Employing Web Search Query Click Logs for Multi-Domain Spoken Language Understanding

Dilek Hakkani-Tür Gokhan Tur Larry Heck Asli Celikyilmaz
Ashley Fidler Dustin Hillard Rukmini Iyer Sarangarajan Parthasarathy

Speech Labs, Microsoft, Mountain View, CA

dilek@ieee.org,gokhan.tur@ieee.org,lheck@microsoft.com
asli@ieee.org,{v-asfidl,dustinh,rukmini,sarangp}@microsoft.com

***Abstract*—Logs of user queries from a search engine (such as Bing or Google) together with the links clicked provide valuable implicit feedback to improve statistical spoken language understanding (SLU) models. In this work, we propose to enrich the existing classification feature set for domain detection with features computed using the click distribution over a set of clicked URLs from search query click logs (QCLs) of user utterances. Since the form of natural language utterances differs stylistically from that of keyword search queries, to be able to match natural language utterances with related search queries, we perform a syntax-based transformation of the original utterances, after filtering out domain-independent salient phrases. This approach results in significant improvements for domain detection, especially when detecting the domains of web-related user utterances.**

I. Introduction

Spoken language understanding (SLU) in human/machine spoken dialog systems aims to automatically identify the user's goal-driven intents for a given domain, as expressed in natural language, and extract associated arguments, or slots, according to a semantic template [1]. For multi-domain SLU systems, a top level domain classification serves as a triage service. The state-of-the-art approach for training domain detection models relies on supervised machine learning methods that use lexical, contextual, and other semantic features. To enrich this feature set, the proposed approach relies on exploiting an abundant set of web query click logs (QCLs), which pair web search queries with their click information. While this is very valuable data waiting to be mined for language understanding, it is not generally straightforward, since most queries are just keywords (instead of natural language sentences), and because the implicit supervision via click information is very noisy, given that most people simply click on the top result link.

Enabling users to speak naturally to computers has been a goal for some time. Many spoken dialog systems motivate users to speak naturally by using explicit prompts, such as *You can speak naturally to me*. On the other hand, the success and broad use of keyword search engines imply the strength of keyword searches; some users attempt to speak in keywords hoping for better machine understanding. While it is difficult to formulate keyword searches for all user intents, a spoken dialog system should be able to handle both styles, as another motivation for this study.

Query click data includes logs of search engine users' queries and the links these users click from a list of sites returned by the search engine. Previous work has shown that click data can be used to improve search decisions [2, among others]. Regarding spoken language processing, our previous work mainly benefited from the mining of training data to train domain detection models when little [3] or no [4] in-domain data was available. In this work, instead of mining more data, we enrich the existing training data sets with new features, computed using the click distribution over a set of related URLs, from search query click logs. Since the form of the natural language utterances differs from the shorter keyword search queries, to be able to match natural language utterances with search queries, we transform the original utterances to query-like sentences using a syntax-based transformation, similar to a method proposed in our previous work [5].

In the next section we briefly describe the task of domain detection in SLU. Then we review the related work from both the information retrieval and spoken language processing communities in Section III. In Section IV, we present our approach along with the query click logs, utterance transformation algorithm, and feature extraction methods. Section V presents the experiments and detailed results using a multi-domain SLU system. We conclude after a brief discussion of the results in Section VI.

II. Domain Detection

In multi-domain SLU systems, domain classification is often completed first, serving as a top-level triage for subsequent processing. For example, the conversational system may support requests related to airline travel, weather, calendar scheduling, directory assistance, and so on. While in some cases the boundaries of domains are not clear, this modular design approach has the advantage of flexibility; specific modifications (e.g., insertions, deletions) to a single domain class can be implemented without requiring changes to the other domains [6], [7]. Also, such an approach often yields more focused understanding in each domain, since the intent determination and slot filling only need to consider a relatively small set of classes over a single (or limited set) of domains.

978-1-4673-0367-5/11/$26.00 ©2011 IEEE

It must be noted that this triaging approach does not prevent the use of domain-specific SLU model outputs for domain detection. Furthermore, it can be extended to hierarchical SLU models with multiple levels of domains and subdomains. For example, a SLU system in the travel assistance domain may hierarchically represent related subdomains such as flight reservations, hotel booking, and car rentals.

Similar to intent determination systems like AT&T How May I Help You [8], domain detection is often framed as an utterance classification problem [3]. More formally, given a user utterance or sentence x_i, the problem is to associate a set $y_i \subset C$ of semantic domain labels with x_i, where C is the finite set of domains covered. To perform this classification task, the class with the maximum conditional probability, $p(y_i|x_i)$ is selected:

$$\hat{y_i} = \underset{y_i}{\operatorname{argmax}}\, p(y_i|x_i)$$

Usually, supervised classification methods are used to estimate these conditional probabilities, and a set of labeled utterances is used in training. Classification may employ lexical features such as word n-grams, contextual features such as the previous turn's domain, semantic features such as named entities in the utterance [9], syntactic features such as part-of-speech tags, topical features such as latent semantic variables [10] and so on.

