Semantic Aspect Retrieval for Encyclopedia

Chao Han, Yicheng Liu, Yu Hao, and Xiaoyan Zhu

***Abstract*—With the development of Web 2.0, more and more people contribute their knowledge to the Internet. Many general and domain-specific online encyclopedia resources become available, and they are valuable for many Natural Language Processing (NLP) applications, such as summarization and question-answering. We propose a novel encyclopedia-specific method to retrieve passages which are semantically related to a short query (usually comprises of only one word/phrase) from a given article in the encyclopedia. The method captures the expression word features and categorical word features in the surrounding snippets of the aspect words by setting up massive hybrid language models. These local models outperform the global models such as LSA and ESA in our task.**

***Index terms*—Aspect retrieval, online encyclopedia, semantic relatedness.**

I. Introduction

With the development of Web 2.0, more and more people contribute their knowledge to the Internet. Many general and domain-specific online encyclopedia resources become available, such as Wikipedia[1] and Baidu Baike[2] (the largest Chinese online encyclopedia website). They are well-organized by the categories and interrelations of their entries, meanwhile their content has relatively higher quality than general web pages. So these resources are valuable for many Natural Language Processing (NLP) applications, such as summarization and Question-Answering (QA).

In this paper, we only focus on a specific task: given a "entity-aspect" pair as query, we retrieve passages semantically related to the aspect word from the article corresponding to the entity in the encyclopedia. In the input, the "entity" must be a title of certain article in the encyclopedia; the "aspect" describes some attribute or sub-topic of the entity, and usually comprises of only one word or phrase. For example, for the entity-aspect pair "apple-nutrient", we retrieve the passages which describe apple's nutrient from the "apple" article in the encyclopedia.

The motivations of this task are as follows.

First, this task is an important approach for automatically answering complex natural language questions. A considerable proportion of questions can be converted into a simple description by an entity-aspect pair, as shown in Table I. We can answer this kind of questions directly by giving user the passages related to the aspect from the encyclopedia article corresponding to the entity.

Manuscript received November 1, 2010. Manuscript accepted for publication December 21, 2010.

The authors are with the Department of Computer Science and Technology, Tsinghua University, China (e-mail: hanc04@gmail.com).

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

Second, we choose passage as the unit we retrieve because passage retrieval is a very practical way to supply useful information to users in question-answering and information retrieval field. Usually, for a question answering system, returning users the exact answer is not the best choice [1], users would like to see some surrounding text to make sure that the answer is credible.

Third, because of the higher quality of online encyclopedia, the passages we retrieve can be used in some subtasks such as answer quality validation and so on.

TABLE I
CONVERSION FROM QUESTIONS TO ENTITY-ASPECT PAIRS

Question	Entity–Aspect Pair
What is the nutrient of apples?	apple - nutrient
How about the climate of China?	China - climate
What is a tiger like?	tiger - appearance
What causes diabetes mellitus?	diabetes mellitus - pathogenesis

Besides, to retrieve the passages from a given article of the encyclopedia is an interesting and useful task. Imagine this scenario: a mobile Internet user, who wants to know the nutrient of apple from Wikipedia, would scroll the small-size screen over and over to get what he wants if the search engine gives the whole article to her. It benefits users a lot if the system locates the screen at a more accurate position.

The difficulty lies on how to measure the semantic relatedness between the aspect word and the candidate passages. Simple method based on term vector space model does not work for two reasons: 1) The aspect word may not appear in the article. For example, in the "apple" article in Baidu Baike, the passage about "nutrient" is written as "*Apple contains a lot of pectin, which is a kind of soluble fiber, can make the content of cholesterol and bad cholesterol…*"[3], without using the word "nutrient" directly. 2) Even if the aspect word appears in some passage, the content of the passage may not be related to the aspect, either.

Considering those matters above, the method we adopt should satisfy at least three requirements as follows.

1) The method can handle arbitrary queries, and can measure semantic relatedness between short query and relatively long text.

