# IsNL? A Discriminative Approach to Detect Natural Language Like Queries for Conversational Understanding

*Asli Celikyilmaz, Gokhan Tur, Dilek Hakkani-Tür*

Microsoft Silicon Valley, USA

{asli,gokhan.tur,dilek}@ieee.org

## Abstract

While data-driven methods for spoken language understanding (SLU) provide state of the art performances and reduce maintenance and model adaptation costs compared to handcrafted parsers, the collection and annotation of domain-specific natural language utterances for training remains a time-consuming task. A recent line of research has focused on enriching the training data with in-domain utterances by mining search engine query logs to improve the SLU tasks. However genre mismatch is a big obstacle as search queries are typically keywords. In this paper, we present an efficient discriminative binary classification method that filters large collection of online web search queries only to select the natural language like queries. The training data used to build this classifier is mined from search query click logs, represented as a bipartite graph. Starting from queries which contain natural language salient phrases, random graph walk algorithms are employed to mine corresponding keyword queries. Then an active learning method is employed for quickly improving on top of this automatically mined data. The results show that our method is robust to noise in search queries by improving over a baseline model previously used for SLU data collection. We also show the effectiveness of detected natural language like queries in extrinsic evaluations on domain detection and slot filling tasks.

**Index Terms**: natural language, keyword search, natural language understanding, web search, semantic parsing.

## 1. Introduction

The goal of human to machine dialog systems is to provide the user with a seamless experience, in which users can speak to the machine naturally as if they are conversing with another human. The spoken language understanding (SLU) component of the dialog systems plays a crucial role in extracting the requested information from the user input. A typical SLU engine employs several semantic parsing methods such as domain detection, user act (intent) determination or slot filling to better understand the user input [1]. Compared to a typical keyword-based web search query, the input to a dialog system is a natural language (NL) utterance, which usually contains verbs, phrases and clauses (see Table-1).

Most state of the art approaches to SLU are based on supervised machine learning methods, which use training data from the corresponding application domain. Among these approaches are generative models such as hidden Markov models [2], discriminative classification methods [3, 4, 5] and probabilistic context free grammars [6, 7]. Although very effective in semantic parsing of utterances, they require a large number of in-domain NL sentences. Manually collecting NL sentences for training does not scale well because of the language variability issues of the NL interfaces. In particular, not only there is no limitation on what the user might say, but the models must generalize from a tractably small amount of training data.

| **Natural Language (NL) Queries** | |
|---|---|
| **(S)** *what time do [lakers]$_{team}$ play in the [opening day]$_{date}$* | |
| **(M)** *what are some [recent]$_{date}$ [funny]$_{genre}$ movies* | |
| **(G)** *[top-rated]$_{review}$ [wii]$_{type}$ games for [kids]$_{genre}$* | |
| **(M)** *what are all of [channing tatum]$_{artist}$ 's movies* | |
| **Keyword Queries** | |
| · *calories per day* | · *wifi signal booster xbox 360* |
| · *oscar winners [2013]$_{date}$* | · *[jessica simpson]$_{artist}$* |
| · *[stolen honor : wounds that never heal]$_{movie}$* | |

Table 1: Sample natural language and keyword queries mined from web search query click logs. Queries are labeled with selected semantic tags including domain labels; (M):Movie, (G):Game, (S):Sports, and slot tags, e.g., *type*, *genre*, *artist*, etc.

In a closely related research area, the information retrieval (IR) researchers have recently shown that the web search query click logs (QCL) are valuable resources that can be used as implicit supervision to improve the predictions of the future search results [8]. Specifically, the web search query-click log data includes the queries issued by the users. The queries have corresponding url-links that the users clicked from a list of urls returned by the search engine (see Figure 1). It is the the strong semantic relation between the queries issued by the users and the clicked urls that help to understand the queries. Only recently this relational but noisy data has been a valuable information source for building spoken dialog systems. For instance a recent study on the use of QCL data for building SLU models has shown improvements, in particular, on the domain and slot detection tasks [9, 10]. Typically, they mine the QCL data to extract additional NL-like queries, which are then used to build more robust and efficient SLU models.

