context. Finally, we labeled 3072 concepts that belong to 29044 categories (7780 relevant categories). We notice that each concept has an average of 9.5 categories, of which only 2.5 are relevant according to the ground truth.

In our experiments, Wiki3C ranks all the categories for a concept according to its context and compares the rank results with the

ground truth. We use MAP (Mean average precision), R-precision (R-prec) and bpref as evaluation metrics [17] to measure the quality of category ranking.

3.2 Overview of Experimental Results

Table 2 shows the performance comparison among all the approaches. Overall, all context-aware approaches outperform the three baselines. It proves that category selection is related to context, which is the basic assumption for the task of context-aware concept categorization in this paper. Moreover, baseline 1 only uses basic category structure information without any computation of relatedness. Their performance is much worse than that of baseline 2 uses the relatedness between the target concept and its categories. This proves the Wikipedia categories are not equal for a given concept.

In the context-aware approaches, we compare the performances of basic model and probabilistic model with different article selection strategies. Overall, the probabilistic model outperforms basic model. In addition, using filtering algorithm to select top K child articles can improve the performance significantly. We will discuss about it in Section 3.3.3. As shown in Figure 4, we can see that the curve of probabilistic model using filtered child articles is the closest to the upper right-hand corner of the graph, and achieves the best performance (MAP = 0.7542) with an improvement of 15.2% over baseline 2. This indicates that probabilistic model using filtered child articles is superior to the others. Moreover, child article selection strategy outperforms split article selection in both models. Furthermore, using the filtering algorithm in child article selection demonstrates a further improvement (4.3% for basic model and 2.0% for probabilistic model). The performance of the combination between the two selection strategies will be discussed in Section 3.3.2.

Table 2. Performance comparison.

Approach	MAP	R-prec	bpref
Random (baseline 1)	0.5245	0.4304	0.4213
Without context (baseline 2)	0.6546	0.5504	0.5389
Basic model (child)	0.6979	0.5942	0.5868
Basic model (child + filter)	0.7279	0.6142	0.6088
Basic model (split)	0.6679	0.5613	0.5414
Probabilistic model (child)	0.7392	0.6382	0.6321
Probabilistic model (child + filter)	**0.7542**	**0.6512**	**0.6411**
Probabilistic model (split)	0.6942	0.5932	0.5845

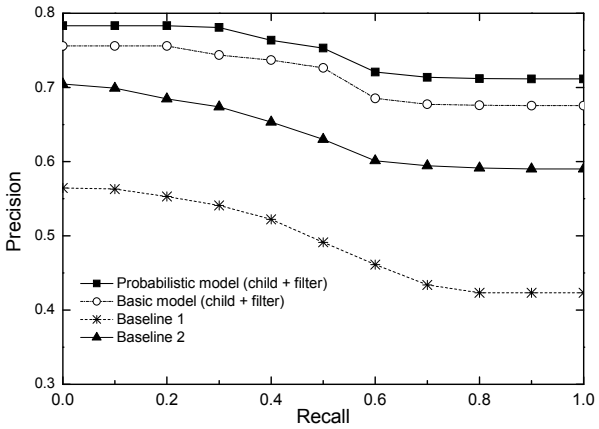


Figure 3. Comparison of recall-precision curves among different approaches.

3.3 Analysis of Parameters

3.3.1 Parameter D

In our approach, we use the concepts surrounding a target concept as the context. The concepts in the same paragraph with the target concept are considered as contexts and their number is determined by the window size parameter D. We experiment with D varying from 1 to 10 to see its impact on the quality of category ranking. In Figure 4, we notice that the performances of all approaches improve gradually as D increases (MAP of the approach without using context is constant at 0.6546). But when D is approaching 10, the performance deteriorates slightly. An explanation is that concepts separated by a significant distance are also less relevant than concepts located closer to each other. If D is set too large, some unrelated and noise contextual concepts will be included.