2) The method should be unsupervised. Because the encyclopedia corpus is large, and it is hard to obtain large enough training set.

[2]http://baike.baidu.com.

[3]All texts from Baidu Baike are originally written in Chinese. We translate them into English in this paper.

3) The computing complexity of our method should be low because of the demand of fast response speed in online applications.

Besides, to certain extent, the method should have the ability of "rejection" when it is not quite confident for the answer.

In this paper, we propose a novel encyclopedia-specific method, which satisfies the requirements above, to retrieve passages for a given "entity-aspect" query. The method exploits features from category information and surrounding snippets. We compare the method with traditional semantic methods such as LSA [9] and ESA [10].

The remainder of the paper is organized as follows. In section 2 related work is discussed. In section 3 we present our approach. Section 4 is the experimental result. Finally, in section 5 we will conclude the paper.

II. Related Work

Nowadays, online encyclopedia websites assemble vast quantities of human knowledge. Consulting an online encyclopedia has become an importance approach for users to achieve the information they need.

Researchers have made great efforts to make it easy to utilize the online encyclopedia resource, especially Wikipedia. Ye et al. [2] proposed to summarize Wikipedia articles as definitions with various lengths to satisfy different user needs. Li et al. [3] proposed Facetedpedia, supplying users a faceted interface for navigating the result articles. The work of Hahn et al. [4] facilitates infobox data allowing users to query Wikipedia like a structured database through an attribute-value pairs extraction approach.

To the best of our knowledge, there is no previous work exactly on the task discussed in this paper. The key point of our task is to measure the semantic relatedness between the aspect word and candidate passages.

A lot of work has been done to quantify semantic relatedness of texts.

The work in [5] treats texts as bags of words and computes similarity in vector space. Lexical resources, such as WordNet, are used in [6] [7] [8].

Latent Semantic Analysis (LSA) [9] uses the Singular Value Decomposition (SVD) to analyze the statistical relationships among terms in a document collection. At first a matrix X with row vectors representing terms and column vectors representing documents, is constructed from the corpus. The cells of X represent the weights of terms in the corresponding documents. The weights are typically TF-IDF. Then SVD, which can be viewed as a form of principal components analysis, is applied to X, and the dimension is reduced by removing the smallest singular values. LSA measures the similarity of terms using the compressed matrix after dimension reduction, instead of the original matrix. The similarity of two terms is measured by the cosine similarity between their corresponding row vectors.

Explicit Semantic Analysis (ESA) [10] is a method based on concepts of Wikipedia or other corpus. ESA maps a text to a high-dimensional vector space with the value of each dimension representing the strength on some explicit concept, for example, an explicit concept can be a concept in Wikipedia. Then we can obtain the similarity between two texts by some measure such as the cosine value between the two corresponding vectors.

III. Our Approach

In our task, the main difficulty comes from the fact that the aspect query is too short – one word in most cases. So the first step of our approach seeks to expand the representation capacity of the aspect query.

TABLE II
SOME SURROUNDING SNIPPETS OF "NUTRIENT" IN ARTICLES OF CATEGORY FRUIT

1. Kiwi is rich in *vitamins* C, A, E in addition to *potassium*, *magnesium*, *cellulose*, but also contains other rare fruit **nutrient**s - *folic acid*, *carotene*, *calcium*, *progesterone*, *amino acids*, *natural inositol*. According to the analysis, every *100 grams* of Kiwi pulp will contain *100* to *300 milligrams* of *vitamin* C, 20 to 80 times higher than apple.
2. Lemon contains *citric acid*, *malic acid* and other *organic acids* and *hesperidin*, *naringin*, *Saint grass sub-glycosides* and other *glycosides*, also contains *vitamin* C, B1, B2 and *niacin*, *carbohydrates*, *calcium*, *phosphorus*, *iron* and other **nutrients**.
3. Citrus fruit is juicy and delicious, rich in *sugars*, *organic acids*, *minerals*, and *vitamins* and other **nutrients**. Its nutritional value is very high.

