Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence

Named Entity Recognition in Travel-Related Search Queries

Brooke Cowan, Sven Zethelius, Brittany Luk,
Teodora Baras, Prachi Ukarde, and **Daodao Zhang**
Expedia, Inc., 333 108th Avenue NE, Bellevue, WA 98004
{brcowan, svenz, bluk, tbaras, pukarde, daozhang}@expedia.com

Abstract

This paper addresses the problem of named entity recognition (NER) in travel-related search queries. NER is an important step toward a richer understanding of user-generated inputs in information retrieval systems. NER in queries is challenging due to minimal context and few structural clues. NER in restricted-domain queries is useful in vertical search applications, for example following query classification in general search. This paper describes an efficient machine learning-based solution for the high-quality extraction of semantic entities from query inputs in a restricted-domain information retrieval setting. We apply a conditional random field (CRF) sequence model to travel-domain search queries and achieve high-accuracy results. Our approach yields an overall F1 score of 86.4% on a held-out test set, outperforming a baseline score of 82.0% on a CRF with standard features. The resulting NER classifier is currently in use in a real-life travel search engine.

1 Introduction

This paper investigates the application of named entity recognition (NER) to search-query processing in the travel domain. Given a query, the goal of the NER task described in this paper is two-fold: (1) segment the input into semantic chunks, and (2) classify each chunk into a predefined set of semantic classes. For example, given the query *hotel in mountain view with pool*, the desired output is *hotel in* [LOCATION *mountain view*] *with* [AMENITY *pool*], where the class AMENITY represents a travel-related amenity, and the class LOCATION is a geographic location or point of interest. Note that throughout this paper, we expand the canonical definition of *named entity* to include any domain-related semantic concept, such as an amenity.

NER in restricted-domain queries is potentially useful in many applications. For example, domain-specific NER could be used in general search following query classification to verticals such as product search, people search, or restaurant search. Alternatively, it could be used in a domain-specific search application such as a natural-language based travel search engine. Oftentimes, vertical search applications are backed by a database. For example, in travel search, the query *hotel in mountain view with pool* might generate a system response that contains an ordered set of relevant hotels stored as objects in a database. In such settings, a system that can identify relevant semantic entities (e.g., LOCATION = "mountain view", AMENITY = "pool") can map text-based queries to database objects, and thereby allowing the search engine to return more relevant results to the user.

NER and other NLP tasks are acknowledged to be challenging in search applications. The most successful NER solutions have been developed with well-edited text such as newswire (Ratinov and Roth 2009; McCallum and Li 2003; Collins and Singer 1999). In contrast, search queries are typically very short, e.g., 2–3 words (Spink et al. 2002), and offer little contextual evidence for the interpretation of terms. Furthermore, there is evidence that up to 70% of search queries form noun phrases (Barr, Jones, and Regelson 2008) as opposed to full sentences. When existing NLP tools trained on well-edited natural language text are directly applied to queries, there is a strong distributional mismatch between train and test data that results in poor performance. In the case of NER, state-of-the-art methods for extracting entities from well-edited text rely heavily on orthographic features such as capitalization. However, casing is acknowledged to be unreliable in queries (Rüd et al. 2011), thereby diminishing the utility of case-based features for NER in queries and posing additional challenges for robust entity extraction.

In this paper, we investigate the use of a linear-chain conditional random field (CRF) model (Lafferty, McCallum, and Pereira 2001) for the extraction of entities from limited-domain search queries. We show that this approach can achieve high-accuracy results when trained and tested on search queries from the target domain. CRFs and other probabilistic sequence models have proven to be highly effective for NER when trained and tested on well-edited text (Finkel, Grenager, and Manning 2005; McCallum and Li 2003). It has been argued that standard machine learning techniques such as sequence models are ill-suited to the semantic processing of queries (Bendersky, Croft, and Smith 2011; Manshadi and Li 2009; Guo et al. 2009; Shen et al. 2008). However, we show that a standard CRF trained and tested on in-domain data performs very well on queries. Our best CRF classifier achieves an overall entity-level F1 score of

86.4% on a held-out test set, outperforming a baseline score of 82.0% for a CRF with standard features.

The remainder of the paper is structured as follows. We discuss related work in Section 2. In Section 3, we motivate the need for a probabilistic approach to NER in the travel domain. Section 4 describes the CRF model and features, and the data sets we use for training and evaluation. In Section 5, we report experimental results, and we conclude the paper with contributions and future work in Section 6.

