Leveraging Web Query Logs to Learn User Intent Via Bayesian Discrete Latent Variable Model

Asli Celikyilmaz ASLI@IEEE.ORG
Microsoft Speech Labs, Mountain View, CA 94010 USA

Dilek Hakkani-Tür DILEK@IEEE.ORG
Microsoft Speech Labs | Microsoft Research, Mountain View, CA 94010 USA

Gokhan Tür GOKHANT@IEEE.ORG
Microsoft Speech Labs | Microsoft Research, Mountain View, CA 94010 USA

Abstract

A key task in Spoken Language Understanding (SLU) is interpreting user intentions from speech utterances. This task is considered to be a classification problem with the goal of categorizing a given speech utterance into one of many semantic intent classes. Due to substantial utterance var, significant quantity of labeled utterances is needed to build robust intent detection systems. In this paper, we approach intent detection as a two-stage semi-supervised learning problem, which utilizes a large number of unlabeled queries collected from internet seach engine click logs. We first capture the underlying structure of the user queries using bayesian latent feature model. We then propagate this structure onto the unlabeled queries to obtain quality training data via a graph summarization algorithm. Our approach improves intent detection compared to comparison to our baseline, which uses a standard classification model with actual features.

1. Introduction

Building Spoken Language Understanding (SLU) systems to understand natural language queries has long been a topic of interest in the research community (Tür & Mori, 2011). In this study, we focus on Spoken Language Underestanding (SLU) intent detection in order to classify natural language utterances into one or more of the previously defined semantic classes, the *intents*. Thus, intent detection is generally considered as a multi-class classification task. As a running example, a natural language utterance from a human-machine dialogue *"show me the nearest theaters in palo alto"* is classified as belonging to the *find theaters* intent by the SLU intent determiner.

Semantic classifiers like intent determination require operations that allow significant utterance variability. For example, although the utterance *"where is inception playing"*, has the same semantic intent, i.e., *find theater*, as the running example above, they have no lexical overlap. This makes the task intrinsically challenging in that, not only are there no *a priori* constraints on what the user might say, the system must also be able to generalize from a tactably small amount of training data. A useful resource for this task is information from search queries, because they directly reflect users' intents. Furthermore, most users click on URLs returned by a search engine related to their query, proving an implicit supervision abut the broad category of the query.

Lately, with the increased use of web search, user search query logs have become a valuable source of unlabeled information for improving understanding of user queries (Li et al., 2008; Anastasokos, 2009; Li, 2010). Previous research has shown that mining query logs improves numerous search engine functions, including query suggestion, classification, ranking, targeting advertisement, etc. A common method is to represent query click (Q-C) data as a click-graph, namely a bipartite-graph representation of click-through data (Figure 1). The edges therein connect queries and URLs (or clustered URLs) and are weighted by the associated click counts. A click graph

Appearing in *Proceedings of the 28^{th} International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

contains a vast amount of user click information, giving rise to opportunities for semi-supervised learning, which leverages both labeled and unlabeled examples at training time.

In this paper, we utilize query logs for a different purpose. Intuitively, since queries with similar click patterns should be semantically similar, we mine query logs to improve SLU intent detection performance. We address some of the issues with using a high dimensional Q-C dataset in understanding user intent and focus on a factor analysis approach for feature generation. Our aim is to discover the latent process underlying the Q-C data and take it as a relational matrix of queries defined as samples using the click data as multi-dimensional features. Since we extract this relational data from unstructured query logs, we assume that the process representing the true dimensionality of the user queries is not known. We utilize an unbounded factor analysis approach and build an infinite dimensional latent factor analysis, namely the the Indian Buffet Process (IBP) (Griffiths & Ghahramani, 2005), specifically to model the latent factor structure of the given set of queries. In particular, we treat the user-query-to-factor relationship non-parametrically.

We deal with a large number of unlabeled queries and would like to use them at training time to obtain a quality labeled data. Since labeled large number of unlabeled data is costy and time consuming, here we present our new algorithm for extracting additional quality labeled data (representative matrix) using unlabeled web query logs based on the latent factors extracted using IBP. We use a small set of labeled training data represented as latent factors, and implement a graph summarization (similar to the work in (Celikyilmaz et al., 2009)) to obtain a representative labeled matrix from unlabeled queries. In the experiments we show performance improvements over a standard method using actual url clicks as features.