In Baidu Baike, each article is assigned to at least one category by its editors, and under each category, there are variant number of articles. For example, "pear" belongs to four categories: "fruit", "plant", "foodstuff" and "crops"; in category "fruit", there are about 1800 articles, such as "apple", "pear", "watermelon" and so on.

According to the characteristic of the way the encyclopedia articles are written and organized, we think that for an aspect word, the surrounding snippets of its occurrences contain the information we need to enrich the query. As shown in Table II, surrounding snippets of "nutrient" in articles of category "fruit" have some features in common.

We pick up two types of features.

The first type is **expression word feature**. In Table II, the underlined words, such as "contain", "pulp", "rich" and "high", are frequently used to help expressing the meaning of the content for the aspect.

The other type is **categorical word feature**. The italic words in Table II, such as "potassium", "iron" and "calcium", "citric acid" and "amino acids", are entities in the encyclopedia, and their category information is useful. For example, "potassium", "iron" and "calcium" may be used for the description of nutrients of different kinds of fruit, but they are all "chemical elements". The difference between these words and the expression words of the first type is that not

only the words themselves but also their categories are important: we use some chemical elements to describe the nutrients of certain fruit, whatever the chemical element is potassium or calcium.

A. Hybrid Language Model for Category-Aspect Pair

For certain category c and a potential aspect word w, we assemble them together as <*c*,*w*> and call it a category-aspect pair.

To utilize the surrounding snippets of the aspect word and capture the two types of features discussed above, we set up a hybrid language model HLM*c*,*w* for each category-aspect pair <*c*,*w*> as follows.

Step 1: Construct the surrounding snippets collection for <*c*,*w*>.

We index all Baidu Baike articles using Lucene[11]. For category-aspect pair <*c*,*w*>, search all occurrences of word *w* in the articles of category *c*, and extract all the surrounding snippets with length of 200 Chinese characters for each snippet.

In the surrounding snippets collection, as we can imagine, there exists a proportion of "outliers", which means the snippets contain the aspect word *w*, but the content of them are not related to the aspect; the occurrence of *w* here is "occasional". Thus we do a simple preprocess to reduce the influence of the outlier snippets: concatenate all snippets into a document *d*, then compute the cosine similarity under vector space model [5] between each snippet and *d*; then filter out at least 30% of snippets with smallest similarity values and save no more than 200 snippets.

Step 2: Build the language model $WLM_{c,w}$ for words information.

After obtaining the surrounding snippets collection through Step 1, we concatenate all snippets into a document *d*, with which infer a unigram language model $WLM_{c,w}$ [12].

For any text *p*, we have

$$P(p \mid WLM_{c,w}) = \prod_{i=1}^{n} P(t_i \mid WLM_{c,w}) \quad (1)$$

where t_i is the *i*th term (word) in the text *p*. And for each term *t*,

$$P(t \mid WLM_{c,w}) = \alpha \cdot \frac{tf(t,d)}{dl_d} + (1-\alpha) \cdot \frac{cf_t}{cs} \quad (2)$$

where α is a weighting parameter between 0 and 1, $tf(t, d)$ is the frequency of *t* occurs in *d*, dl_d is the document length of *d*, cf_t is the frequency *t* occurs in the entire collection, and *cs* is the total number of terms in the whole encyclopedia.

Step 3: Build the language model $CLM_{c,w}$ for categories information.

The difference between Step 2 and 3 is that the terms for language model $CLM_{c,w}$ are not words, but categories. For document *d* which is constructed by all snippets in Step 2, we do not use it to infer a language model directly. Instead, *d* which is a document consisting of words is mapped into a document *d'* consisting of categories by the procedure as follows:

Extract all the entries of Baidu Baike occurred in *d*, and add the categories of each entry into *d'*. For example, if "calcium" is found in *d*, we add its categories "metal", "chemical element", "nutriology" and "milk calcium" into *d'*.

