A Two-Step Named Entity Recognizer for Open-Domain Search Queries

Andreas Eiselt
Yahoo! Research Latin America
Av. Blanco Encalada 2120,
Santiago, Chile
eiselt@yahoo-inc.com

Alejandro Figueroa
Yahoo! Research Latin America
Av. Blanco Encalada 2120,
Santiago, Chile
afiguero@yahoo-inc.com

Abstract

Named entity recognition in queries is the task of identifying sequences of terms in search queries that refer to a unique concept. This problem is catching increasing attention, since the lack of context in short queries makes this task difficult for full-text off-the-shelf named entity recognizers. In this paper, we propose to deal with this problem in a two-step fashion.

The first step classifies each query term as token or part of a named entity. The second step takes advantage of these binary labels for categorizing query terms into a pre-defined set of 28 named entity classes. Our results show that our two-step strategy is promising by outperforming a one-step traditional baseline by more than 10%.

1 Introduction

Search engines are key players in serving as interface between users and web resources. Hence, they started to take on the challenge of modelling user interests and enhance their search experience. This is one of the main drivers of replacing the classical document-keyword matching, a.k.a. bag-of-word approach, with user-oriented strategies. Specifically, these changes are geared towards improving the precision, contextualization, and personalization of the search results. To achieve this, it is vital to identify fundamental structures such as named entities (e.g., persons, locations and organizations) (Hu et al., 2009). Indeed, previous studies indicate that over 70% of all queries contain entities (Guo et al., 2009; Yin and Shah, 2010).

Search queries are on average composed of 2-3 words, yielding few context and breaking the grammatical rules of natural language (Guo et al., 2009; Du et al., 2010). Thus, named entity recognizers for relatively lengthy grammatically well-formed documents perform poorly on the task of Named Entity Recognition in Queries (NERQ).

At heart, the contribution of this work is a novel supervised approach to NERQ, trained with a large set of manually tagged queries and consisting of two steps: 1) performs a binary classification, where each query term is tagged as token/entity depending on whether or not it is part of a named entity; and 2) takes advantage of these binary token/entity labels for categorizing each term within the query into one of a pre-defined set of classes.

2 Related Work

To the best of our knowledge, there have been a few previous research efforts attempting to recognize named entities in search queries. This problem is relatively new and it was first introduced by (Paşca, 2007). Their weakly supervised method starts with an input class represented by a set of seeds, which are used to induce typical query-contexts for the respective input category. Contexts are then used to acquire and select new candidate instances for the corresponding class.

In their pioneer work, (Guo et al., 2009) focused on queries that contain only one named entity belonging to four classes (i.e., movie, game, book and song). As for learning approach, they employed weakly supervised topic models using partially labeled seed named entities. These topic models were trained using query log data corresponding to 120 seed named entities (another 60 for testing) selected from three target web sites. Later, (Jain and Pennacchiotti, 2010) extended this approach to a completely unsupervised and class-independent method.

In another study, (Du et al., 2010) tackled the lack of context in short queries by interpreting query sequences in the same search session as extra contextual information. They capitalized on a collection of 6,000 sessions containing only queries targeted at the car model domain.

International Joint Conference on Natural Language Processing, pages 829–833,
Nagoya, Japan, 14-18 October 2013.

They trained Conditional Random Field (CRF) and topic models, showing that using search sessions improves the performance significantly. More recent, (Alasiry et al., 2012a; Alasiry et al., 2012b) determined named entity boundaries, combining grammar annotation, query segmentation, top ranked snippets from search engine results in conjunction with a web n-gram model.

In contrast, we do not profit from seed named entities nor web search results, but rather from a large manually annotated collection of about 80,000 open-domain queries. We consider search queries containing multiple named entities, and we do not benefit from search sessions. Furthermore, our approach performs two labelling steps instead of a straightforward one-step labelling. The first step checks if each query term is part of a named entity or not, while the second assigns each term to one out of a set of 29[1] classes by taking into account the outcome of the first step.

