Using Encyclopedic Knowledge to Understand Queries

Kejun Zhao* Xiaofeng Meng* Hehan Li* Zhongyuan Wang*†
*Renmin University of China, Beijing, China
†Microsoft Research China, Beijing, China
{kejunzhao,xfmeng,hehanli}@ruc.edu.cn zhy.wang@microsoft.com

ABSTRACT

Query understanding is a challenging but beneficial task. In this paper, we propose a context-aware method to use the encyclopedic knowledge to aid in query understanding. Given a query, we first use a dictionary constructed from the encyclopedic knowledge bases to detect the possible entities and their associated categories. Then, we use a topic based method to derive semantic information from the query. By comparing the topical similarity between various candidate phrases, we get the most likely entities and their related categories. Experimental results show that our method has achieved a great improvement over previous approaches and the efficiency is acceptable for online search.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: [Information Search and Retrieval]

General Terms

Algorithms, Experimentation

Keywords

query understanding; named entity recognition and categorization; topic model; knowledge base

1. INTRODUCTION

With the explosive growth of data on the web, users are not satisfied with the numerous plain web pages returned by search engines, leading to the requirement of more intelligent search related applications such as entity search, query recommendation and question answering. To achieve these goals, first of all, we need to understand the queries.

To understand a query, first we need to recognize the entities in it and then map them to the most likely categories or concepts. During this process, there are actually two tasks. The first is named entity recognition(NER), which is to detect the named entities in plain text. The second is entity categorization, which is to infer the most appropriate category for each entity. For example, given the Chinese query "爱回家2014过新年在第几集"(Which episode of *Come Home Love* shows the 2014 New Year festival), the search engine needs to recognize that "爱回家"(*Come Home Love*) is a TV play in order to generate related search result. A person can easily understand short noisy text like queries because human beings have equipped abundant knowledge in our brain. However, this is not an easy task for computer.

- First, it is hard to decide the boundary of the entities. For the given query, there is a song named "回家"(*Come Home*), a book named "爱回家2"(*Come Home Love2*). Chunking decision is crucial for NER.
- Second, it is hard to decide the conceptual meanings of the entities. For example in Wikipedia, there are three entities named "爱回家"(*Come Home Love*), one is a book, one is a movie and the other is a TV play. The only way to disambiguation is by the context.

These two challenges are not isolated. As we can see from the example, correctly detecting named entities is essential for categorization. On the other hand, capturing the categorical information, which can be considered as the semantics suggested by the context, will help to decide the boundaries of the named entities.

This problem is not well solved. Some examples of incorrectly identified entity mentions in queries are given in Table 1. As we can see, the well-known Stanford NER [2] tool performs well only for named entities of a few categories such as person, location, and organization, while these kinds of entities only make up no more than 20% of all the entities

NMSearch'15, October 23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2810355.2810358.

Table 1: Example of incorrect entity recognition results

Query	Stanford	Longest
邓卓棣美国国籍(Deng Zhuoli United States nationality)	邓卓棣(Deng Zhuoli)(PER), 美国(United States)(LOC)	邓卓棣(Deng Zhuoli), 美国国籍(United States nationality)
梦幻西游大唐符石(Tang runestone of the *Menghuanxiyou* computer game)	梦 幻 西(Meng Huanxi)(PER)	梦幻西游大(*Menghuanxiyouda*-mobile game), 唐(the Tang dynasty), 符石(runestone)
爱回家2014过新年在第几集(Which episode of *Come Home Love* shows the 2014 New Year festival)		爱回家2 (*Come Home Love2*-book), 过新年(*New Year*-song)

appearing in real queries from our sampled data. Besides, most of the existing approches adopt a layered framework, which means chunking and POS tagging first, then followed by NER and dependency detection. In such a framework, errors occurring in the early layers are inevitably passed on to later layers. Thus the performance is largely limited by the precision of segmentation, especially for languages like Chinese. The longest cover scheme uses a dictionary-based approach, which detects the longest segment matching the entity names in the knowledge base for a given query. As the result shows, this scheme is also error-prone because it does not capture the semantics of the text.

In this paper, we propose a knowledge-based context-aware method to recognize and categorize named entities in queries, denoted as **NERC** (**N**amed **E**ntity **R**ecognition and **C**ategorization). Here, by recolonizing, we mean to detect the keyphrases in the query, which are entities existing in the knowledge base. That is, we suppose the knowledge base is complete and entities out of vocabulary (OOV) are ignored. Specifically, given a query, we first use a dictionary constructed from certain knowledge bases to discover the possible entities and their associated categories. Then, we use a topic based method to derive semantic information from the query. We then calculate the similarity between the semantics and the associated categories, and get the most likely entities and their related categories.