After this mapping, we can obtain a category language model $CLM_{c,w}$ in similar way as Step 2.

For any text *p*, we first map *p* into *p'* in the same way document *d* is processed. Then we have

$$P(p' \mid CLM_{c,w}) = \prod_{i=1}^{n} P(c_i \mid CLM_{c,w}) \quad (3)$$

where c_i is the *i*th term (category) in the *p'*. And for each term *c*,

$$P(c \mid CLM_{c,w}) = \alpha \cdot \frac{tf(c,d')}{dl_{d'}} + (1-\alpha) \cdot \frac{cf_c}{cs} \quad (4)$$

where $tf(c, d')$ is the frequency of *c* occurs in *d'*, $dl_{d'}$ is the length of *d'*, cf_c is the number of articles belonging to category *c*, and *cs* is the total number of articles in the whole encyclopedia.

Step 4: Build the hybrid language model $HLM_{c,w}$.

The hybrid language model $HLM_{c,w}$ comprises two language model instances: $WLM_{c,w}$ and $CLM_{c,w}$, which are based on the surrounding snippets collection of *w* from the articles of category *c*.

For any text *p*,

$$P(p \mid HLM_{c,w}) = \lambda \cdot P(p \mid WLM_{c,w}) + (1-\lambda) \cdot P(p' \mid CLM_{c,w}) \quad (5)$$

where λ is the parameter to adjust the weights of two models.

B. Passage Ranking

Now get back to our task. Given the user query in the form of entity-aspect pair, such as "pear-nutrient", we have to compute the semantic relatedness score, denoted by *score*(*p*), between the aspect word *w* and each candidate passage *p* in the article. For one category-aspect pair <*c*,*w*>, we already know $P(p| HLM_{c,w})$. But usually there are more than one category for an entity, for example, "pear" belongs to four categories: "fruit", "plant", "foodstuff" and "crops". So the weight sum of $P(p| HLM_{c,w})$ for all categories should be used:

$$score(p) = P(p \mid w) = \sum_{i=1}^{k} P(p \mid HLM_{c_i,w}) \cdot P(HLM_{c_i,w} \mid w) \quad (6)$$

where c_i is the *i*th category of the entity, $i = 1, 2, \ldots, k$.

We estimate the conditional probability

$$P(HLM_{c_i,w} \mid w) = P(c_i \mid w) = \frac{P(w, c_i)}{P(w)} = \frac{P(c_i)P(w \mid c_i)}{\sum_{j=1}^{k} P(c_j)P(w \mid c_j)} \quad (7)$$

For each category c_i, we think they are equiprobable, i.e. $P(c_i) = 1/k$, $i = 1, 2, \ldots, k$. So we have

$$P(HLM_{c_i,w} \mid w) = \frac{P(w \mid c_i)}{\sum_{j=1}^{k} P(w \mid c_j)} \quad (8)$$

and

$$P(w \mid c_i) = \frac{df(w, c_i)}{cs(c_i)} \quad (9)$$

where $df(w,c_i)$ is the document frequency of w in all articles of category c_i, and $cs(c_i)$ is the total number of articles of category c_i.

The candidate passages are ranked by $score(p)$ in (6).

C. Model Database

For each category-aspect pair $<c,w>$, the construction of $HLM_{c,w}$ is a time-consuming procedure, because all the articles of category c is retrieved and searched. On average, about 4 to 10 seconds time is required for one query.

To make the algorithm available in online applications, we have to construct all the $HLM_{c,w}$ and store them into database in advance.

In Baidu Baike, there are totally 358,057 categories and more than 50,000 terms after removing stopwords and rare words. Thus the amount of category-aspect pairs is more than 1.79×10^{10}, which is a huge number we can't accept. So it is necessary to reduce the scale.