2 Related Work

In the past several years, search query processing has received increasing attention as a step toward the understanding of user-generated inputs to IR systems. The query processing literature includes a handful of papers that investigate solutions for NER in search queries. (Rüd et al. 2011) adapt a NE classifier trained on newswire to queries. Their method involves submission of query terms to a search engine to generate auxiliary textual evidence for the tagger. In contrast, our approach involves training a discriminative tagger directly on labeled query data. While we incur an offline cost in manual labor, we avoid the online cost of search-engine calls, making our solution more lightweight at test time, which is critical for our travel search engine.

(Guo et al. 2009) apply weakly-supervised Latent Dirichlet Allocation to query-based NER using partially-labeled seed entities. Similarly, (Paşca 2007) investigates a weakly-supervised method for extracting named entities from queries using a small number of seed instances per class. These approaches are similar to other work on semi-supervised methods for NE discovery (Riloff and Jones 1999), but instead of sentences from edited texts, the document base originates from query logs. An advantage of a weakly-supervised or semi-supervised approach compared to a supervised one is that the labeling effort is extremely limited. However, in general these approaches also require a much larger set of unlabeled data than we had access to for this work.

Finally, (Shen et al. 2008) design an algorithm for deciding whether a query is a personal name. Like ours, this work is an attempt to identify a class of named entities directly without supplemental calls to a search engine. The authors build probabilistic name-term dictionaries in an offline stage, and do probabilistic classification of queries using the dictionaries in an online stage.

Semantic tagging in queries is another research area closely related to this paper. The goal of the semantic tagging task is to classify each token in the input query into a predetermined set of classes. The main differences between this and NER are that (1) in semantic tagging, the relationship between two adjacent tokens with the same tag is not clear, and (2) the definition of the classes is generally looser than what is typically considered a named entity. (Li, Wang, and Acero 2009) experiment with semantic tagging of user queries in the product domain. They use a supervised CRF, as well as various semi-supervised CRFs that leverage labels derived from click stream data and product databases. The amount of supervised data used is more extensive than what we use in this paper. (Manshadi and Li 2009) address the semantic tagging task by inferring a semantic parse of web search queries using a PCFG that generates bags of words. A major drawback to this approach is its exponential time complexity. Though queries are generally short, the online cost would be prohibitively expensive for us.

In addition to work specifically on NER and semantic tagging, a few other efforts are worth mentioning. (Bendersky, Croft, and Smith 2011; 2010) propose a supervised approach to the joint annotation of capitalization, POS tags, and named entity segmentation in queries that employs search-engine features in the same vein as (Rüd et al. 2011). (Pantel and Fuxman 2011) associate query substrings directly with entities from a database. (Bergsma and Wang 2007) develop a probabilistic classifier for query segmentation using a hand-labeled data set.

Lastly, there are a few papers that elucidate the underlying linguistic structure of queries (Li 2010; Barr, Jones, and Regelson 2008), which may help to explain why standard machine learning models can in fact work well on queries. Indeed, there is a common assumption throughout the literature that standard approaches to NLP are not applicable to search-query processing. In this paper, we show that a standard NLP technique can be used successfully for NER in queries within in a restricted domain.

3 Motivation for Probabilistic NER in the Travel Domain

The goal of our team at Expedia is to implement a high-quality system for the identification and classification of travel-related entities in search queries. Our team develops and maintains a language understanding module that takes a user-generated query such as *hotel in mountain view with pool*, and produces a semantic frame (see Figure 1) that represents the meaning of the input with respect to the travel domain. The primary use case for the semantic frame is to provide a natural-language interface to travel search engines. The frame identifies important spans of text in the input and classifies them as one of several travel-related concepts, including `LOCATION`, `AMENITY`, `NAME`, `BRAND`, `STAR-RATING`, `PRICE`, `DATE`, and `DURATION`. Currently, our understanding module provides support for 14 distinct travel concepts. In addition to segmenting and classifying substrings of the input, it normalizes certain concepts: for example, it maps the string *4th of july* to a date/ time span in standardized UTC format, and it maps amenities to objects in Expedia database tables.