III. Related Work

Previous work on web search has benefited from the use of query click logs for improving query intent classification. Li *et al.* used query click logs to determine the domain of the query (typically not in natural language), and then inferred the class memberships of unlabeled queries from those of the labeled queries using the URLs the users clicked [11]. For example, the queries of two users who clicked on the same URL (such as, www.hotels.com) are assumed to belong to the same domain (*hotels* in this case). They formed a bipartite graph of the queries and URLs the users clicked on, then transferred the labels from queries to URLs and to other queries using a label propagation algorithm [12], [13].

In our earlier work, we extended this idea in order to use the noisy supervision obtained from query click information in the semi-supervised label propagation algorithm by sampling high-quality query click data mined from query logs for domain detection [3]. This resulted in a 20% relative reduction in the domain detection error rate for SLU in a semi-supervised setup.

In [4], we proposed, using web search query logs, to bootstrap domain detection for new domains. While sampling user queries from the query click logs to train new domain classifiers, we introduce two types of measures based on the behavior of the users who entered a query and the form of that query. We show that both types of measures result in reductions in the error rate, as compared to randomly sampling training queries. In controlled experiments over five domains, we achieved the best gain from the combination of the two types of sampling criteria.

Most related to slot filling, Li *et al.* exploited query click logs leveraging domain-specific structured information for web query tagging [14]. They built semi-supervised models using these derived labels. Liu *et al.* proposed automatically populating gazetteers to be used in slot filling from web queries [15]. Using a seed gazetteer, they mined the query click logs to expand it using a generative model. They learned target websites based on the seed gazetteer entries; for example, `www.imdb.com/title` is a candidate website for movie names. Then they added other queries that hit the same website with high frequency as new gazetteer candidates, and then used statistical methods to weight them. The contextual words (such as in *cast of Avatar* or *when was the movie As Good As It Gets released*) were then stripped out using the existing seed gazetteer entries. In our previous work, we exploited query click logs for bootstrapping weighted named entity gazetteers [9] and slot filling models [16].

In this paper, we propose using query click logs to compute new features for domain detection after transforming the input utterance into web search query form.

IV. Approach

The proposed approach relies on leveraging the implicitly annotated data coming from the query click logs as additional features for training domain detection classification models. While this is straightforward in cases where a given user utterance is found in the query click logs with relatively high frequency, the language users employ with a SLU system is very different from typical queries. Note that, for some domains, such as generic intents like frustration or chit-chat, or where the users are scheduling their own meetings, queries are very unlikely to occur in the search logs.

This study is motivated by the assumption that people typically have conceptual intents underlying their requests. They then generate different sequences of words depending on whether they interact with a web search engine, another human, or an intelligent SLU system. When they wonder about the *capacity of a Boeing 737*, they can form a simple query such as *capacity Boeing-737* when interacting with a search engine. The top wikipedia page will have the information requested. When they are interacting with an intelligent natural language dialog system, they can generate a more natural utterance, such as *what is the capacity of a Boeing 737 airplane*. In our previous work on sentence simplification [5], we proposed using a syntactic parsing based sentence transformation method to convert these input utterances to *capacity 737*, so that the classifier can perform better on them.

One immediate advantage we have noticed with this approach is that these transformed sentences look very much like search engine queries. Hence, it might be possible to use the URL click distributions given that query. Our approach thus has two components:

- Sentence transformation to query language
- Feature extraction from query click logs

The high level approach is depicted in Figure 1. The exact and transformed user utterances are checked against the query

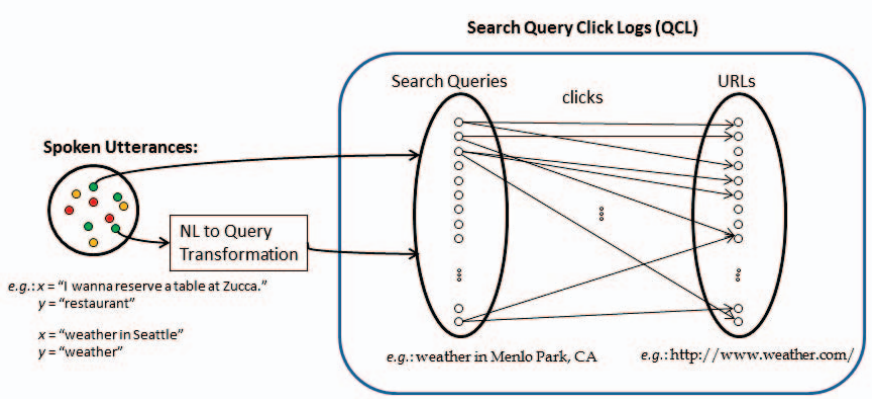

Fig. 1. The conceptual process for exploiting query click logs for domain detection.

click logs. If they are still not seen in the query click logs, this information is also provided to the classifier, as it indicates that the input probably belongs to a domain where there are no queries categorically related to information on the web, such as calendar scheduling.