With the above improvements on SLU in mind, in this paper, we focus on rather efficient methods to mine NL queries for improving the SLU models. We start by summarizing the mining methods used in the previous SLU work, which sets the background of this paper. We then present a new feature-based NL classifier model, namely the **IsNL** to classify search queries into "NL" or "keyword" categories based on semantic, syntactic and structural features extracted from the queries and external resources. Using an active learning method, we select the training data that best generalizes the SLU models. Specifically, we collect queries to: (*i*) extend the vocabulary; (*ii*) and capture NL patterns and phrases that did not exist in the training data. Our end goal is to improve the performance of the understanding tasks, specifically, domain detection and slot filling. In the empirical evaluations we show that the IsNL classifier is an ef-

fective tool to enrich the data for training SLU models.

The next section presents the background of mining query click logs to obtain NL queries, while Section 3 details the IsNL classifier tool, and active learning approach. Section 4 evaluates the IsNL classifier's robustness to SLU domain and slot detection errors and compares it against results reported in previous studies [11, 12]. Finally Section 5 concludes with a discussion of future work.

## 2. Implicitly Supervised Approach to Mining Data for SLU

### 2.1. Spoken Language Understanding (SLU) Models

The standard method to building a statistical SLU system is to train semantic classifiers, i.e., the domain/intent and slot filling models using in-domain data, as summarized below:

► **Domain Detection**; is taken as an utterance classification task, in which word $n$-grams, syntactic and semantic features such as domain indicating dictionaries, salient phrases (details below), etc., are used as explanatory variables to predict a domain label for a given utterance. In this paper, for the domain classifier we use icsiboost [13], an implementation of the AdaBoost.MH algorithm, a member of the boosting family of classifiers [14]. Boosting is an iterative procedure that builds a new weak learner $h_t$ at each iteration. Each example of the training data is assigned a weight. These weights are initialized uniformly and updated on each iteration so that the algorithm focuses on the examples that were wrongly classified during the previous iteration. At the end of the learning process, the weak learners used at each iteration $t$ are linearly combined to form the classification function, $f(x,l) = \sum_{t=1}^{T} \alpha_t h_t(x,l)$ with $\alpha_t$ the weight of the weak learner $h_t$ and $T$ the number of iterations of the algorithm.

► **Semantic Tagging**; Following the state-of-the-art approaches for slot filling [4, 5, among others], we use discriminative statistical models, namely conditional random fields, (CRFs) [15]. More formally, slot filling is framed as a sequence classification problem to obtain the most probable slot sequence $\hat{Y} = \text{argmax}_Y\, p(Y|X)$, where $X = x_1, ..., x_T$ is the input word sequence and $Y = y_1, ..., y_T, y_i \in C$ is the sequence of associated class labels, $C$. CRFs define the conditional probability, $p(Y|X)$ as [15]:

$$p(Y|X) = \frac{1}{Z(X)} exp\left(\sum_k \lambda_k f_k(y_{t-1}, y_t, x_t)\right) \quad (1)$$

in which both $X$ and $Y$ are sequences given a set of features $f_k$ (such as $n$-gram lexical features, state transition features, or others) with associated weights $\lambda_k$. $Z(X)$ is the normalization term. After the transition and emission probabilities are optimized, the most probable state sequence, $\hat{Y}$, can be determined using the well-known Viterbi algorithm.

### 2.2. Motivation for Mining In-Domain Queries

To build an efficient and robust SLU system, the domain classifier and slot filling model training employ as many in-domain data as possible, because such semantic models require significant utterance variability to generalize for test cases. For example, queries "*where is [skyfall]$_{movie}$ playing*", and "*show me the nearest theaters in [palo alto]$_{city}$*" share the same domain, i.e., movies, but they have no lexical or slot type overlap. This makes the SLU tasks challenging in that, not only are there no a priori constraints on what the user might say, the system must also generalize from a tractably small amount of training data. In addition, collecting in-domain data does not necessarily scale well for certain domains, e.g., specific tail domains[1] such as *fly-fishing*, for which there are limited online resources, or specialized forms of head domains such as the *ancient (or rare) books*, even though the *books* is a head domain. Furthermore, collecting in-domain data is a time-consuming and exhaustive process, which might take a long time before one can quickly test-drive a newly constructed dialog system for the desired domain.