3 NERQ-2S

NERQ-2S is a two-step named entity recognizer for open-domain search queries. First, it differentiates named entity terms from other types of tokens (e.g., word and numbers) on the basis of a CRF[2] trained with manually annotated data. In the second step, NERQ-2S incorporates the output of this CRF into a new CRF as a feature. This second CRF assigns each term within the query to one out of 29 pre-defined categories. In essence, considering these automatically computed binary entity/token labels seeks to influence the second model so that the overall performance is improved.

Given the fact that binary entity/token tags are only used as additional contextual evidence by the second CRF, these labels can be reverted in the second step. NERQ-2S identifies 28 named entity classes that are prominent in search engine open-domain queries (see table 1). This set of categories was deliberately chosen as a means of enriching search results regarding general user interests, and thus aimed at providing a substantially better overall user experience. In particular, named entities are normally utilized for devising the lay-out and the content of the result page of a search engine.

[1]In actuality, we considered 29 classes: 28 regards named entities and one class for non-entity (token). For the sake of readability, from now on, we say indistinctly that the second step identifies 28 named entity classes or 29 classes.

[2]CRFsuite: http://www.chokkan.org/software/crfsuite

At both steps, NERQ-2S uses a CRF as classifier and a set of properties, which was determined separately for each classifier by executing a greedy feature selection algorithm (see next section). For both CRFs, this algorithm contemplated as candidates the 24 attributes explained in table 2. Additionally, in the case of the second CRF, this algorithm took into account the entity/token feature produced by the first CRF. Note that features in table 2 are well-known from other named entity recognition systems (Nadeau and Sekine, 2007).

4 Experiments

In all our experiments, we carried out a 10-fold cross-validation. As for **data-sets**, we benefited from a collection comprising 82,413 queries, which are composed of 242,723 terms[3]. These queries were randomly extracted from the query log of a commercial search engine, and they are exclusively in English. In order to annotate our query collection, these queries were first tokenized, and then each term was manually tagged by an editorial team using the schema adopted in (Tjong Kim Sang and De Meulder, 2003).

Attributes were selected by exploiting a **greedy** algorithm. This procedure starts with an empty bag of properties and after each iteration adds the one that performs the best. In order to determine this feature, this procedure tests each non-selected attribute together with all the properties in the bag. The algorithm stops when there is no non-selected feature that enhances the performance.

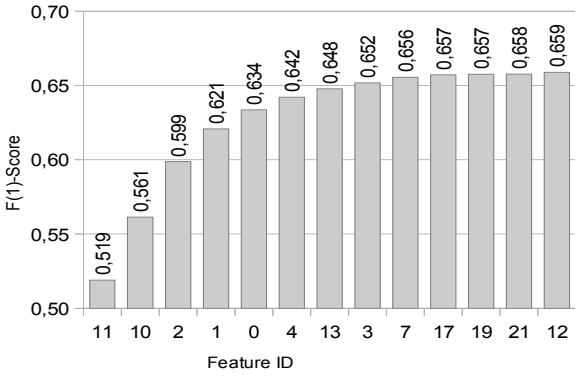

Figure 1: Attributes selected by the greedy algorithm and their respective contribution (baseline). See also table 2 for id-feature mappings.

As for a **baseline**, we used a traditional one-step approach grounded on CRF enriched with 13

[3]Due to privacy laws, query logs cannot be made public.

ID	Name	Example	ID	Name	Example
0	Airline Code	AA, LA, JJ	15	Food	Sushi, Bread, Dessert
1	Beverage	Cocktails, Beer	16	Food Ingredient	Honey, Avocado
2	Brand Name	Bacardi, Apple	17	Food Taste	Sweet, Cheesy
3	Business	Hotel, Newspaper	18	Horoscope Sign	Libra, Taurus
4	Cooking Method	Pressure Cooking	19	Measurement Name	Inches, Kilogram
5	Cuisine	Mexican, German	20	Media Title	Age of Empires 2
6	Currency Name	Dollar, Euros, Pesos	21	Occasion	Festival, Ceremony
7	Diet	Vegan, Fat free	22	Organization Name	Yahoo, Caf Soleil
8	Disease and Condition	Cancer, Diabetic	23	Person Name	Marry Poppins
9	Dish	Ratatouille, Tiramisu	24	Phone Number	3153423595
10	Domain	forbes.com, lan.com	25	Place Name	Chile, Berlin
11	Drink	Bloody Mary, Sangria	26	Product	Camera, Cell phone
12	Email Address	john.doe@example.com	27	Treatment	Steroids, Surgery
13	Event Name	Christmas, Super Bowl	28	*Token* (no NE-class)	how, to, image
14	File Name	msimn.exe, .htaccess			