2. Mining Query Logs for Intent Detection

Query click (Q-C) data is a log of unstructured text including both the users queries sent to a search engine and the links that the users substantially clicked on from the list of sites returned by that search engine. A common representation of such data is a query-click graph as shown in (Figure 1). Traditionally, the edge of the click graph is weighted based on the raw click frequency (number of clicks) from a query to a URL. Some of the challenges in extracting useful information from Q-C's is that the feature space is high dimensional (there are thousands of url clicks linked to many queries) and there are millions of queries logged daily. Extracting queries and related information for a specific domain, such as certain events, disasters, or regions is a challenging task that requires machine learning, especially for non-parametric approaches. Since it is very costly to label such log data, which is also very noisy, it is advantageous to label a portion of the data and then to use semi-supervised learning methods to obtain a quality training data set.

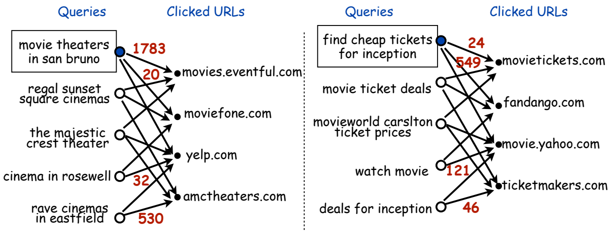

Figure 1. A sample query click graph. The squared queries are samples from training dataset which are natural language utterances.

Previous reserach has shown the advantages of utilizing query click logs to improve the query intent classification task. Li *et.al* (Li et al., 2008) use query click logs to determine the intent of the web search queries and then to infer the class membership of unlabeled queries from those of the labeled queries. They define a bipartite graph using the correspondence between the queries and the clicked URLs and then use a label propagation method to transfer labels from the queries to the URLs and then to other queries. Hakkani-Tür *et.al* (Hakkani-Tür et al., 2011) extend previous work by using the lexical content of queries, where labels are propagated according to the lexical similarity of the search queries to natural language utterances. Furthermore they integrate noisy click labels asa feature for label propogation model.

In our task we implement a graph summarization algorithm to capture representative queries from a large set of unlabeled queries that are similar to a rather smaller set of labeled queries. Rather than representing the queries as a high-dimensional vector of click frequencies, we follow a two stage process. As a preprocessing step, we capture the latent factor structure the labeled queries via IBP and reduce the dimensionality of the queries to manageable size and then infer the factor structure of the large number of unlabeled queries. As a second step, using graph summarization we compile additional queries in this latent factor space and construct a representative matrix in order to produce quality training data. In the next section we give details of each step.

2.1. Indian Buffet Process- IBP

We discover the underlying semantic relationships between the user search queries by *decomposing* the click url frequency features into K latent factors using the IBP model (Griffiths & Ghahramani, 2005). IBP defines a distribution over binary matrices with an infinite number of columns, only a finite number of which contain non-zero entries (the factors). It has a scale parameter α that defines the sparsity of the factor space.

Best represented by using a culinary methapor, customers (queries) enter a restaurant and select from an infinite number of dishes (factors) arranged in line. The first customer selects a dish k proportional to m_k, the number of times that dish has been previously sampled by prior customers, in other words, according to its popularity. When the customer has sampled all the previously sampled dishes, he samples an additional $Poisson(\alpha)$ dish that has never been sampled before. When all the customers have finished sampling dishes, the resulting binary matrix Z becomes a draw from IBP(α), where z_{mk} represents whether the binary factor is present or not. To construct a finite latent factor model, each binary variable z_{mk} is assumed to be drawn from the following two stage generative process:

$$\begin{gathered} \pi_k \sim Beta(\alpha/K, 1), \\ z_{mk} \sim Bernoulli(\pi_k) \end{gathered} \tag{1}$$

where π_k is the probability of observing the feature k.

Teh and colleagues (Teh et al., 2007) showed that taking the limit of this model as $K \rightarrow \infty$ yields an IBP latent factor model and that we can obtain a strictly decreasing ordering of the latent probabilities π_k by starting with a "stick" of unit length and recursively breaking it at a point $beta(\alpha, 1)$ along its length, discarding the excess, for $k = 1, 2, ...$:

$$\begin{gathered} \mu_k \sim Beta(\alpha, 1), \\ \pi_k = \prod_{j=1}^{k} \mu_j \\ z_{mk} \sim Bernoulli(\pi_k) \end{gathered} \tag{2}$$

In recent work, IBP has been used to extract unobserved (latent) transciption factors to uncover patterns in gene expression data (Beal et al., 2005; Rai & DaumeIII, 2008). In this study, we use IBP similarly, that is, to discover the latent structure of the clicked queries in order to understand a user intent based on the relationship between the queries a user types and the web links he clicks.