There is an opportunity to tailor search query click logs (QCL) to automatically expand the in-domain specific vocabulary because query logs directly reflect users interaction patterns (intentions). Specifically, users click on URLs returned by a search engine related to their queries, hence providing implicit supervision of the broad category of the user's task. For example, two users typing in queries "*reviews for skyfall*" and "*skyfall critics*" and both clicking on "*rottentomatoes.com*", are likely to have the same "*check-reviews*" intent from *movies* domain. Given the motivation, the next section explains our approach for mining web search query click logs for training SLU models.

### 2.3. Web Search Query Click Logs

Large-scale engines such as Bing or Google log more than 100M search queries each day. Search query click logs can be represented as a bipartite graph, with two types of nodes, corresponding to queries and URLs. Figure 1, on the right, shows the conceptual bipartite graph for such search query click logs. Left vertices correspond to queries and right vertices correspond to whole URLs. An edge $e_{i,j}$ is added to the graph if a user who typed query $q_i$ clicks on the URL $u_j$.

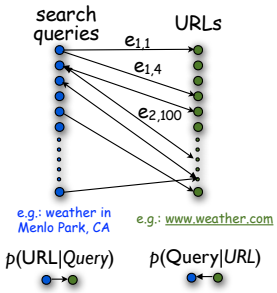


Figure 1: The search query click logs represented as a weighted bipartite graph.

Example clicks for some queries are shown below:

| | |
|---|---|
| Query: | *who directed the count of monte cristo* |
| URL: | www.imdb.com/title/tt0047723/fullcredits |
| URL: | en.wikipedia.org/wiki/The_Count_of_Monte_Cristo |
| Query: | *zucca reviews* |
| URL: | www.yelp.com/biz/zucca-ristorante-mountain-view |
| URL: | reviews.opentable.com/0938/14689/reviews.htm |

Each clicked link has a corresponding frequency indicating the number of users entering the query clicked on that link.

### 2.4. Domain-Independent Salient Phrases

The QCL data contains two types of queries $\mathcal{Q}$: the NL queries $q^{(N)}$, and the keyword search queries $q$. To mine only the NL queries in $\mathcal{Q}$, we search for queries that contain salient phrases that exists in our in-domain NL utterances obtained from our real SLU system [11]. Inspired by the How May I Help You (HMIHY) intent determination system [16], we find phrases that are salient in several different domains. Such phrases, e.g., *show me all the* or *i wanna get information on*, frequently appear in NL utterances directed to spoken dialog systems for

[1] The *head* and *tail* domain concepts indicate the most frequent and seldom visited web sites, respectively (as borrowed from the information retrieval research). Hence, the frequency of the web sites show an exponentially decaying curve over different domains.

| NL Query | Keyword Query |
|---|---|
| *what are the signs of throat cancer* | *throat cancer symptoms* |
| *how many calories do i need in a day* | *calories per day* |
| *how do i know if i am anemic* | *anemic* |

Table 2: Sample natural language query and corresponding keyword search query pairs mined from query click logs

information access. To this end, we use the available labeled training data from several domains and extract salient phrases as follows:

For each n-gram $n_j$ in this data set, we compute a probability distribution over domains $d_i \in D$: $P(d_i|n_j)$, and then compute the Kullback-Leibler (KL) divergence between this distribution and the prior probabilities over all domains $P(d_i)$:

$$S(n_j) = KL(P(d_i|n_j)||P(d_i)) \quad (2)$$

The word $n$-grams with the least divergence from the prior distribution are chosen as the domain-independent salient phrases.

### 2.5. Mining the Training Data

Once we compile a list of domain independent salient phrases, we mine for search queries, $q_k^{(N)} \in \mathcal{Q}$, that include these phrases. These NL-like queries form the seed set for mining *pairs* [17]. Pairs are defined by combining as a set of queries (NL or keyword query) that are most semantically similar to the NL queries, $q_k^{(N)}$. Thus, using the query click graph, we find a set of query pairs, where each pair includes an NL query and a semantically similar keyword query. The similarity between an NL query, $q_k^{(N)}$, and a query, $q_i$, is defined as:

$$sim(q_k^{(N)}, q_i) = \sum_j P(q_i|u_j) \times P(u_j|q_k^{(N)}) \quad (3)$$

This is similar to a two step walk on the query click graph. It should be noted that the graph walking over all possible URLs is an exhaustive and time-consuming process. Therefore, first we find the URLs that has the maximum click probability given the corresponding queries, $q_k^{(N)}$:

