Detecting Candidate Named Entities in Search Queries

Areej Alasiry
Birkbeck, University of London
United Kingdom, London
areej@dcs.bbk.ac.uk

Mark Levene
Birkbeck, University of London
United Kingdom, London
mark@dcs.bbk.ac.uk

Alexandra Poulovassilis
Birkbeck, University of London
United Kingdom, London
ap@dcs.bbk.ac.uk

ABSTRACT

The information extraction task of Named Entities Recognition (NER) has been recently applied to search engine queries, in order to better understand their semantics. Here we concentrate on the task prior to the classification of the *named entities* (NEs) into a set of categories, which is the problem of detecting candidate NEs via the subtask of query segmentation.We present a novel method for detecting candidate NEs using grammar annotation and query segmentation with the aid of top-n snippets from search engine results and a web n-gram model, to accurately identify NE boundaries. The proposed method addresses the problem of accurately setting boundaries of NEs and the detection of multiple NEs in queries.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval —*Query formulation*

Keywords: named entity recognition, query logs.

1. INTRODUCTION

The challenges posed by the structure of search engine queries, such as their short length, lack of grammar and orthographic features, imply that classical NER techniques cannot be used without modification. In response, researchers have recently proposed several approaches to address these challenges. A relatively simple yet effective method is to start with a set of seed instances for each NE category, and then scan the query log in order to extract the corresponding contexts, which can then be used to identify new instances for each category. Both Paşca [5] and Guo et al. [2] applied this method, with the context including the prefix and/or suffix surrounding instances that were detected. For example, the word 'lyrics' could be the context for song names; the same context appears in queries such as 'song lyrics', 'James Brown songs lyrics', and 'Music and Lyrics'. The instances extracted could be: (i) 'song', which is not a NE, (ii) 'James Brown songs', where the NE boundaries are incorrect, or (iii) 'Music and', where the context is part of the NE which is a movie name. Thus a problem with this approach is that the queries' structure was not considered. Jain and Pennacchiotti [4] presented an unsupervised approach that utilises the syntactic representation of words such as Capitalisation. Although confidence scores were assigned to each extracted NE based on its presence in a Web corpus, relying on Captilisation will often miss many potential NEs in the query log that were not typed with capital letters.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
ACM 978-1-4503-1472-5/12/08.

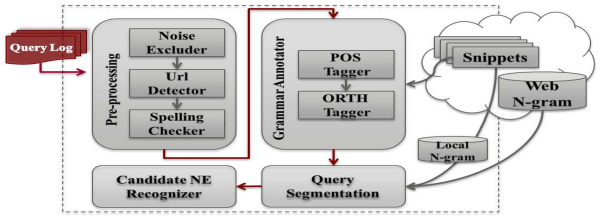

Figure 1: Query Log Candidate NE Recognition

Here, we present a processing pipeline which overcomes the above drawbacks by: (1) Tagging each query token with its part of speech (POS) and the corresponding orthographic features (ORTH) based on the context in which tokens appear, using top-n query snippets; (2) Finding the most probable query segmentation to accurately set the boundaries of candidate NEs; and finally (3) Using these annotations for NER. The contribution of our work lies in the following: (i) Presenting a NER approach for queries by considering English language structural features; (ii) Using query segmentation to better set the boundaries of candidate NEs, especially in multiple NE queries; and (iii) Fine-grained evaluation of NER for search engine queries.

2. CANDIDATE NE RECOGNIZER

Figure 1 presents the processing pipeline we use to identify candidate NEs in a query log. It involves four main stages: (1) Pre-processing the query log; (2) Tagging each query with the corresponding grammatical annotation; (3) Segmenting the query to identify boundaries; and (4) Recognising the candidate NEs with respect to annotations and segmentation.

Pre-Processing: It is important that the query log is processed to exclude *noise*, which is defined as any query that is a sequence of symbols containing no words. *URLs* are also detected in advance before any further natural language processing of the query log. Finally, the *spelling* of each query is checked and corrected.

Grammar Annotation: For each query in the log, the top-n *snippets* are retrieved. Each snippet consists of few sentences from a document retrieved for the query. The advantage of using snippets is that they contain tokens surrounding the query that usually provide enough context to annotate the query, avoiding the cost of parsing complete web documents as in [4]. In addition, the approach we are proposing operates in an offline scenario, therefore snippets are retrieved and stored in advance. The set of grammatical annotations include, parts-of-speech (POS) and orthographic (ORTH) annotations. We assume there are no dependencies between query tokens, adopting the bag-of-words approach as in [1]. It is important to note that we differentiate between common nouns and proper nouns to help identify NEs, and therefore the POS tagging is more specific than that used in [1]. The orthographic features capture the string representation of each query token, such as capital initial, all capital, or mixed letter cases.