In the rest of the paper, we first introduce some related work in section 2 and then describe the details of our method in section 3. Experimental results are shown in section 4.Finally, section 5 draws conclusions.

2. RELATED WORK

Much research has been devoted to named entity recognition and categorization [10]. However, most of the achievements are based on full text. Supervised learning methods such as linear mixture models [4], SVM, CRF [13] and HMM [3] are widely used for NER. However, these methods cannot work well on queries which are usually sparse, noisy and lack of syntax structure. Some researchers also pay their attention to short text like tweets [14]. The methods they use are still dominated by the above machine learning models, except that some tweet-specific features are added in such as hash tags and smileys.

Some other researchers take advantage of knowledge bases to solve the problem. Machine learning techniques are involved to utilize the context for disambiguation [7, 9]. Our approach is similar with them in entity recognition. However, in their work, only unambiguous entity terms in the context are adopted to disambiguate the target entity. As the context is less qualified in queries, we want to derive more semantics rather than the co-occurred entities. Other researchers also try to capture more semantics behind the short text [6, 12]. However, their work start from pre-identified entity mentions, without specifying how these mentions are identified. As detecting the entity mentions is already a challenge for short text, especially for Chinese short text where chunking and POS tagging are even less reliable, it is not realistic to suppose that the terms are always correctly segmented and identified. Our paper also incorporates knowledge bases and semantic information to analyze short texts, but we form a unified approach to fill the gap between entity recognition and categorization.

3. NAMED ENTITY RECOGNITION AND CATEGORIZATION

Our framework consists of two parts. As shown in Figure 1, an offline part mines the topic distribution of categories, which is introduced in section 3.1; and an online part detects the most likely entities for given queries, which is introduced in section 3.2.

3.1 Topic distribution

The intuition behind our method is that the topics under a query should be similar with the topics under related categories. Consider the query "爱回家2014过新年在第几集"(Which episode of *Come Home Love* shows the 2014 New Year festival). A person can easily recognize the entity "爱回家"(*Come Home Love*) and decide that the most like-

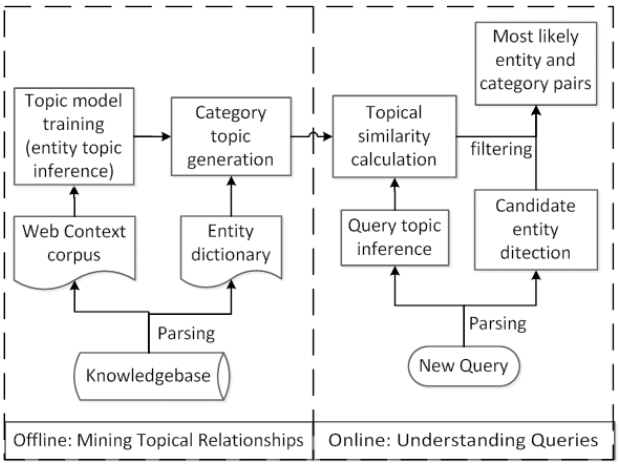

Figure 1: Framework overview

ly category of the entity should be TV series, because the context "集"(episode) strongly suggests that the topic of this query is about TV plays. We try to capture this semantic information by topic modeling.

3.1.1 Knowledge base

Many knowledge bases have emerged in recent years including Wikipedia[1], DBpedia[2], Yago [5], etc. In this work, we use the Chinese version of Wikipedia which contains more than 800,000 articles. Besides, since the entity coverage is limited in Wikipedia, we use another knowledge base called Baidubaike[3] as complement in entity recognition.

We first construct an entity dictionary using the entity pages, redirect pages and disambiguation pages in the knowledge base as described in [11]. Each entity name associates with a list of categories. Similarly, for entities in Baidubaike, we extract the category tags from the entity pages. Note that different entities may share the same name. Here we don't distinguish them in names, instead, we regard them as entities sharing the same name but with different categories. One may argue that there exist different entities sharing the same name and the same category. This case requires finer-grained entity disambiguation techniques, which is more an entity linking task and not discussed in this work.