Table 1. Sample user queries on movie domain and semantic intent classes.

Query	Intent
"showtimes for up"	FIND SHOWTIME
"avatar"	FIND MOVIE
"palo alto cinemas"	FIND THEATER

2.2. Nonparametric Query-Log-Factor Model

Our aim in building a nonparametric query-log-factor model is to collect queries from web click logs to compile quality training data by means of factor analysis. We compile a set of natural language utterances U_L from the movie domain. Each utterance is manually labeled with a semantic intent. Some examples are shown in Table 1. For each natural language utterance we mine the click logs to extract the frequencies of user clicks on given URLs, namely we compile a Q-C matrix for the training utterances. The list of unique URLs clicked by users forms the *url-vocabulary.*

To mine additional unlabeled queries from the click logs, we manually formulate a list of a few highly related URLs, e.g, for movies domain we use imdb.com, fandango.com, movie.tickets.com.etc. The click logs contains queries not only directed to these domain specific urls but also many other urls that the users have clicked on. We compile a noisy dataset first and then use the refined url vocabulary extracted from the training dataset to do a reverse mining. In other words, we compile a set of unlabeled queries Q_U from the click logs that the user's have clicked on at least once and filter the URLs based on our vocabulary. The idea here is to capture queries with similar semantic intents using click relations. For example when the two queries *"reviews for up"* and *"up critics"* both result in a click on the site "imdb.com", it is likely that the intent of these queries is *"review/rate movies"*. The same would be true for clicks going to "fandango.com", which allow users to purchase movie tickets.

Since both compiled labeled and unlabeled Q-C data, U_L and Q_U, is higher dimensional (there are thousands of URLs extracted as features) and especially since the Q_U contains a large number of unlabeled queries, we first need to reduce the dimensionality of the data and do some down-sampling for effective training and inference. We start by using IBP on the training matrix U_L to discover the latent factors Z_L, a matrix of binary latent features. Then we use the rest of the unlabeled queries Q_U as testing data and predict their binary latent factor matrix Z_U, accordingly. As a result, we represent our data with rather reduced dimension of binary factors instead of large dimensions of actual features. In the next section, we show our graph summa-

Algorithm 1 Graph Summarization

```
Input: Labeled data $Z^L = \{x_i^L, y_i\}$, of size $n^L$ and unlabeled data $Z^U = \{x_j^U\}$, of size $n^U$.
for $i = 1$ to $n^L$ do
    ⋆ Select M unlabeled queries $x_j^U$ from $Z_U$ that are closest to $x_i^L$ based on (3),
    ⋆ Calculate a representative vector $\bar{x}_i^r$ from the selected M queries using (4).
    ⋆ Label the representative vector $\{\bar{x}_i^r, y_i\}$ with the same label as $x_i^L$ and append to the labeled dataset $Z^L$.
end for
```

rization approach to extracting representative labeled queries from unlabeled queries.

2.3. Graph Summarization for Representative Query Extraction

Our approach to summarization is to compile representative data vectors from unlabeled queries and then to propagate the output labels of the training vectors to obtain additional labeled data (a common method in semi-supervised learning). Suffice it to say that similar data points are likely to represent denser regions in the hyper-space (graph) and are therefore likely to have similar labels. If the points are close enough, we can even characterize a group of similar data points with respect to the graph and then capture their summary information with new *representative vertices*. Algorithm 1 shows the graph summarization process for data collection.

We start with the binary latent matrices from the querylog-factor model. We have a large number of unlabeled data vectors $x_j^U \in Z^U, j = 1, ..n^U$ and a rather small number of labeled data vectors, each represented as low-dimensional binary latent features, $x_i^L \in Z^L, i = 1, ..n^L$ with corresponding labels y_i. We represent each vector as a node in a graph $g = (V, E)$, where E is the weighted edge between the nodes and is measured via:

$$w_{ij}(x_i^L, x_j^U) = 1 - \sum_{f=1}^{F} \frac{|x_{if}^L - x_{jf}^U|}{F} \tag{3}$$

where F is the total number of latent factors. The more similar the vectors, the larger the edge weights would be. For each labeled vector in the training dataset x_i^L, we choose M similar unlabeled queries and find a representative vector via:

$$\bar{x}_i^r = \frac{\sum_{i \neq j=1}^{M} \frac{1}{2} w_{ij}(x_i^L + x_j^U)}{\sum_{i \neq j=1}^{M} w_{ij}} \tag{4}$$

Each new representative data vector $\bar{x}_i^r$ gets the label of the represented training utterance x_i^L. The new dataset is represented as a low dimensional binary matrix containing labeled utterances and representative queries which are also labeled via graph summarization. In the experiments, we show the performance improvements in the testing data when using the latent binary factors from IBP extracted from high-dimensional unstructured query logs. In addition we discuss the effects of dataset expansion using graph summarization on the SLU intent detection task.