Table 1: Named entity classes recognized by NERQ-2S.

out of our 24 features (see table 2), which were chosen by running our greedy feature selection algorithm. Figure 1 shows the order that these 13 features were chosen, and their respective impact on the performance. Regarding these results, it is worth highlighting the following findings:

1. The first feature selected by the greedy algorithm models each term by its non-numerical characters (id=11 in table 2). This attribute helps to correctly tag 80.42% of the terms when they are modified (numbers removed).

2. The third chosen feature considers the value of the following word, when tagging a term (id=2 in table 2). This attribute helps to correctly annotate 79.68%, 74.55% and 74.87% of tokens belonging to person, place and organization names, respectively.

3. Our figures also point out to the relevance of the three word features (id=0,1,2 in table 2). These features were selected in a row, boosting the performance from $F(1) = 0.561$ to $F(1) = 0.634$, a 13.01% increase with respect to the previously selected properties.

In summary, the performance of the one-step baseline is $F(1) = 0.659$. In contrast, figure 2 highlights the 16 out of the 25 features utilized by the second phase of NERQ-2S. Note that the "*new*" bar indicates the token/entity attribute determined in the first step. Most importantly, NERQ-2S finished with an $F(1) = 0.729$, which means a 10.62% enhancement with respect to the one-step baseline. From these results, it is worth considering the following aspects:

1. In terms of features, 11 of the 13 attributes used by the one-step baseline were also exploited by NERQ-2S. Further, NERQ-2S profits from four additional properties that were also available for the one-step baseline.

2. The five more prominent properties selected by the baseline, were also chosen by NERQ-2S with just a slight change in order.

3. The "*new*" feature achieves an improvement of 23.51% ($F(1) = 0.641$) with respect to the previous selected property. The impact of the entity/token attribute can be measure when compared with the performance accomplished by the first five features selected by the baseline ($F(1) = 0.634$).

In light of these results, we can conclude that: a) adding the entity/token feature to the CRF is vital for boosting the performance, making a two-step approach a better solution than the traditional one-step approach; and b) this entity/token property is complementary to the list shown in table 2.

The confusion matrix for NERQ-2S shows that errors, basically, regard highly ambiguous terms. Some interesting misclassifications:

1. Overall, 17.38% of the terms belonging to place names were mistagged by NERQ-2S. From these, 72.11% were perceived as part of organization names.

2. On the other hand, 17.27% of the terms corresponding to organization names were mislabelled by NERQ-2S. Here, 15.52% and 12.84% of these errors were due to the fact that these terms were seen as tokens and parts of place names, respectively.

ID	Feature	Example
	Word Features	
0	Current term (t_i)	abc123
1	Previous term (t_{i-1})	before
2	Next word (t_{i+1})	after
	N-grams	
3	Bi-gram of t_{i-1} and t_i	before abc123
4	Bi-gram of t_i and t_{i+1}	abc123 after
	Pre- & Postfix	
5	1 leftmost character from t_i	a
6	2 leftmost characters from t_i	ab
7	3 leftmost characters from t_i	abc
8	1 rightmost character from t_i	3
9	2 rightmost characters from t_i	23
10	3 rightmost characters from t_i	123
	Reductions	
11	t_i without digits	abc
12	t_i without letters	123
	Word Shape	
13	Shape of t_i ("a" represents letters; "0" digits, "-" special characters)	aaa000
14	Shape of t_i (same elements joined)	a0
	Position & Lengths	
15	Position of t_i from left	3
16	Position of t_i from right	2
17	Character length of t_i	6
	Boolean	
18	t_i is a number? (only digits)	*false*
19	t_i is a word? (only letters)	*false*
20	t_i is a mixture of letters and digits?	*true*
21	t_i contains "."?	*false*
22	t_i contains apostrophe?	*false*
23	t_i contains other special characters?	*false*

Table 2: List of used features. Examples are for the third term of query "*first before abc123 after*".