We have developed a probabilistic named entity recognizer for a subset (`LOCATION`, `NAME`, and `AMENITY`) of the travel entities we model. For these entities, the machine-learned approach is used in place of one based on a combination of pattern matching against domain-specific lists of entities, and heuristic disambiguation rules. The rules include, for example, manually-constructed lists of terms excluded from automatic membership in certain classes (e.g., *nice* is excluded from the `LOCATION` class to prevent it from being automatically tagged as a location in queries like *nice room in san francisco*). It also includes hand-written rules for resolving entity class ambiguity (e.g., *If a query sub-*

```
Query:     "hotel in mountain view with pool 4th of july"
Concepts: { Type:   HOTEL STRUCTURE
            Text:   "hotel"
            Span:   [0,0] }

          { Type:   LOCATION
            Text:   "mountain view"
            Span:   [2,3] }

          { Type:   AMENITY
            Text:   "pool"
            Span:   [5,5] }

          { Type:   RELATIVE DATE
            Text:   "4th of july"
            Span:   [6,8]
            Start:  2015-07-04T00:00:00
            End:    2015-07-05T00:00:00 }
```

Figure 1: Semantic frame produced for the query *hotel in mountain view with pool 4th of july*

string matches a term in the LOCATION *list and a term in the* NAME *list, then it is a* LOCATION). The heuristic pattern-matching approach is still used to model the other 11 concepts supported by the understanding module.

We targeted the LOCATION, AMENITY, and NAME classes – defined in Table 1 – for replacement with a probabilistic approach. These classes are highly salient to the domain and well-represented in our data sets. For example, in our development set, 83% of the queries contain a location, 30% contain a travel-related name or brand, and 3% contain an amenity. The correct identification of these classes is crucial to the selection of an appropriate set of responses by any downstream travel search engine. Additionally, in our experience, these three entity classes have proven difficult to model using regular-expression-based patterns. The heuristic rules we developed to handle them had complex interactions that were difficult to maintain for arbitrary natural language query inputs. In contrast, we have found other entity classes like dates and times to be much more amenable to modeling with patterns and rules. The probabilistic NER approach handles many difficult cases like (*hotel in* [**LOCATION** *mountain view*] vs *hotel with* [**AMENITY** *mountain view*]) that were difficult to handle with hard constraints.

4 Model, Features, and Data

We use a first-order, linear-chain CRF model (Lafferty, McCallum, and Pereira 2001) to implement NER in travel-domain queries. The CRF is an undirected graphical model that can be used to calculate the conditional probability of an output sequence $\mathbf{y} = \langle y_1, ..., y_T \rangle$ given an input sequence $\mathbf{x} = \langle x_1, ..., x_T \rangle$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}) \qquad (1)$$

Here, Z is a normalization factor over state sequences. For our NER task, the input $\mathbf{x}$ is a sequence of tokens in a travel-domain query. The output $\mathbf{y}$ is a sequence of labels that encode segmentation and class information (e.g., B-LOC to represent the beginning of a LOCATION entity). λ_k is the value of the learned weight associated with feature f_k. The weights in the parameter vector $\Lambda = \langle \lambda_1, ..., \lambda_K \rangle$ are set to maximize the conditional log-likelihood of the labeled sequences in a training set. There are a range of optimization procedures (conjugate gradient, quasi-Newton, etc.) that may be used to learn the weights.

Features depend on pairs of adjacent output tags at time t and $t-1$, as well as the entire input sequence $\mathbf{x}$. For example, a simple binary feature for our task might be defined as

$$f_7 = \begin{cases} 1 & \text{if } y_t = \text{B-AMEN}, y_{t-1} = \text{O}, x_t = \text{“spa”} \\ 0 & \text{otherwise} \end{cases}$$

This feature fires when the current class label is the beginning of an amenity, the previous class label is not part of any named entity, and the current token is *spa*. This feature should have a positive weight since the token *spa* is likely to be an amenity as long as the previous token is not part of a name (e.g., *The Webbington Hotel and Spa*). A negative-weight feature would look like

$$f_{56} = \begin{cases} 1 & \text{if } y_t = \text{B-AMEN}, y_{t-1} = \text{B-LOC}, x_t = \text{“city”} \\ 0 & \text{otherwise} \end{cases}$$

since it's far more likely for the token *city* to be part of a city name than an amenity when the preceding token is the beginning of a location. The ability of a CRF to incorporate arbitrary features of the input and output is one of its most appealing properties. Another highly appealing property of a linear-chain CRF is that inference can be done efficiently using dynamic programming.