3. Empirical Study

Our task is to build an intent classification model for an SLU system utilizing unlabeled queries from click logs. We demonstrate the effects of using IBP with graph summarization on intent detection using data related to movie domain.

3.1. Data:

We compiled and annotated around 1200 natural language utterances for the movie domain and identified 7 semantic intent classes, i.e., *find movie, find theater, find showtime, buy ticket, review movies, review theaters, get trailer*, where sample queries are shown in Table 1. For the training utterances, we mined click logs for click-frequency to compile a training Q-C dataset and captured around 1500 unique clicked URLs, the training url vocabulary, forming our actual feature set for the movie domain. We then reverse mined close to 120K queries from the logs using a user defined set of domain urls such as imdb.com, fandango.com, yahoo.movies.com, etc., to compile another Q-C data of large number of unlabeled queries with a high dimension of clicked urls. We only used the clicks that match the training url vocabulary to match the number of dimensions in the unlabled Q-C data and the training Q-C data. We compiled another Q-C matrix for around 200 testing utterances using the url vocabulary as features. Since these utterances are mostly natural language like *"find me 2009 movies by James Cameron"* rather than keyword search queries like *"2009 James Cameron movies"*, almost 30% of the queries, both in the training and test data, did not exist in query click logs.

3.2. The Models:

For the IBP, we used $\gamma \sim Gamma(5, 0.1)$ fixing $\alpha = 5$. In graph summarization, we fixed $M = 100$ closest queries to calculate representative data vectors.

First we developed the IBP model on the training Q-

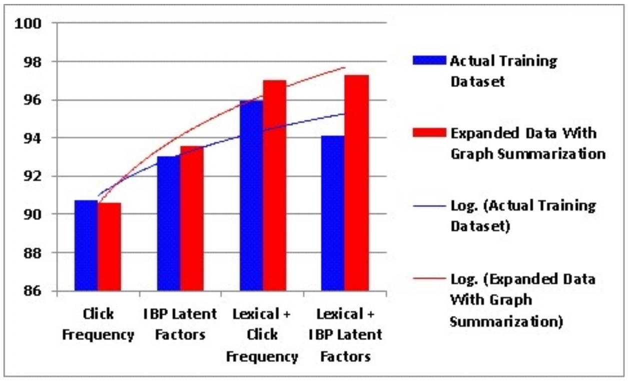

Figure 2. Classification accuracies obtained from movie domain test data. Four sets of feature combinations are shown on the x-axis. We compare the test accuracies of models trained on (*i*) only labeled training data (shown in blue) and (*ii*) expanded labeled data via representative queries obtained from graph summarization method (shown in red columns). The trendline indicates the performance improvement with the use of representative queries at training time with different feature sets.

C data to capture the latent factors. Then, using the Q-C data of unlabeled queries as test, we captured their latent factor representations. Using *graph summarization* we calculated one representative query vector from unlabeled queries corresponding for each labeled training utterance. With the representative data points, we obtained an expanded Q-C training matrix.

Using the Q-C expanded dataset, we build an intent detection classification model based on icsiboost [1], an implementation of AdaBoost.MH algorithm (Schapire & Singer, 2000). We build several models using different feature set combinations including actual query click frequencies of each URL, the latent factors extracted via IBP to analyze their individual effects. We also combine lexical features extracted from training queries, i.e., the word unigrams, bigrams and trigrams together with the latter features to anayze their joint effects. Each individual feature set is depicted in Figure 2.

3.3. Experiment Results

In Figure 2 we show the results obtained from the movie domain dataset. When only the training dataset is used without unlabeled queries via graph summarization the latent factors extracted via IBP outperforms the model which uses the actual click frequency features. This result is particulary interesting in the sense that when the predicted factors from IBP is used in the classification model for intent detection task compared to the actual click frequncies, we get better performance. This indicates that one can use predicted latent factors as features instead of high dimensional feature sets without loss of generality. This is in part due to the fact that the high dimensional click frequencies are highly likely correlated, affecting the performance of the classifier model. On the other hand, when we use lexical features obtained from training dataset as additional features, the intent detection performance improves with both the IBP and actual feature models. However this time the difference between using the factors or actual features is less compared to before.