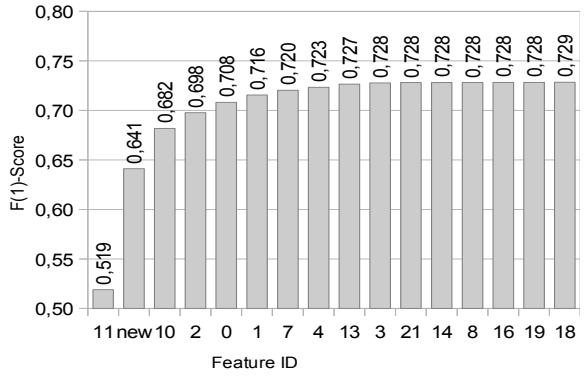

Figure 2: Attributes selected by the greedy algorithm and their respective contribution (NERQ-2S). See also table 2 for id-feature mappings. The word "*new*" denotes the binary token/entity attribute determined in the first step.

Incidentally, NERQ-2S mislabelled 10.40% of the tokens (non-named entity terms), while the one-step baseline 17.57%. This difference signals the importance of first-step consisting of an specialized and efficient token/entity term annotator. With regard to the first step of NERQ-2S, nine out of the 24 properties were useful, and the first step finished with an $F(1) = 0.8077$. From these nine attributes, eight correspond to the top eight features used by our one-step baseline, and one extra attribute (id=20). Thus, the discriminative probabilistic model learned in this first step is more specialized for this task. That is to say, though the context of a term might be modelled similarly, the parameters of the CRF model are different.

The confusion matrix for this binary classifier shows that 11.44% of entity terms were mistagged as token, while 22.24% of tokens as entity terms. This means a higher percentage of errors comes from mislabelled tokens.

On a final note, as a means of quantifying the impact of the first step on NERQ-2S, we replaced the output given by the first CRF model with the manual binary token/annotations given by the editorial team. In other words, the "*new*" feature is now a manual input instead of an automatically computed property. By doing this, NERQ-2S increases the performance from $F(1) = 0.729$ to $F(1) = 0.809$, which means 10.97% better than NERQ-2S and 22.76% than the one-step baseline. This corroborates that a two-step approach to NERQ is promising.

5 Conclusions and Further Work

This paper presents NERQ-2S, a two-step approach to the problem of recognizing named entities in search queries. In the first stage, NERQ-2S checks as to whether or not each query term belongs to a named entity, and in the second phase, it categorizes each token according to a set of predefined classes. These classes are aimed at enhancing the user experience with the search engine in contrast to previous pre-defined categories.

Our results indicate that our two-step approach outperforms the typical one-step NERQ. Since our error analysis indicates that there is about 11% of potential global improvement by boosting the performance of the entity/token tagger, one research direction regards combining the output of distinct two-sided classifiers for improving the overall performance of NERQ-2S.

References

Areej Alasiry, Mark Levene, and Alexandra Poulovassilis. 2012a. Detecting candidate named entities in search queries. In *SIGIR*, pages 1049–1050.

Areej Alasiry, Mark Levene, and Alexandra Poulovassilis. 2012b. Extraction and evaluation of candidate named entities in search engine queries. In *WISE*, pages 483–496.

Junwu Du, Zhimin Zhang, Jun Yan, Yan Cui, and Zheng Chen. 2010. Using search session context for named entity recognition in query. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09*, page 267, New York, New York, USA. ACM Press.

Jian Hu, Gang Wang, Fred Lochovsky, Jian-Tao Sun, and Zheng Chen. 2009. Understanding users query intent with Wikipedia. In *Proceedings of WWW-09*.

A. Jain and Marco Pennacchiotti. 2010. Open entity extraction from web search query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 510–518.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January. Publisher: John Benjamins Publishing Company.

Marius Paşca. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07*, page 683, New York, New York, USA. ACM Press.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaoxin Yin and Sarthak Shah. 2010. Building Taxonomy of Web Search Intents for Name Entity Queries. In *Proceedings of WWW-2010*.