Query Segmentation: Many approaches of query segmentations have been proposed in literature. For instance, Hagen et al. [3] used raw n-gram counts and Wikipedia to segment queries. Although their approach achieved high accuracy, in our case we need to segment queries according to their appearance in top-n related snippets. We define query segmentation as follows. For a query Q consisting of tokens $t_1, t_2, ..., t_n$, the set of all possible segmentations is $\mathbf{S(Q)} = \{S_1, ..., S_m\}$, where $m \leq 2^{n-1}$. Each segmentation $S_i \in \mathbf{S(Q)}$ consists of one or more segments, and each segment, say s_{ij}, is a sequence of query tokens that obeys the original order of the tokens. We define the best segmentation S^β as the most probable one over all $S_i \in \mathbf{S(Q)}$. The probability of each S_i is calculated as $Pr(S_i) = \prod_{s_{ij} \in S_i} Pr(s_{ij})$, where $Pr(s_{ij})$ is estimated using a *local* n-gram model ($M_{snippet}$) created from the set of retrieved snippets for the query Q. This probability is smoothed by the probability of the segments given a *web* n-gram model (M_{web}) using an empirically set parameter λ between 0 and 1, to obtain

$$Pr(s_j) = \lambda Pr(s_j|M_{snippet}) + (1 - \lambda)Pr(s_j|M_{Web}).$$

Recognition of Candidate NEs: A small set of rules was defined by examining a random sample of grammatically annotated and segmented queries as described above. In the sample, three main cases were observed: (1) A sequence of proper nouns contained in a segment (approximately 93%), (2) A conjunction, e.g.'&', or preposition, e.g. 'of' followed and preceded by proper nouns such as 'University of Wisconsin' (approximately 2%), and (3) A sequence of proper nouns that include numbers, such as 'Microsoft Office Professional 2003' (approximately 5%). Hence, rules were created to reflect these cases. Applying these rules with respect to segmentation boundaries will result in, for example, the detection of two NEs from a segmented query "[marriot] [new jersey]".

3. EXPERIMENTAL EVALUATION

Experiment: The approach was applied using a 2006 MSN query log consisting of approximately 15M queries. Removing duplicates and after a further pre-processing stage, approximately 5.5M queries were left. Thereafter, the spelling of each query was checked and corrected using Yahoo API provided through the Spelling Suggestion YQL table. In addition, for each query, the top eight query snippets highlighting the query tokens along with their immediate context were retrieved and stored using Google Custom Search API. Then each snippet was Grammatically annotated using GATE and accordingly each query token was then annotated with the most probable POS and orthographic feature. We used the Bing web n-gram model to find the probability of each distinct segment, and smoothed the probability of the same segment given the local n-gram model with $\lambda = 0.6$ being empirically set. Finally, candidate NEs were extracted using the set of defined NER rules. Table 1 presents examples of segmented queries and corresponding extracted NEs.

Table 1: Sample of Extracted Candidate NEs

Segmented Query	Extracted NE/s
[abc] [dallas]	abc & dallas
[microsoft] [speech to text] [windows xp]	microsoft & windows xp
[leon] [final fantasy]	leon & final fantasy
[mercedes benz] [used parts]	mercedes benz

Table 2: Query Log NER Evaluation

	Accuracy	Precision	Coverage
Evaluation I	0.806	0.794	0.9727
Evaluation II	0.678	0.6431	0.9664

Evaluation and Results Analysis: In previous work, NER was evaluated by manually checking a random sample [4], or the top-n [5][2] of instances extracted for each NE class without considering the NEs they missed or the context of the extracted ones. For example, 'safari' is a NE in the query 'download safari', while it is not in the query 'safari trips'. Since we use hand-crafted rules to extract NEs, the quality of these rules should be reflected in the evaluation. Therefore, we manually checked a uniform random sample of 1000 queries.The performance of our method was assessed at the query-level, which is stricter than the previous evaluation methods at the NE-level. True and False Positives (TP, FP), and True and False Negatives (TN, FN) were counted, and these are defined as follows: **TP**: Query has one or more NEs and all were correctly detected. **TN**: Query has no NE and none were detected. **FP**: One or more query tokens were incorrectly tagged by the rules as NE. **FN**: One or more query NEs were incorrectly missed by the rules. The accuracy was measured using two evaluation methods (see table 2): *Evaluation I*, where NEs detection is tagged as correct regardless of its boundary (with accuracy 80.6%) (e.g. extracting 'Sony PS3 news'), and a stricter version, *Evaluation II*, where NEs boundary must be accurate (with accuracy 67.8%) (e.g. extracting 'Sony PS3').

4. REFERENCES

[1] M. Bendersky, W. B. Croft, and D. A. Smith. Joint annotation of search queries. In Proc. of ACL, pages 102-111, 2011.

[2] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In Proc. of SIGIR, pages 267-274, 2009.

[3] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In Proc. of WWW, pages 97-106, 2011.

[4] A. Jain and M. Pennacchiotti. Domain-independent entity extraction from web search query logs. In Proc. of WWW, pages 63-64, 2011.

[5] M. Paşca. Weakly-supervised discovery of named entities using web search queries. In Proc. of CIKM, pages 683-690, 2007.