We only save the model for the category-aspect pair$<c,w>$ which satisfies the two conditions below:

1) The category c should contain at least 300 articles.

2) $P(w|c)$, as shown in (9), is larger than 0.3 and $df(w,c)$ is larger than 50.

There are 1660 categories which have at least 300 articles in Baidu Baike. The categories with small number of articles are almost rare and concerned by users by little chance or created by editors' mistakes.

By Condition 2, we reduce the scale of aspect words dramatically. The aspect words should reflect generality of entities under the same category. So we select aspect words by $P(w|c)$.

After the filtering procedure, the scale of the model database is reduced to less than one million, which is an acceptable value.

D. Rejection of Unreliable Answers

The methods based on LSA or ESA will always give an answer – the passage most related to the aspect word, whatever the entity-aspect query is, even if the query is meaningless such as "pear- pathogenesis", because they just compute and a result will come out finally for any situation. So it is very difficult to guarantee the quality of the result. Sometimes, a meaningless answer is much worse than no answer.

Our approach supplies a way to reject to give user answers with low confidence: if the $HLM_{c,w}$ models needed in (6) do not exist in the model database built in last section, the system can choose not to return any answer to users, because in this situation, w may not be a proper aspect word or our approach cannot handle it confidently.

IV. Experiments

A. Data Set

There is no open data set for the evaluation of our task, so we built the data set under the help from several volunteers.

First we collected more than 10,000 questions from Baidu Zhidao[4], which is a Chinese community question-answering website as Yahoo! Answers[5]. After a preliminary filtering by program, we picked out proper questions as those in Table I, and converted them into entity-aspect pair.

For each entity-aspect pair, we cut the corresponding article of Baidu Baike into passages. Each passage is a section or some continuous paragraphs with length no longer than 500 Chinese characters. The average number of passages of each article is 26.05.

Then the volunteers gave a label "related" or "unrelated" to each passage with respect to the aspect word. We totally labeled 411 queries. The average number of related passages for each query is 3.10.

We classify all queries into two types. For the queries of Type 1, the aspect word appears in the text of the article, while for the queries of Type 2, the aspect word does not appear in the article.

B. Analysis of the Results

We compared our approach with three methods: vector space model, LSA and ESA.

For all methods, we removed the stop words and the words with low frequency from text. The size of remaining word list is about fifth thousand. And for anyone of the methods, we didn't do any keyword expansion.

We trained the LSA and ESA model with the top 10,000 articles with the largest pagerank value in Baidu Baike. The dimension of LSA model is set to 200. And the weighting parameter λ in (5) for our approach is set to 0.2 empirically.

TABLE III
Result for all queries

Method	VSM	LSA	ESA	HLM
MAP@10	0.4901	0.5938	0.3836	**0.6835**
MRR@10	0.5466	0.6422	0.3946	**0.7518**
SUC@1	0.4185	0.4647	0.2728	**0.6302**
SUC@3	0.6204	0.7908	0.4541	**0.8491**
SUC@5	0.7129	0.8808	0.5661	**0.9270**

For evaluating the performance of each method, we use the classical metrics in information retrieval field: MAP@n, MRR@n and SUC@n.

MAP@n is the mean average precision for the first n results.

MRR@n is the mean reciprocal rank for the first n results.

[4] http://zhidao.baidu.com.
[5] http://answers.yahoo.com.

SUC@n is the mean success rate for the first n results. For each test case, it is one "success" if there is at least one result is labeled "related" in the first n results.

The results for all queries are shown in Table III. And the results for queries of Type 1 and 2 are shown in Table IV and Table V respectively. Our approach is noted as HLM.

From the results, we can see that our approach consistently outperforms all other methods including LSA. On SUC@1, which represents the success rate at the first result, our approach performs significantly higher than other methods for about 20 percent.

Comparing the results in Table IV and Table V, the performance of all methods drops to certain extent. VSM turns to a random ranking in Table V, because without the aspect word appearing in the text, VSM can't distinguish any passages.