In particular, for the approaches that only use split articles, their performance does not change much when D varies, since the number of split articles for a category is often small (less than 5), and some of them are context-independent. However, compared to the baseline approach, using contextual information improves MAP from 0.6546 to 0.6942 (6.0% increase for probabilistic model) and 0.6669 (1.9% increase for basic model) respectively. For the approaches that using child articles, the performance increases significantly as D increases. It reaches the best performance when D = 8, after which the performance stays roughly the same. Therefore, D is fixed to 8 in other experiments.

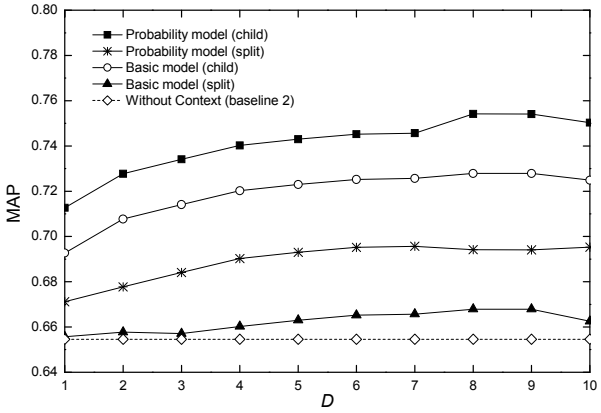


Figure 4. Performance sensitivity to window size *D*.

3.3.2 Parameter α and β

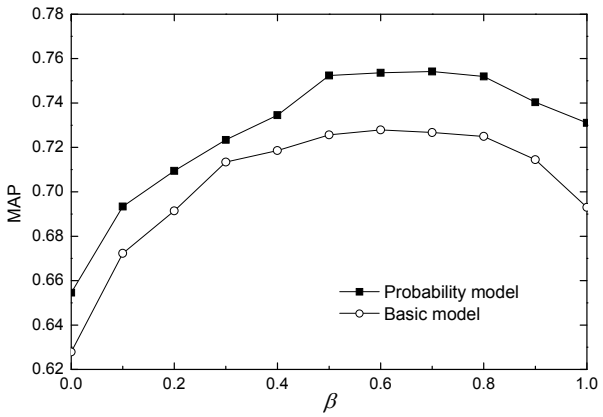
In our approach, there are two main parameters that should be determined in our experiments: α and β. The parameter β in Formula (9) controls influence of context for category ranking. Specifically, β is used to adjust the ratio of target concept's relevance and contextual relevance in the ranking model. Figure 5(a) shows the effect of varying β from 0 to 1, with a step up size of 0.1. In this experiment, we set $\alpha = 1$, thus, we actually only use the child articles to represent a category. Clearly, the introduction of contextual concepts can significantly improve the performance. But when β is approaching 1, the performance deteriorates sharply, though it is still better than without using contextual concepts. For the basic approach, the performance keeps improving until $\beta = 0.6$, while the best performance occurs when β = 0.7 for the probabilistic model.

As mentioned in Section 2.2 and 2.3, both the split articles and child articles can represent a category. We combine them together to calculate the ranking score for a category. In Formula (6), α is a weight parameter to control the proportion of child articles in the linear combination. Figure 5(b) shows how MAP varies accordingly with α when β is fixed at 0.7. The best performances of probabilistic model and basic model are reached at α = 1.0. Clearly, the combination does not achieve better performance than using child articles alone. In the solution of split article selection, we expect to find a generalized concept that can best represent the category. This leads to categories with similar name having a similar relevance and some feature will be lost in these categories. For example, 7 categories of the concept "*Iron Man*" contain a common split article "*Character* (*film*)". Therefore, they may have a similar relevance to the context in Paragraph 1 of Figure 1. But in fact, the correct category "*Film characters*" should be more relevant than others such as "*Comics characters introduced in 1963*", "*Characters created by Stan Lee*", etc. In conclusion, child articles are more effective than split articles to represent categories in the task of context-aware concept categorization.