We adapt the Stanford NLP Toolkit's CRF implementation (Finkel, Grenager, and Manning 2005) for our NER system and use a subset of the provided feature templates to define our baseline. Table 2 contains a comprehensive list of our experimental feature templates, grouped into sets. The BASE set contains our baseline features. Though standard for formal, edited text, we do not include word shape features in any of our feature sets since case-based features are unsuited to our data. The POS set contains part-of-speech tag features; WC contains word cluster features; and GAZ contains gazetteer features.

To our knowledge, there is no publicly-available labeled data set for NER in the travel domain. We developed our own data sets by extracting and annotating 3,500 random examples from user logs. Our data sets are available to the public for research purposes by request to the authors. Figure 2 shows a few tagged examples from our training set. The queries originated from two sources:

1. External search engines: Queries entered into various external search engines by users who ultimately completed a transaction on our site.

ENTITY	DESCRIPTION	EXAMPLES
LOCATION	geo-political entities, points of interest, airports, locational modifiers	*puerto rico*, *french riviera*, *space needle*, *pdx*, *o'hare*, *beach*, *downtown*
NAME	hotel names and brands, car vendors, and airlines	*howard johnson*, *jet blue*, *hawksbill beach resort all inclusive*, *budget*
AMENITY	modifiers and descriptors of traveler intent	*non-smoking*, *family friendly*, *swimming pool*, *all inclusive*, *ski*, *romantic*, *beach*

Table 1: The three entity classes for the travel domain targeted for probabilistic NER. There is often ambiguity between classes, for example *beach* can be LOCATION, AMENITY, or part of a NAME. *budget* can be a name or outside of any entity, for example, *tokyo budget hotel*. In the latter case, *budget* would be identified as a PRICE by the heuristic pattern-match recognition system.

SET	FEATURE	DESCRIPTION
BASE	WORD	curr word
	#	prefix/suffix substrings of len 6
	DISJP	each of the prev 4 words
	DISJN	each of the next 4 words
	W-PW	curr word + prev word
	W-NW	curr word + next word
	PSEQ	prev class
	PSEQW	curr word + prev class
	PSEQW2	prev word + curr word + prev class
	PSEQpW	prev word + prev class
POS	TAG	POS-tag(curr word)
	PTAG	POS-tag(prev word)
	NTAG	POS-tag(next word)
WC	DISTSIM	word-cluster(curr word)
	PDISTSIM	word-cluster(prev word)
	NDISTSIM	word-cluster(next word)
	PSEQcDS	word-cluster(curr word) + prev class
	PSEQpDS	word-cluster(prev word) + prev class
	PSEQpcDS	word-cluster(prev word) + word-cluster(curr word) + prev class
GAZ	GAZc	curr word matches a gazette entry
	GAZpC	curr word matches a gazette entry + prev class
	GAZ#c	curr word matches a gazette entry of len #
	GAZ#pC	curr word matches a gazette entry of len # + prev class

Table 2: Baseline set of feature templates from the Stanford NLP Toolkit (BASE) and experimental feature templates: POS = part of speech tag features, WC = word cluster features, GAZ = gazette features. Each feature is also conjoined with the value of the current class. A bias feature containing the current class alone is not listed but is also present in the baseline set of feature templates.

Data set	Location	Name	Amenity
Train	81%	28%	25%
Dev	83%	30%	3%

Table 3: Percentage of queries containing each of the entity classes in the development and training sets.

2. Internal search engine: Queries entered into an internal travel search engine used by company employees.

700 training examples, 500 development examples, and 300 test examples were randomly selected from around 120K external search engine queries from June 2013. Another 700 training examples, 300 development examples, and 300 test examples were randomly selected from around 170K internal search engine queries from August through November 2013. The method for acquiring the remaining 700 training examples is described below.