On the other hand, the post-process classification model benefits from low dimensional representation of the data vectors in terms of reduced complexity measured in traning time. As shown in Figure 3, the graph summarization to label queries from click graphs takes considerably less time when the latent factors are used as features. Specifically summarizing 120K click queries takes around 30 hours when the actual click frequencies are used as features in comparison to the IBP latent factor features which takes couple of hours to obtain representative matrices. Although not shown in the graph, a similar learning time reduction pattern is also observed with the boosting classification model building between using the latent factors as features and actual features.

Looking at both tables it can be drived that the latent factors are better alternatives to building an intent detection model in terms performance improvement and complexity reduction for building intent detection. It should also be pointed out that the algorithm presented here is intented to be implemented for an online application where the utterances (testing samples) are received online and goes through the framework being discussed in this paper. Hence, since at training time we do not have access to the testing data, we cannot find the latent features for both the training and testing data.

4. Conclusions and Future Work

We presented a framework that utilizes user search query logs in oder to compile a quality trianing dataset for building an intent detection model in spoken language understanding. We showed advantages of using a non-parametric binary latent factor model to capture the true structure of high dimensional and noisy click log data on the intent detection. We showed in the experiments the benefits of using a low dimensional data

[1] http://code.google.com/p/icsiboost

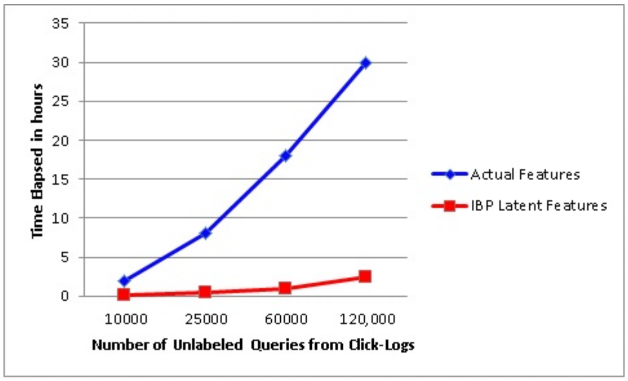

Figure 3. The training elapsed time in hours with respect to the the number of unlabled queries from query-click logs used to calculate representative data points via graph-summarization.

representation in terms of training data expansion and model complexity reduction.

As a future work, we would like to expand the experiments to other domains that interests SLU systems. We want to incorporate some supervision to the latent factor analysis in order to collect in domain queries represented as high dimensional data. With this approach we would like to discover correlation between related clicked urls and user intents as a prior information for latent factor analysis and build a semi-latent factor model.

Acknowledgments

The authors would like to thank Dr. Ashley Fidler for her constructive comments in preparation of the paper.

References

Anastasokos, T. Filtering approach to ad recommendation using the query ad click graph. In *Proceedings of the CIKM*, 2009.

Beal, M.J., Falcani, F., Ghahramani, Z., Ragel, C., and wild, D.L. A bayesian approach to reconstructing genetic regulatory network with hidden factors. In *Proceedings of the NIPS*, 2005.

Celikyilmaz, A., Thint, M., and Huang, Z. A graph-based semi-supervised learning for question-answering. In *Proceedings of the ACL*, 2009.

Griffiths, T.L. and Ghahramani, Z. Infinite latent feature models and the indian buffet process. In *Proceedings of the NIPS*, 2005.

Hakkani-Tür, D., Heck, L., and Tür, G. Exploiting query click logs for utterance domain detection in spoken language understanding. In *Proceedings of the ICASSP*, 2011.

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, X. Understanding the semantic structure of noun phrase queries. In *Proceedings of the ACL*, 2010.

Li, X., Wang, Y.-Y., and Acero, A. Learning query intent from regularized click graphs. In *Proceedings of the SIGIR*, 2008.

Rai, P. and DaumeIII, H. The infinite hierarchical factor regression model. In *Proceedings of the NIPS*, 2008.

Schapire, R.E. and Singer, Y. Boostexter: A boosting based system for text classification. In *Machine Learning*, volume 2/3, pp. 135–168, 2000.

Tür, G. and Mori, R. De (eds.). *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech.* John Wiley and Sons, New York, NY, 2011.

Teh, Y.W., Gorur, D., and Ghahramani, Z. Stick-breaking construction for the indian buffet process. In *Proceedings of the AISTATS*, 2007.