3.1.2 Training a topic model

Topic modeling represents a document as a distribution over a number of topics. The topics are multinomial distributions over the vocabulary. Since topics indirectly represent the co-occurrence patterns of the words in the vocabulary, they essentially capture the relationships between the words. Therefor, topic modeling is a good way to reduce dimensions to represent a document, while the same time to preserve the semantics under a document. In our paper, we use the Latent Dirichlet Allocation (LDA) model [1].

We extract entity pages from Wikipedia and Baidubaike to generate the training corpus. For each entity page, we remove the html tags and get the text content of the page, then do text segmentation, and remove the stop words and punctuation, thus form a document. After the process, we finally collected 3 million documents as the training corpus.

We then use JGibbLDA[4] to train the LDA model. It adopts a collapsed Gibbs sampling method to estimate the hidden variables. We set α, the document topic proportion hyper parameter, as 0.1, and β, the word-topic distribution hyper parameter as 0.01, with 1000 iterations. We set the number of topics as 50, 100, 200, and it turns out 100 can generate a relatively better result at a lower computation cost in our experiment.

3.1.3 Estimating topic distributions of categories

After training, LDA assigns a topic distribution θ_d to each document. As each document is derived from an entity page, it is reasonable to regard θ_d as a representation of the entity. On the other side, each category contains several entities. We aggregate these entities' topic distribution θ_e as the category's topic distribution.

$$\theta_c = \sum_e P(e|c) * \theta_e \quad e \in c \tag{1}$$

Where θ_e is the topic distribution of e, i.e. the topic distribution θ_d of the entity page d_e. $p(e|c)$ is the typicality of entity e in category c. We set $p(e|c)$ equally as $1/n$, where n is the number of entities in category c. More sophisticated assignment of $p(e|c)$ may improve the precision but will also involve much more computation. We leave this for future work.

3.2 Query NERC

3.2.1 Candidate entity generation

We use the entity dictionary to detect candidate entities. In order to get higher recall, we search each possible segment of the query. For example, for the given query "爱回家2014过新年在第几集", some of the detected candidate entities with the associated categories are shown in Table 2.

[1]http://www.wikipedia.org/
[2]http://wiki.dbpedia.org/
[3]www.baike.baidu.com is one if the largest Chines encyclopedias containing more than 7.05 million entities
[4]http://jgibblda.sourceforge.net/#Griffiths04

Table 2: Candidate Entities with Relatedness Score

Entity	Category	SD_{KL}	r_{idf}	r_{key}
爱(*Love*)	song	0.166	0.0023	0.0065
回家(*Come Home*)	TV series	0.175	0.0103	0.0094
	song	0.166	0.0097	0.0089
爱回家(*Come Home Love*)	movie	0.134	0.0103	0.0111
	TV series	0.175	0.0118	0.0145
爱 回 家2(*Come Home Love2*)	book	0.132	0.0121	0.0102
2014	year	0.014	0.0002	0.0001
新年(*New Year*)	festival	0.106	0.0013	0.0016
过新年(*New Year*)	song	0.166	0.0009	0.0008

3.2.2 Estimating topic distributions of queries

Inferring the topics of a given query is to directly apply the trained topic model on it. Taking a query as a document, the likelyhood function for the topic distribution θ_q is:

$$\begin{aligned}\mathscr{L}(\theta_q|q,\Phi,\alpha) &= \prod_i \phi_{w_i|z_i}\theta_{z_i} \times \frac{\Gamma(\sum_t \alpha_t)}{\prod_t \Gamma(\alpha_t)} \prod_t \theta_t^{\alpha_t-1} \\ &\propto \prod_i \phi_{w_i|z_i} \prod_t \theta_t^{n_t|q+\alpha_t-1}\end{aligned} \tag{2}$$

where w_i is the i-th term in query q; z_i is a variable denoting the topic of the i-th term w_i; $n_t|q$ is the total number of terms in the query assigned to topic t. Φ is the topic-term distributions estimated in the training process.

3.2.3 Entity categorization and filtering

After detection, a number of candidates are generated, denoted as $E = \{e_1, e_2, \dots\}$. These candidates might conflict with each other. Given a query q, we assume it should bias on the topics. That is, when we derive the topic distribution θ_q, the vector should be skewed on a few dimensions, representing the semantics of this query, and should be coherent with the topic distribution θ_c of the most likely categories.