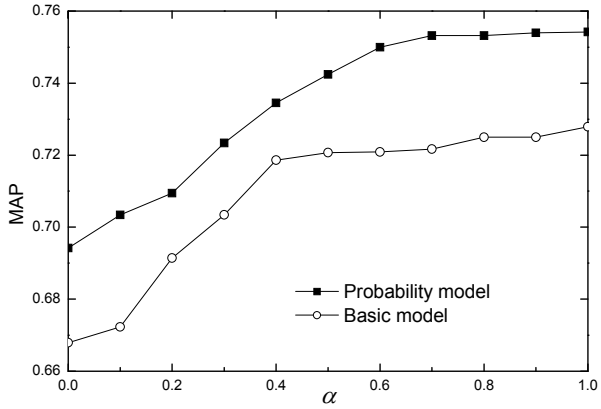
3.3.3 Parameter K and μ

Another two key parameters that should be determined in our approach are K and μ. K is used for the child article selection in Section 2.2. It can be considered as a pseudo category size to filter noise and irrelevant articles in a category. Figure 6(a) shows that the performance increases significantly as K increases. However, when K exceeds 10, the performance deteriorates gradually. When K is large enough, we actually select all child articles to represent a category without filtering. In the other experiments, we set K = 8, which is optimal for all models in this experiment.

According to Formula (2), μ is the Dirichlet prior for smoothing. Because some articles corresponding to "unpopular" concepts in Wikipedia may be very short and incomplete, their relevance always tends to be zero even though they belong to the same category. The introduction of μ makes unseen link of an article have a probability of occurrence. The probability is proportional to that of the link given by all categories that the concept belongs to. Thus, each concept shares some links that belong to the common categories. Figure 6(b) demonstrates the changing performance by changing μ from 0 to 4000. When μ increases, the performances of the two strategies of article selection improve until μ = 1000. After that, the performances deteriorate slightly and start to stabilize. It is worth to mention that when μ is too large, each concept tends to have a similar distribution as that of the categories to which it belongs. Therefore, each concept loses its own features. Specifically, the performance of selecting child articles to represent categories can be improved significantly by the introduction of μ; whereas the impact of μ is relatively small for the approach using split article selection. This is because we actually use only one article to represent a category. The selected article has the maximum relatedness among all split articles for the category, and is relatively long and complete. Besides, split articles usually correspond to general concepts. So the bias mentioned above is small.

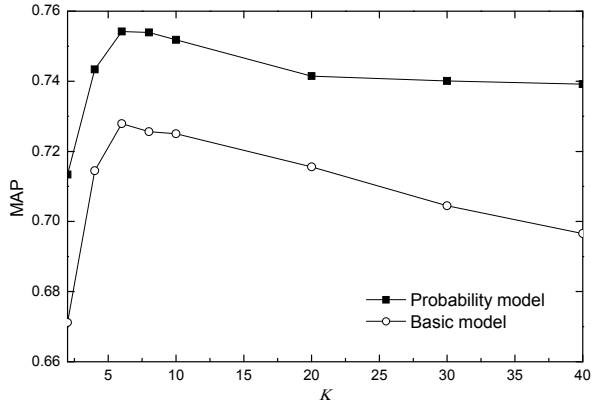


(a) MAP sensitivity to β

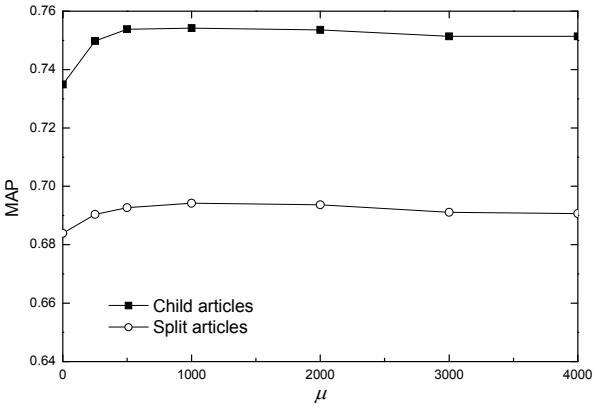


(b) MAP sensitivity to α

Figure 5. Performance sensitivity to α and β.