$$\hat{u} = argmax_u P(u|q_k^{(N)}) \quad (4)$$

Second, we approximate the similarity measure as follows:

$$sim(q_k^{(N)}, q_i) = P(q_i|\hat{u}) \times P(\hat{u}|q_k^{(N)}) \quad (5)$$

We then use the pairs that have the highest similarity as training data for the IsNL classifier. Some of the pairs mined from Bing search engine logs are shown in Table 2. Note that, there are cases where the words or phrases in the input query are transformed into other words (such as "*what are the biggest U S companies*" is transformed into "*fortune 500 companies*"). We mined 30 million unique queries that include Domain Independent Salient Phrases (DISP) from the Bing search logs, and then walking through the click graph, we extracted 15 million NL and keyword query pairs.

## 3. IsNL: Detecting Natural Language Queries

### 3.1. Baseline Natural Language Classifier, $\texttt{NLC}_B$

We use the NL and keyword search queries that we mined from the query click logs to build our baseline binary natural language classifier model, denoted as $\texttt{NLC}_B$. This baseline IsNL classifier predicts if a given query is keyword or well-formed natural language sentence. The natural language queries, which contain the domain independent salient phrases are taken as positive "*NL*" class and the most frequent keyword queries are taken as the negative "*Keyword*" class for the supervised natural language classifier model using icsiboost trainer (as detailed in §2). The model provides confidence scores indicating the likelihood of the query belonging to either class $c$={"*NL*" or "*keyword*"}; $p(c =$"*NL*"$|q_i)$ and $p(c =$"*keyword*"$|q_i)$.

The baseline $\texttt{NLC}_B$ has two sets of features :

- ***Lexical features*** constitute the word *n*-grams,
- ***Structural features*** are indicators beyond word *n*-grams such as query *length*.

After the baseline classifier $\texttt{NLC}_B$ is trained, we performed an active learning strategy to improve the efficiency (the results are shown in the experiments).

### 3.2. Active Learning for IsNL Classifier Training, $\texttt{NLC}_A$

Active learning is a proven method for reducing the cost of creating the training sets by selecting unlabeled examples that are maximally informative for the statistical learning method and handing them to a human annotator for labeling [18]. In this paper, we use a different version of certainty-based active learning to extend our training data using unlabeled web search queries. Instead of manually annotating low confidence examples, which the IsNL classifier is not confident about, we used the active learning framework to reduce the noise in the data. This is due to the fact that most NL-like queries are actually music lyrics or movie quotes, which are very hard to filter automatically.

To this end, we pulled a month of web query logs and extracted the search queries (no specific domain information is used at this point). We decoded each query using $\texttt{NLC}_B$ with "NL" or "keyword" tags and obtained the confidence values. Based on the confidence scores, we automatically selected 100K possible "NL" queries and 100K possible "keyword" queries:

$$q_k \in \begin{cases} "NL", & p("NL"|q_k) > threshold \\ "keyword", & p("keyword"|q_k) > threshold \end{cases}$$

Using a different profanity classifier, we filtered out the queries with profanity words from "NL" queries prior to annotation. Profanity filtering enables confusions that may be caused by similar entities such as music lyrics or movie names, etc. Next, using online crowd-sourcing tools (similar to Amazon's Mechanical Turk), we labeled 200K queries (15K unique) as "NL' or "keyword". We used 2 labelers per query and filtered out the queries when there is a disagreement. We asked the labelers to judge queries with well formed phrases such as "*restaurants with live music*" or "*top rated restaurants in the city*" as "NL" query rather than a "keyword" query. We then re-trained a new IsNL classifier with this manually labeled data, denoted as $\texttt{NLC}_A$, using the baseline model features with an addition of semantic features. These features represent:

- ***Semantic features*** represent semantic categories such as music lyrics, movie quotes. Such classes help to identify that the queries such as "*Celsius 41.11: The temperature at which the brain begins to die*" is the title of a 2004 movie rather than an NL query. We also included a *profanity* indicator as an additional binary semantic feature so that a query that contains a profanity word is considered not an NL query [2].