Even for queries of Type 1, HLM is better than VSM. This is because even if the aspect word appears in some passage, the content of the passage may not be related to the aspect, either. The appearance can be an outlier.

TABLE IV
RESULT FOR QUERIES OF TYPE 1

Method	VSM	LSA	ESA	HLM
MAP@10	0.6912	0.6949	0.4176	**0.7353**
MRR@10	0.7896	0.7505	0.4252	**0.8297**
SUC@1	0.6862	0.5904	0.3138	**0.7340**
SUC@3	0.8723	0.9043	0.4947	**0.9149**
SUC@5	0.9255	0.9681	0.5826	**0.9734**

TABLE V
RESULT FOR QUERIES OF TYPE 2

Method	VSM	LSA	ESA	HLM
MAP@10	0.3205	0.5086	0.3550	**0.6398**
MRR@10	0.3418	0.5509	0.3688	**0.6862**
SUC@1	0.1928	0.3587	0.2383	**0.5426**
SUC@3	0.4081	0.6951	0.4198	**0.7937**
SUC@5	0.5336	0.8072	0.5522	**0.8879**

It is worth noticing that ESA performs badly in this task. We think the reason lies in the fact that ESA uses the articles themselves as concepts directly. It is easy to understand that two different aspect words w_1 and w_2 for category c may have a lot of co-occurrence in the articles under category c. So ESA cannot distinguish w_1 and w_2 easily.

V. CONCLUSION

We propose a novel encyclopedia-specific method to retrieve passages which are semantically related to an aspect query from a given article in the encyclopedia. The method captures the expression word features and categorical word features in the surrounding snippets of the aspect words by setting up massive hybrid language models. These local models outperform the global models such as LSA and ESA. By store these models into database in advance, we make a trade-off between time cost and space cost so as to make the method usable for online situation. In addition, our approach has the ability to reject to give user answers with low confidence.

REFERENCES

[1] J. Lin, D. Quan, V. Sinha, K. Bak-shi, D. Huynh, B. Katz, and D. R. Karger, "The role of context in question answering systems," in *Proceedings of the 2003 Conference on Human Factors in Computing Systems*, 2003.

[2] S. Ye, T. Chua and J. Lu, "Summarizing Definition from Wikipedia," in *Proceedings of the 47thAnnual Meeting of the ACL. Singapore*, 2009.

[3] C. Li, N. Yan, S. B. Roy, L. Lisham and G. Das, "Facetedpedia: Dynamic Generation of Query Dependent Faceted Interfaces for Wikipedia," in *Proceedings of International World Wide Web Conference, Raleigh, North Carolina, USA*, 2010.

[4] R. Hahn, C. Bizer, C. Sahnwaldt, C. Herta, S. Robinson, M. Brgle, H. Dwiger, and U. Scheel, "Faceted Wikipedia Search," in *13th International Conference on Business Information Systems (BIS)*, 2010.

[5] R. B. Yates and B. R. Neto, *Modern Information Retrieval*, Addison Wesley, New York, NY. 1999.

[6] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA. 1998.

[7] A. Budanitsky and G. Hirst, "Evaluating Wordnet-based Measures of Lexical Semantic Relatedness," *Computational Linguistics*, 2006, pp. 13-47.

[8] P. Roget, *Roget's Thesaurus of English Wordsand Phrases*, Longman Group Ltd., 1852.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harsh-man, "Indexing by Latent Semantic Analysis," *Journal of the American Society For Information Science*, 1990, pp. 391-407.

[10] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis," in *Proceedings of IJCAI*, 2007, pp. 1606-1611.

[11] E. Hatcher and O. Gospodnetic, *Lucene in action*, Manning Publications, 2005.

[12] J. M. Ponte, and W. B. Croft, "A Language Modeling Approach to Information Retrieval," in *Proceedings of the 21st Intl. ACM SIGIR Conf.*, 1998, pp. 275-281.