(a) MAP sensitivity to K



(b) MAP sensitivity to μ

Figure 6. Performance sensitivity to μ and K.

4. RELATED WORK

The work described in this paper mainly uses the knowledge in Wikipedia. Previous work using Wikipedia mainly focuses on semantic relatedness computing, word sense disambiguation, text classification and clustering. Völkel et al. [18] provide an extension to be integrated in Wikipedia that allows the typing of links between articles and the specification of typed data inside the articles to be easy-to-use. Strube and Ponzetto [19] are the first to propose a Wikipedia based approach to computing measures of semantic relatedness. Gabrilovich and Markovitch [2] enrich document representation through Wikipedia to improve the performance of text categorization. Empirical results prove that this knowledge-intensive representation brings text categorization to a qualitatively new level of performance across a diverse collection of datasets. They [10] also propose explicit semantic analysis to measure semantic relatedness. With their approach, they employ text classification techniques to explicitly represent the meaning of any text in terms of Wikipedia-based concepts. In addition, due to the use of Wikipedia concepts, their model is easy to be explained to human users. Cucerzan [20] uses Wikipedia structured knowledge for named entity disambiguation task. Through a process of maximizing the agreement between the contextual information extracted from Wikipedia and the context of a document, as well as the agreement among the category tags associated with the candidate entities, the implemented system shows high disambiguation accuracy on both news stories and Wikipedia articles. Banerjee et al.[21] propose a method of improving the accuracy of clustering short texts by enriching their representation with additional features from Wikipedia. Compared to traditional bag of words representation, the Wikipedia based enriched representation can improve the clustering accuracy significantly.

Within the community of information retrieval (IR) and information extraction (IE), there are various studies related to Wikipedia [14, 15, 22-25]. In 2007, INEX [26, 27] introduces an entity ranking track that aims to evaluate the entity retrieval in Wikipedia. In TREC entity track [28] which aims at finding related entities on the Web, many approaches use Wikipedia as an external resource for query expansion, named entity recognition or entity type detection. Wu and Weld [14] propose an open IE system which performs a self-supervised learning to distill relations from natural-language text. Their system constructs training data by heuristically matching Wikipedia infobox attribute values with corresponding sentences.

The following work is particularly relevant to this paper. Milne and Witten [29] use Wikipedia Link-based Measure (WLM) to compute the semantic relatedness between two articles. Unlike other techniques based on Wikipedia, WLM is able to provide accurate measures efficiently, using only the links between articles rather than their textual content. They [12] also use link to cross-refer documents with Wikipedia. Mihalcea and Csomai [13] propose Wikify, the first system to extract concepts from text documents and link them to Wikipedia articles. Their system extracts keyword automatically and disambiguates word sense by linking concepts extracted from document to the corresponding Wikipedia pages. Our work extends the above studies by ranking the Wikipedia categories of the extracted concepts according to their context, and it can be used in text classification, text cluster, IR and IE.

5. CONCLUSION

Wikipedia produces rich category information that can be used in many text mining tasks, such as text classification, text clustering, word sense disambiguation, etc. However, Wikipedia category representation for each concept is simplistic: a concept contains a number of categories, without considering their relevance to the context where the concept appears. In this paper, we introduce and study the task of ranking categories based on context. We aim at categorizing extracted concepts based on contextual information. It provides an in-depth understanding of concepts in a specific context and generates context-aware categories that are of higher level than concepts. It goes one step further than classical text mining task in Wikipedia, such as semantic relatedness computing, word sense disambiguation, concept extraction, etc.

This paper proposes an unsupervised learning solution to the task of context-aware concept categorization, named Wiki3C. In the solution, we treat the extracted concepts surrounding a target concept as context, followed by ranking the categories of target concept according to the relevance to the context. Two strategies of article selection are chosen to represent category. In addition, we use a probabilistic model to compute the semantic relatedness between concepts. Experimental results prove the effectiveness of Wiki3C. It is worth mentioning that there is still huge potential for further research to improve the performance for context-aware concept categorization. Another interesting future research issue is to apply the results of context-aware concept categorization to other related task, such as text classification, text clustering, topic modeling, contextual online advertising, etc.