[2]The profanity lexicon is compiled using a different classifier trained on domain independent queries.

| Data and Features | F-Measure |
|---|---|
| Baseline (Length $>$ 10) | 61.0% |
| **NLC**$_B$ + lexical | 84.0% |
| **NLC**$_B$ + lexical+structural | 73.6% |
| **NLC**$_B$ + lexical+structural+semantic | 73.6% |
| **NLC**$_A$ + lexical+structural | 83.1% |
| **NLC**$_A$ + lexical+structural+semantic | **89.4%** |

Table 3: F-measure using baseline mined data, $\mathtt{NLC}_B$, and manually annotated data, $\mathtt{NLC}_A$, with varying features.

# 4. Experiments and Results

We present both intrinsic and extrinsic experimental results on the IsNL classifier performance. For intrinsic evaluations, we used a test set of manually annotated queries. For extrinsic experiments, we checked how the SLU domain and slot filling models are affected when the NL queries are injected into the existing in-domain training data.

## 4.1. Intrinsic Evaluations

The data used in our experiments comes from two sources:
• **(B) Mined Data from Implicitly Supervised Approach:** We randomly sampled 280K queries ($\sim$100K NL and $\sim$170K keyword queries) from the 15 million NL pairs (as explained in §2).
• **(A) Data from Active Learning Approach**: As explained in §3.2, we obtained an additional 200K queries, in which 100K NL and 100K keyword queries.

We separated out 5000 queries to construct the test set and evaluated the performance of the IsNL classifier **NLC**$_A$ based on active learning against the baseline model **NLC**$_B$. The active learning model is performing better as expected. This shows that enriching the training data with additional queries selected with the active learning is an effective method for classification of the NL queries. In addition, we obtain the best performance when we use structural and semantic features, and the effect of the semantic features is noticeable.

## 4.2. Extrinsic Evaluations

In this experiments, we demonstrate the effectiveness of the IsNL classifier in building unsupervised domain classification and slot filling models. Although the models are trained with queries, their performances are tested on spoken utterances.

### *4.2.1. Domain Classification*

In our previous work, we have proposed the use of web search queries hitting to target domain web pages (like rottentomatoes.com or fandango.com for the movies domain) to bootstrap domain classification models in an unsupervised fashion [11]. In that work, we used longer queries with DISP. In accordance with the intrinsic evaluations, we used the queries which are classified as NL-like, and replicated the same experiments.

Our experiments have a controlled setup where a single domain is first learned starting with the data obtained only from the query logs (in an unsupervised fashion), and then replicated for all 5 target domains using Boosting using only the word $n$-grams. Table 4 presents results of our experiments on a test set of about 1K utterances from our conversational understanding system. Additional details of our experimental setup are explained in our previous paper [11].

### *4.2.2. Slot Filling*

In our previous work, we presented a novel approach in which we used web search queries hitting to the target *structured* web pages (such as *imdb.com* for the movies domain) [12]. Then, we matched the information extracted from the structure of the web pages (such as actor or director name) with these web queries to obtain unsupervised semantic annotations of these queries. In there we showed that using queries which contain a stopword performed the best for that task. Table 5 presents results using a test set of about 300 utterances from a conversational understanding system on movies domain. More details of this experimental setup can be found in the previous paper [12].

| Query Selection | Top Class Error Rate |
|---|---|
| Baseline | 16.96% |
| Upper Bound | 6.50% |
| Random | 13.70% |
| With DISP | 7.98% |
| IsNL Classifier | 8.57% |

Table 4: Using IsNL classifier for selecting 5K queries for unsupervised domain classification. Baseline is with no query selection, and upper bound is using existing manually labeled data of 4K utterances. DISP:Domain independent salient phrase.

| Query Selection | F-Measure |
|---|---|
| Upper Bound | 64.26% |
| Random | 48.91% |
| (1) With Stop-words | 57.73% |
| (2) IsNL Classifier | 58.08% |
| (1) + (2) | 58.94% |

Table 5: Using IsNL classifier for selecting 50K queries for unsupervised slot filling. Upper bound is using existing manually labeled data of 2,700 utterances.