One challenge we faced when creating the data sets was the low representation of queries containing amenities. For example, in our development set, 83% of the queries contain a location, 30% contain a travel-related name or brand, and only 3% contain an amenity. To address this problem, we used a semi-supervised NER engine based on the work in (Riloff and Jones 1999) to discover queries containing amenities in an auxiliary set of 100K external search engine queries. We started with a seed set containing the 232 amenities in our amenity gazette, and ran the semi-supervised engine for one full iteration (i.e., find examples containing seeds, extract patterns, find new exemplars). We used 700 queries obtained in this manner to supplement our training data. As a result, the distribution of classes in our training set differs from that in our development and test sets (see Table 3).[1]

Prior to annotation, we performed basic tokenization, and we lower-cased all of the data due to inconsistency in casing. Four of the authors wrote annotation guidelines and tagged the data. Annotators marked any ambiguous or uninterpretable queries or parts of queries with the special tag "UNK" for *unknown*. These queries were subsequently eliminated from the data sets. There are 2,084 queries (8,720 tokens) in the resulting training set, 790 queries (2,924 tokens) in the dev set, and 589 queries (2,177 tokens) in the test set. On average, there are 1.8 entities per query in the training set (1.6 in dev). The average number of tokens per query in the training set is 4.2 (3.7 in dev). To obtain a measure of inter-annotator agreement, one of the annotators tagged 200 examples originally tagged by two different annotators. The resulting agreement was 91.9% using Cohen's κ statistic.

[1] Empirically, we have found that classifiers trained on this data do not overpredict amenities.

downstream|B-name *casino*|L-name *oklahoma*|U-loc
maui|U-loc *vacation*|O *packages*|O
london|U-loc *hotels*|O
tachi|B-name *palace*|I-name *hotel*|I-name *and*|I-name *casino*|L-name ,|O *california*|U-loc
dreams|B-name *riviera*|I-name *cancun*|L-name
cheap|O *hotels*|O
san|B-loc *diego*|L-loc ,|O *seaworld*|U-loc
nyc|U-loc *to*|O *burlngton*|U-loc ,|O *vt*|U-loc
cape|B-loc *may*|I-loc *courthouse*|L-loc *motels*|O
hotels|O *green*|B-loc *river*|L-loc *utah*|U-loc
saint|B-loc *martin*|L-loc *all*|B-amen *inclusive*|L-amen *resorts*|O
kyoto|U-loc *hotels*|O
hotels|O *near*|O *pratunam*|B-loc *market*|L-loc
car|O *rental*|O
comfort|B-name *suites*|I-name *at*|I-name *royal*|I-name *ridges*|L-name

Figure 2: Tagged examples from our training set.

5 Experiments

We use the data sets described in Section 4 for our experiments. To evaluate each recognizer's ability to identify entities in travel queries, we report the F1 score. We use the standard definitions of precision and recall, and we compute F1 as their harmonic mean. All metrics are reported at the entity level as opposed to the conventional CoNLL phrase-level reporting (Sang and Meulder 2003). We use the BIO (**B**eginning **I**nside **O**utside) representation scheme in all of our reported experiments. Despite evidence to the contrary (Ratinov and Roth 2009), we found empirically that on our data sets, a BIO-tag classifier produced higher scores than BILOU, which adds two additional prefixes to represent the **L**ast token in an entity and **U**nit-length entities. Presumably, our BILOU tagset was too large relative to the size of the training set, leading to sparse statistics, though we have not tested this hypothesis. For optimization of the CRF weights, we use the quasi-Newton implementation in the Stanford toolkit with the default settings.

Table 4 shows experimental results using several feature sets on both development and test data. We report the overall F1 score as well as the F1 score of each entity class. The highest score in each column is shown in bold face. Scores that are statistically different from the baseline at the 0.01 significance level, according to the sign test, are marked with an asterisk. The first row shows results with our baseline set of features as defined in Table 2. Subsequent rows add various combinations of feature sets on top of the baseline. The feature sets are described in detail in Table 2. In order to provide part-of-speech tags for the POS features, we run our data sets through an in-house, domain-specific POS tagger. The tagger is a standard maximum entropy sequence model tagger trained with hand-labeled, in-domain data that achieves a measured tag-level accuracy of 94.0%. For the word cluster features (WC), we use Brown clustering (Brown et al. 1992) to induce word clusters from a set of 231K unlabeled user queries entered into our company's internal travel search engine. In particular, we use the implementation of Brown clustering described in (Liang 2005) and set the number of clusters to 10.[2] For the GAZ features, we use proprietary lists of domain-specific entities. Though neither comprehensive nor noise-free, they offer good coverage: there are around 1K airline names, 120 car vendors, 200 hotel attributes, 900 hotel brands, 200K hotel names, and 175K locations.