6. ACKNOWLEDGMENTS

We would like to express our gratitude to the anonymous reviewers for their insightful feedbacks. The author Chengkai Li is partially supported by HP Labs Innovation Research Program award.

7. REFERENCES

[1] X.-H. Phan, et al. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, Beijing, China, pages 91-100, 2008.

[2] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, Boston, Massachusetts, pages 1301-1306, 2006.

[3] X. Hu, et al. Exploiting Wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Paris, France, pages 389-396, 2009.

[4] D. Carmel, et al. Enhancing cluster labeling using wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, Boston, MA, USA, pages 139-146, 2009.

[5] S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, pages 708-716, 2007.

[6] R. Bunescu and M. Pasc. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for*

Computational Linguistics (EACL), Trento, Italy, pages 2006.

[7] J. Pehcevski, et al. Entity ranking in Wikipedia: utilising categories, links and topic difficulty prediction. *Inf. Retr.*, 13(5):568-600, 2010.

[8] Y. Li, et al. Improving weak ad-hoc queries using wikipedia as external corpus. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, The Netherlands, pages 797-798, 2007.

[9] G. Katz, et al. Using Wikipedia to boost collaborative filtering techniques. In *Proceedings of the fifth ACM conference on Recommender systems*, Chicago, Illinois, USA, pages 285-288, 2011.

[10] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, Hyderabad, India, pages 1606-1611, 2007.

[11] S. P. Ponzetto and M. Strube. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, Vancouver, British Columbia, Canada, pages 1440-1445, 2007.

[12] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, Napa Valley, California, USA, pages 509-518, 2008.

[13] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management,* Lisbon, Portugal, pages 233-242, 2007.

[14] F. Wu and D. S. Weld. Open Information Extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118-127, 2010.

[15] E. Hovy, et al. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, Singapore, pages 948-957, 2009.

[16] J. Kamps and M. Koolen. Is Wikipedia link structure different? In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, Barcelona, Spain, pages 232-241, 2009.

[17] B. Chris and M. V. Ellen. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, Sheffield, United Kingdom, pages 25-32, 2004.

[18] M. Völkel, et al. Semantic Wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, Scotland, pages 585-594, 2006.

[19] M. Strube and S. P. Ponzetto. WikiRelate! computing semantic relatedness using wikipedia. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, Boston, Massachusetts, pages 1419-1424, 2006.

[20] S. Cucerzan. Large Scale Named Entity Disambiguation Based on Wikipedia Data. In *The EMNLP-CoNLL Joint Conference*, Prague, 2007.

[21] S. Banerjee, et al. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, The Netherlands, pages 787-788, 2007.

[22] D. N. Milne, et al. A knowledge-based search engine powered by wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, Lisbon, Portugal, pages 445-454, 2007.

[23] D. P. T. Nguyen, et al. Relation extraction from wikipedia using subtree mining. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, Vancouver, British Columbia, Canada, pages 1414-1420, 2007.

[24] K. Balog, et al. Entity search: building bridges between two worlds. In *Proceedings of the 3rd International Semantic Search Workshop*, Raleigh, North Carolina, pages 1-5, 2010.

[25] Y. Yan, et al. Unsupervised relation extraction by mining Wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, Suntec, Singapore, pages 1021-1029, 2009.

[26] G. Demartini, et al. Overview of the INEX 2009 entity ranking track. In *INEX 2009*, pages 256–264, 2009.

[27] P. V. Arjen, et al. Overview of the INEX 2007 Entity Ranking Track. In *Focused Access to XML Documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 245-251, 2008.

[28] B. Krisztian, et al. Overview of the TREC 2009 Entity Track. In *Proceedings of TREC-2009*, Gaithersburg, USA, 2009.

[29] D. Milne and I. H. Witten. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25-30, 2008.