We use symmetrised KL-divergence to evaluate the similarity of the topic distribution of the query and each candidate category.

$$\begin{aligned}SD_{\mathrm{KL}}(\theta_q\|\theta_c) &= D_{\mathrm{KL}}(\theta_q\|\theta_c) + D_{\mathrm{KL}}(\theta_c\|\theta_q) \\ D_{\mathrm{KL}}(\theta_q\|\theta_c) &= \sum_i \theta_q(i) * \ln\frac{\theta_q(i)}{\theta_c(i)}\end{aligned} \tag{3}$$

For the query "爱回家2014过新年在第几集", the SD_{KL} score of some candidate categories are shown in the third column of Table 2.

With the KL-divergence only we are not able to pick out the most likely entities because it only measures the semantic similarity between the query and the category. Thus we involve the idf and keyphraseness of the entities to calculate the relatedness score of a candidate entity e with the given query q:

$$\begin{aligned}r_{idf}(e,c,q) &= \frac{idf_e}{SD_{\mathrm{KL}}(\theta_q\|\theta_c)+\epsilon} \\ r_{key}(e,c,q) &= \frac{key_e}{SD_{\mathrm{KL}}(\theta_q\|\theta_c)+\epsilon}\end{aligned} \tag{4}$$

where the idf_e, key_e and SD_{KL} are all normalized. ϵ is a smoothing factor. Specifically, the idf is calculated as:

$$idf_e = \ln\frac{|D|}{count(d_e)} \tag{5}$$

where $|D|$ is the total number of documents in Wikipedia, and $count(d_e)$ is the number of documents containing term e. Keyphraseness is calculated as:

$$key_e = \frac{count(d_{key})}{count(d_e)} \tag{6}$$

where $count(d_{key})$ is the number of documents where e appears as an anchor text. The relatedness score of the example query "爱回家2014过新年在第几集" with the candidate entities are shown in the fourth and fifth columns of Table 2.

Now we have calculated relatedness score for each candidate entity, the problem becomes to find out a subset of the candidate entities with the highest relatedness score while the same time without any conflict, i.e.
select $E_{max} = \{e_t, \dots, e_r\} \subseteq E$, satisfy:
$E_{max} = argmax_{E_q} \sum_{e\in E_q} \ell(e) * s(e,c,q)$
and $\forall t, r, e_t \cap e_r = \emptyset$.
where $\ell(e)$ is the length of entity e, which is added otherwise the summation of scores of shorter terms will be likely to exceed longer terms.

We then represent them in a graph and use a depth first search algorithm to generate E_{max}, the most appropriate entity set. Depth-first search can be very time-consuming. To improve the efficiency of out method, we first filter out a large part of the candidate entities via a stop entity list, and for each entity remained, only the top 2 related categories are kept. With this scheme, we are able to prune more than 80% nodes in our search tree for each query; and it has been proved that this will not introduce decline of the performance in the experiment.

4. EXPERIMENT

4.1 Dataset

We test our method on real queries. We sampled 1000 queries from the Sougou World Wide Web competition[5] query dataset and invited 3 students to label them.

[5]http://iir.ruc.edu.cn/ndbccup2015/stsjj.jsp

Table 3: Query understanding result

Method	ER-Pre	ER-Rec	F1	EC-Pre
Wikify!20%	0.590	0.391	0.469	0.743
Wikify!30%	0.362	0.473	0.410	NA
Stanford	0.308	0.182	0.229	NA
NERC-kl	0.578	0.693	0.630	0.782
NERC-idf	0.643	0.688	0.665	0.789
NERC-key	**0.686**	**0.735**	**0.710**	**0.805**

4.2 Entity Recognition

To evaluate the performance of entity recognition, we compare our method with several alternative approaches listed as follows.

- **Wikify!** [8]: they adopt keyword extraction techniques and a threshold scheme to recognize entities and filter out non-keyphrases from documents. As we use different corpus, we re-implemented their method. In the Wikify! system, the author use a ratio of 6% to determine the number of keywords to be extracted from a document, since the density of keyphrases in short text should be higher than that observed in full text, we set the ratio of keyphrases as 20%/30% to filter out non-keyphrases.
- **Stanford** [2]: the Stanford NER tool[6].
- **NERC-kl**: one of our NERC methods, using only SD_{KL} as the relatedness score.
- **NERC-idf**: one of our NERC methods, using r_{idf} as the relatedness score.
- **NERC-key**: one of our NERC methods, using r_{key} as the relatedness score.

The result is shown as Table 3. As pointed out in [8], the precision and recall scores in keyword extraction are traditionally low. Even though, our method NERC-key achieves 9.6% improvement on ER-Pre(entity recognition precision) and 34.4% on ER-Rec(entity recognition recall) over Wikify! with the keyprhase ratio set as 20%. When increasing the keyphrase ratio, the recall of Wikify! is improved but precision decreases more, leading to even lower F1 score. Stanford NER tool performs the worst.