In addition to various feature sets, we experiment with several hybrid models that make use of heuristic information (HEUR) from the pattern-matching NER system described in Section 3. The hybrid models are shown in the last four rows in Table 4. The primary source of heuristics are manually-constructed disambiguation rules. Crucially, whereas in previous instantiations of our system these heuristics were applied as hard constraints, the discriminative CRF allows us to easily incorporate them as soft constraints by including them as features.

Finally, we investigate the use of aliases, abbreviations, and spell correction (AAS). Our manually-curated lists of aliases, created during the development of the pattern-matching NER system, are generally common misspellings in the domain data (like *marriot* instead of *marriott*). Manually-curated abbreviations include, for example, *dtwn* for *downtown*. Spelling correction is carried out by a domain-specific, probabilistic classifier developed by our group. These three components – aliases, abbreviations, and spelling – are considered together in Table 4 since the best performance on the development set arose when using them in conjunction. Aliases and abbreviations are added to the gazettes prior to the extraction of features, and spelling correction is applied as a preprocessing step.

Table 4 clearly shows that we are able to improve on a strong baseline for NER in the travel domain. Overall, we achieve an F1 score of 86.4% on our test set, up from a baseline score of 82.0%. We see a similar trend on the development set. Our highest-performing classifiers on both evaluation sets are those that make use of information from a variety of sources, some automatically generated and some manually curated. That said, it is interesting to note that the boost due to manually-curated information over the best-performing classifier without heuristics (86.1% to 86.4%) is not statistically significant on the test set. In fact, it appears that the addition of extensive in-domain gazette information together with word clusters derived from a moderate amount of unsupervised data (WC GAZ in the table) gener-

[2] We used the software implementation of Brown clustering available on Liang's website.

	Development				Test			
Model	**ALL**	**LOC**	**NAME**	**AMEN**	**ALL**	**LOC**	**NAME**	**AMEN**
BASE	81.5	84.1	68.1	85.7	82.0	85.5	64.7	73.1
POS	81.9	84.5	68.8	84.2	81.4	85.0	63.6	74.5
WC	84.7*	87.3*	72.7	85.7	83.2	86.7	66.7	74.5
GAZ	84.6*	87.5*	71.6	82.1	84.2	87.8	66.0	73.5
WC GAZ	84.7*	87.6*	72.5	80.7	86.0*	**89.8***	68.5	72.0
WC GAZ POS	85.2*	88.1*	72.8	80.7	86.1*	89.4*	70.2	76.0
WC GAZ HEUR	86.3*	89.1*	74.2	84.8	86.0*	89.6*	69.6	70.6
WC GAZ HEUR AAS	**86.9***	**89.6***	75.1	83.3	85.5*	88.8	70.5	70.6
WC GAZ POS HEUR	86.4*	89.0*	75.2	**86.7**	86.0*	89.2*	70.4	**78.4**
WC GAZ POS HEUR AAS	86.4*	89.3*	**76.2**	85.3	**86.4***	89.3*	**72.4**	**78.4**

Table 4: Results on development and test data. Scores in bold face are the highest achieved for the column. Scores that are statistically different from the baseline at the 0.01 significance level are marked with an asterisk.

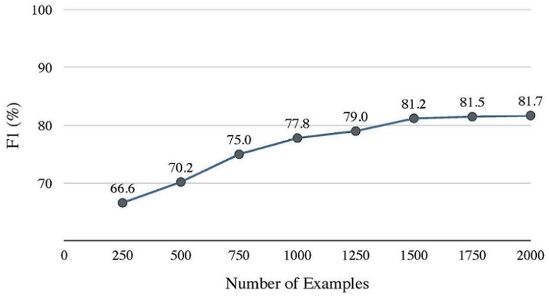

Figure 3: F1 as a function of the size of the training set.

ates the best performance with the least amount of manual effort. Notably, the inclusion of a domain-specific POS tagger does not appear to improve F1 in a statistically significant way. As a general trend, the performance of the various classifiers in the recovery of NAME entities tends to be lower than for LOCATIONs and AMENITYs. We have noticed that NAMEs tend to be longer than the other entity types and are less likely to match gazette entries in the NAME gazette due to aliasing. For example, the query *aegean international hotel* will not match the corresponding entry *aegean intl hotel hot spring & spa* in the NAME gazette. The requirement of an exact gazette match diminishes the value of the NAME gazette and may be too strict. We leave further investigation of this problem to future work.