### *4.2.3. Results and Discussions on Extrinsic Evaluations*

The two immediate points to notice in these results are that, using IsNL classifier significantly outperforms models that only use randomly selected queries as well as it is not significantly worse than the previous heuristics for these two sample SLU tasks - if not better for the slot filling task. Reaching this level of improvement on SLU performance with only an IsNL classifier proves the robustness of this approach, and saves the domain experts time from writing manual rules to select NL queries. Furthermore, task specific heuristics can be combined with the confidence obtained from the IsNL classifier. To show this point, we performed an additional experiment, simply combining the selected queries by the IsNL classifier and the ones with stopwords for slot filling. This improved the F-Measure by 1% absolute pushing it up to 58.94%.

# 5. Conclusions and Future Work

This paper discussed an efficient and effective feature-based approach to the automatic extraction of well formed NL queries for spoken language model training.

There are several directions that can be pursued to improve the accuracy of the natural language classifiers: For example, we have not used syntactic or semantic parsers to extract additional features for the IsNL classifier. This is because parsers introduce latency issues when they are used at runtime and we would like to build a fast tool to filter millions of search queries issued everyday in a reasonable time. However, as a feature work we will investigate shallow semantic parsers to extract additional semantic features for the IsNL classifier. In addition, we believe using several domain specific dictionaries might help to identify queries comprised solely of a single entity.

## 6. References


[1] G. Tur and R. D. Mori, Eds., *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. New York, NY: John Wiley and Sons, 2011.

[2] R. Pieraccini, E. Tzoukermann, Z. Gorelov, J.-L. Gauvain, E. Levin, C.-H. Lee, and J. G. Wilpon, "A speech understanding system based on statistical representation of semantics," in *Proceedings of the ICASSP*, San Francisco, CA, March 1992.

[3] R. Kuhn and R. D. Mori, "The application of semantic classification trees to natural language understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 449–460, 1995.

[4] Y.-Y. Wang and A. Acero, "Discriminative models for spoken language understanding," in *Proceedings of the ICSLP*, Pittsburgh, PA, September 2006.

[5] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proceedings of the Interspeech*, Antwerp, Belgium, 2007.

[6] S. Seneff, "TINA: A natural language system for spoken language applications," *Computational Linguistics*, vol. 18, no. 1, pp. 61–86, 1992.

[7] W. Ward and S.Issar, "Recent improvements in the CMU spoken language understanding system," in *Proceedings of the ARPA HLT Workshop*, March 1994, pp. 213–216.

[8] N. Craswell and M. Szummer, "Random walk on the click graph," in *Proceedings of the ACM SIGIR*, Amsterdam, Netherlands, 2007.

[9] D. Hakkani-Tür, G. Tur, L. Heck, A. Celikyilmaz, A. Fidler, D. Hillard, R. Iyer, and S. Parthasarathy, "Employing web search query click logs for multi-domain spoken language understanding," in *Proceedings of the IEEE ASRU*, Waikoloa, HI, 2011.

[10] D. Hakkani-Tür, L. Heck, and G. Tur, "Exploiting query click logs for utterance domain detection in spoken language understanding," in *Proceedings of the ICASSP*, Prague, Czech Republic, 2011.

[11] D. Hakkani-Tür, G. Tur, L. Heck, and E. Shriberg, "Bootstrapping domain detection using query click logs for new domains," in *Proceedings of the Interspeech*, Florence, Italy, 2011.

[12] G. Tur, M. Jeong, Y.-Y. Wang, D. Hakkani-Tür, and L. Heck, "Exploiting the semantic web for unsupervised natural language semantic parsing," in *In Prooceedings of the Interspeech*, Portland, OR, September 2012.

[13] B. Favre, D. Hakkani-Tür, and S. Cuendet, "Icsiboost," http://code.google.come/p/icsiboost, 2007.

[14] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[15] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the ICML*, Williamstown, MA, 2001.

[16] A. L. Gorin, G. Riccardi, and J. H. Wright, "How May I Help You?" *Speech Communication*, vol. 23, pp. 113–127, 1997.

[17] D. Hakkani-Tür, G. Tur, R. Iyer, and L. Heck, "Translating natural language utterances to search queries for slu domain detection using query click logs," in *In Prooceedings of the IEEE ICASSP*, Kyoto, Japan, March 2012.

[18] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, pp. 201–221, 1994.