In the following subsections we describe each of these key components. One important aspect of this study is that the implicit feedback extracted from query clicks provides an orthogonal view of the domain classification problem once user utterances are transformed into query language. This leads the way to many potential research ideas beyond this study, given the abundance of this contextual information.

A. Web Search Query Click Logs

Example click log queries with resulting clicks 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

Note that each of the clicked links comes with frequencies showing the number of users entering that query clicked on that link. While in certain cases, the URL domain name is a direct indicator of the target domain (e.g., `opentable.com` receives queries about restaurant reservation, imdb.com receives queries about movies, etc.), general information web pages such as wikipedia.com provide only indirect information.

B. Utterance-to-Query Transformation

For domain detection, lexical features (such as word n-grams of the input utterance) are typically the most informative classification features [17]. However, the word n-grams extracted from natural language utterances and keyword search queries would not be the same, given the different forms of these query types described above. Web search queries often represent keyword searches, such as *mountain view restaurant*, which would be realized in natural conversations as complete utterances, such as *find me a restaurant near mountain view*. Non-keywords are often missing in search queries, and keywords may be in a different order than in natural language utterances, requiring transforming of input utterances to a form similar to that of search queries.

In our previous work [5], we have presented a sentence simplification algorithm for improving the intent detection performance of a SLU system and showed its effectiveness using the well known Airline Travel Information System (ATIS) [18] task. In this study, we first exclude the domain-independent salient phrases as described below, perform syntactic parsing on the remaining sub-sentence, and choose the query terms for natural language to query transformation.

- *Domain-Independent Salient Phrases:* Inspired by the How May I Help You (HMIHY) intent determination system [8], we find phrases that are salient for more than one domain. To this end, we use the available labeled training data from other domains. For each n-gram n_j in this data set, we compute a probability distribution over domains: $P(domain_i|n_j)$, and then compute the Kullback-Leibler (KL) divergence between this distribution and the prior probabilities over all domains $P(domain_i)$:

$$S(n_j) = KL(P(domain_i|n_j)||P(domain_i))$$

  Then the word n-grams that show the least divergence from the prior distribution are selected as the domain-independent salient phrases. These are phrases such as *show me all the* or *i wanna get information on* that frequently appear in natural language utterances directed to spoken dialog systems for information access.
- *Syntactic Parsing:*
  The sentence-to-query transformation procedure we employ in this study relies on dependency parses of the sentences, where the structure of a sentence is determined by the relation between a word (a head) and its dependents. Each word has a head it is pointing to. For example, for the noun phrase *blue book*, *blue* points to *book*.
  In this study we employ the Berkeley Parser [19], a state-of-the-art parser trained from a treebank following a latent variable approach by iteratively splitting non-terminals to better represent the data. We use the LTH

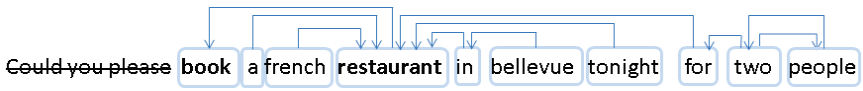

Fig. 2. Dependency parse of an example sentence *could you please book a French restaurant in Bellevue tonight for two people* and demonstration of sentence-to-query transformation shown in bold.

Constituency-to-Dependency Conversion toolkit[1] to form dependency parses from the output parse trees. To adapt the parser to the speech domain, we retrain it using monocase WSJ treebank stripping out punctuation [20] and further employ a self-training approach using the available training data. This process improves the parser's ability to handle monocase words, lack of punctuation and conversational style sentences which rarely occur in textual corpora.

Once the sub-sentence is dependency parsed, the transformation algorithm picks the top level predicate and its dependents (arguments). Figure 2 depicts this example using the parse tree of the subsentence *could you please book a french restaurant in bellevue tonight for two people*, excluding the domain-independent salient phrase *could you please*. The top level predicate, *book*, and its dependents, only one in this case, *restaurant*, are chosen as query terms. This is different than the previous simplification approach, as the goal is not to improve the classification model, but instead get reliable hits in the query click logs. For prepositional phrase dependents we also experimented with the head noun. For example, the sentence *what is a good restaurant for a twenty first birthday dinner in Orlando* can be converted into the query *restaurant dinner Orlando*.