Because our training set is relatively small, it is natural to wonder how much improvement we might see by adding more examples. The learning curve in Figure 3 shows that at around 2,000 training examples, the baseline classifier's performance gain starts to taper off. This suggests that in order to get equal gains in performance, we would need to supplement our training set with increasingly larger amounts of data. We leave the addition of supplemental data to future work as well.

6 Contributions and Future Work

In this paper, we have demonstrated the efficacy of applying a probabilistic sequence classifier to NER for travel-domain queries. We have shown that we can achieve high-quality extraction by labeling a relatively small amount of data and using features derived from in-domain dictionaries and word clusters, without reliance on clues from capitalization. Furthermore, our results show that inclusion of gazette and word cluster features in addition to a strong baseline set of features diminish dependence on labor-intensive heuristic rules developed over several years of work on a rule-based pattern-matching system. The resulting probabilistic NER classifier is both highly accurate and efficient, and has been successfully embedded in a real-life travel search engine.

There are several paths for additional work that we intend to address in the future. Investigating ways to target improved performance for the NAME entity class is a high priority. We plan to investigate the inclusion of phrase-based cluster features to help with longer NAME entities, and to look into methods for better gazette matching. For example, we would like to make use of a large list of search-engine validated aliases for hotel names. Another important area for future research is to evaluate the use of additional training data. The size of our training set is small, and our experiments suggest we may see additional gains with more examples. Moreover, as our language understanding module and the travel search engine in which it is embedded move into the public realm, we anticipate a need for acquiring fresh labeled data sets that don't require as much manual effort on our behalf. For these reasons, we are looking into ways of generating labeled training sets using crowdsourcing.

7 Acknowledgments

The authors would like to thank the Unconstrained Search team at Expedia for their support in the writing of this paper. Special thanks to I-Chiang Chen and Taimur Khan for their help, and to Ondrej Linda, Ang Sun, and the anonymous IAAI reviewers for their useful comments and suggestions.

References

Barr, C.; Jones, R.; and Regelson, M. 2008. The linguistic structure of english web-search queries. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 1021–1030.

Bendersky, M.; Croft, W. B.; and Smith, D. A. 2010. Structural annotation of search queries using pseudo-relevance feedback. In *CIKM'10*.

Bendersky, M.; Croft, W. B.; and Smith, D. A. 2011. Joint annotation of search queries. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 102–111.

Bergsma, S., and Wang, Q. I. 2007. Learning noun phrase query segmentation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 819–826.

Brown, P. F.; deSouza, P. V.; Mercer, R. L.; Pietra, V. J. D.; and Lai, J. C. 1992. Class-based n-gram models of natural language. *Journal of Computational Linguistics* 18(4):467–479.

Collins, M., and Singer, Y. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 100–110.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meetings of the Association for Computational Linguistics*, 363–370.

Guo, J.; Xu, G.; Cheng, X.; and Li, H. 2009. Named entity recognition in query. In *Proceedings of the 32nd International ACM SIGIR conference on Research and Development in Information Retrieval*, 267–274.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*.

Li, X.; Wang, Y.-Y.; and Acero, A. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 572–579.

Li, X. 2010. Understanding the semantic structure of noun phrase queries. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1337–1345.

Liang, P. 2005. Semi-supervised learning for natural language. Masters of engineering, Massachusetts Institute of Technology, Cambridge, MA.

Manshadi, M., and Li, X. 2009. Semantic tagging of web search queries. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 861–869.

McCallum, A., and Li, W. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th Conference on Natural Language Learning*.

Paşca, M. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 683–690.

Pantel, P., and Fuxman, A. 2011. Jigs and lures: Associating web queries with structured entities. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 83–92.

Ratinov, L., and Roth, D. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, 147–155.

Riloff, E., and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence*, 474–479. AAAI.

Rüd, S.; Ciaramita, M.; Müller, J.; and Schütze, H. 2011. Piggyback: Using search engines for robust cross-domain named entity recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 965–975.

Sang, E. F. T. K., and Meulder, F. D. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL*, 142–147.

Shen, D.; Walker, T.; Zheng, Z.; Yang, Q.; and Li, Y. 2008. Personal name classification in web queries. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, 149–158.

Spink, A.; Jansen, B. J.; Wolfram, D.; and Saracevic, T. 2002. From e-sex to e-commerce: Web search changes. *IEEE Computer* 35(3):107–109.