4.3 Entity Categorization

Because different people tend to give different description of the same category for the same entity, and it is hard for each volunteer to select among millions of categories to label the entities, we choose to run our algorithm first, and provide the volunteers the results of our method, and ask them to decide whether it is correct or not. Then the correct entity with correct category scores 1, the correct entity with related category scores 0.5, and others score 0. We measure the precision as:

$$Precision = \frac{\sum_e score_e}{Correct(e)} \tag{7}$$

where $Correct(e)$ is the number of correct entities. This metric eliminates the influence of precision of entity recognition, and only evaluate the performance of categorization.

Unfortunately, we find it is not easy to make a system comparison because to our knowledge most of the existing methods are only designed for a handful of categories such as people, location, organization and some domain-specific proper nouns. However, our method targets on millions of categories. Therefore, we compare our method with the Wikify! system again. Given a query, we use the Wikify! system to link the keywords to certain entities in Wikipedia, and extract the categories from the selected entity pages, then compare them with the results generated by our method. The entity categorization precision(EC-Pre) is shown in the last column of Table 3.

Some examples of the detected entities and the categories are shown in Table 4.

Table 4: Example of Entity Categorization

Query	Entity	Category
爱回家2014过新年在第几集	爱回家(*Come Home Love*)	TV series
	2014	year
	新年(the New Year festival)	festival
梦幻西游大唐符石	梦幻西游(the *Menghuanxiyou* computer game)	game
	大唐(Tang)	game
	符石(runestone)	game
宜宾县李场镇龙川村(Longchuan village, Lichang, Longchuan city)	宜宾县(Yibin city)	city
	李场镇(Lichang town)	town
	龙川村(Longchuan village)	village

4.4 Efficiency

We evaluate the efficiency of our method. The running environment is 4 cores of Intel Xeon E5506 2.0GHz CPU, 32GB RAM, with Ubuntu 10.04 LTS. Figure 2 shows the running time requirement of the NERC process. Over 80% of the queries are processed within 100ms, and most of them are within 10ms. It indicates that our method is efficient for on-line queries, and can be adopted for real-time search.

[6]http://nlp.stanford.edu/software/CRF-NER.shtml

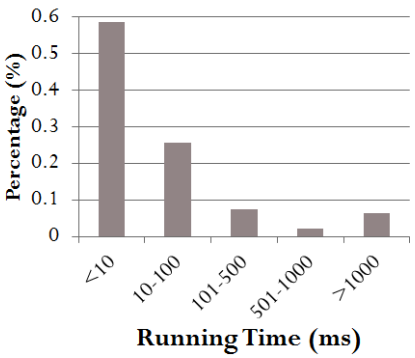

Figure 2: Algorithm Efficiency

5. CONCLUSIONS

Query understanding is important and challenging. In this paper, we propose a knowledge-based context-aware method to recognize and categorize named entities in queries. Experiments on real query dataset show that our method is very effective and efficient. The refinement of the ontology and finer grained disambiguation of entities sharing the same categories are left for future work.

6. ACKNOWLEDGMENTS

This research was partially supported by the grants from the Natural Science Foundation of China (No. 61379050, 91224008); Specialized Research Fund for the Doctoral Program of Higher Education(No. 20130004130001); the National 863 High-tech Program (No. 2013AA013204).

7. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, 2005.

[3] G. Fu and K. Luke. Chinese named entity recognition using lexicalized hmms. *SIGKDD Explorations*, 7(1):19–25, 2005.

[4] J. Gao, M. Li, C. Huang, and A. Wu. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574, 2005.

[5] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[6] D. Kim, H. Wang, and A. H. Oh. Context-dependent conceptualization. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.

[7] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *Proceedings of the AAAI WikiAI workshop*, volume 1, pages 19–24, 2008.

[8] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 233–242, 2007.

[9] D. N. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 509–518, 2008.

[10] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[11] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.*, 27(2):443–460, 2015.

[12] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2330–2336, 2011.

[13] X. Wu, X. Lin, X. Wang, C. Wu, Y. Zhang, and D. Yu. An improved CRF based chinese language processing system for SIGHAN bakeoff 2007. In *Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008*, pages 155–160, 2008.

[14] Q. Zhou. Evaluation report of the fourth chinese parsing evaluation: Cips-sighan-parseval-2014. *CLP 2014*, page 146, 2014.