C. Query Click Feature Extraction

Following the established literature on user utterance intent determination and domain detection, the baseline model uses lexical features, i.e., word n-grams as extracted from user utterances. In order to examine what users with similar intentions or information requests do with the web search results for their query, we search for each transformed utterance in our data set in the Bing web search query and click logs. We search all the queries in the training data set amongst the search queries, and download the list of clicked URLs and their frequencies. To reduce the number of features, we extract only the base URLs (such as opentable.com or wikipedia.com), as is commonly done in the web search literature. We use the list of the 1000 most frequently clicked base URLs for extracting classification features (QCL features). More formally, for each input user utterance, x_j, we compute $P(URL_i|x_j)$, where $i = 1, , 1000$.

In addition to using $P(URL_i|x_j)$ features, we also compute the click probability distribution distance between each given query and the queries belonging to a target domain, D_k, using the KL divergence:

$$KL_k = KL(P(URL_i|x_j)||P(URL_i|D_k))$$

[1]http://nlp.cs.lth.se/software/treebank_converter/

	Subset	No. Utt.	Avg. No. Words
Training	-	16,000	7.60
Development	NL	1,305 (65.2%)	9.47
	Query	695 (34.8%)	4.26
Test	NL	1,243 (65.3%)	9.31
	Query	659 (34.7%)	4.27

TABLE I
DATA SETS USED IN THE EXPERIMENTS. NL REFERS TO NATURAL LANGUAGE SUBSET, QUERY REFERS TO QUERY-LIKE UTTERANCES.

Domain	**Train**	**Dev.**	**Test**
WR1	18.6%	20.6%	17.1%
WR2	15.0%	14.5%	16.4%
WU1	2.7%	2.4%	2.5%
WU2	23.2%	22.5%	23.3%
Other	40.5%	40.0%	40.7%

TABLE II
DISTRIBUTION OF DOMAINS IN EACH DATA SET.

Then for each domain D_k, KL_k, as well as the domain with the lowest KL divergence, are used as additional features. $P(URL_i|D_k)$ is computed using all utterances in the training set that belong to domain k.

V. EXPERIMENTS

A. Data Sets

In order to automatically detect the domain category of each utterance, we use both their word n-grams, and the base URLs clicked by search users who typed in the same query. We compile a dataset of user utterances from the users of a spoken dialog system. As mentioned earlier, some of these utterances are in the form of full conversational style natural language utterances (NL subset), for example, *I'd like to find out about weather in Mountain View tomorrow*, while others are more similar to web search queries, for example, *weather in Mountain View* (Query-like subset). We manually annotated the development and test set queries with style information. Table I shows the properties of the data sets and the (relative) frequencies of the two types of queries in each data set. While the average number of words per NL and query-like utterances is similar between the development and test sets, query-like utterances contain less than half the number of words as NL queries.

Each of the utterances in these data sets is manually labeled with one of 5 domain categories. The domains were chosen to study the effect of using web search query logs on detecting the domains of user requests related to web-related and unrelated tasks. Hence, 2 of these domains (WR1 and WR2) are also covered by information on the web, such as requests about *weather* and *restaurants*, and 2 of them

Coverage	Training	Dev+Test (Overall)	Dev+Test (NL Subset)	Dev+Test (Query-like Subset)
Full Utt.	23.8%	23.7%	5.7%	57.8%
Trans.	37.9%	38.2%	26.1%	60.9%
Full + Trans.	42.2%	42.2%	28.3%	68.4%

TABLE III
COVERAGE OF THE USER UTTERANCES BY THE QUERY CLICK LOGS.

Approach	Overall	NL Subset	Query-like Subset
Majority Class	59.3%	58.9%	60.1%
Full Utt. QCL feats (A)	48.6%	59.0%	29.0%
Trans. Utt. QCL feats (B)	43.5%	50.0%	31.1%
A+B	41.9%	50.0%	26.7%

TABLE IV
ERROR RATES WHEN ONLY FEATURES COMPUTED FROM QCL ARE USED FOR DOMAIN DETECTION.

(WU1 and WU2) are not, such as requests related to '*e-mails* and *voice-mails*, and one domain covers all the rest of the utterances, i.e., the *other* domain. Note that the *other* domain can include web-related utterances as well, such as *search for the inventor of kaleidoscope*. Table II shows the percentage of each domain category in each data set.

B. Searching for User Utterances in Query Click Logs

To measure the use of these logs, we refer to utterance coverage, that is, the percentage of the dialog system user utterances that were observed in the query click logs, as some users may have entered the same exact query during their search session. Table III lists the coverage for the training, development, and test sets, as well as the NL and query-like subsets of the development and test sets, for the full and transformed utterances. More than half of the full query-like utterances had been searched by some web search user, whereas this is only about 5% for NL utterances. Using query transformation, coverage in the query logs is improved for all queries, but especially for the NL queries (from 5.7% to 28.3%).

C. Results

Similar to prior work on other utterance classification tasks, such as dialog act tagging [21] and intent determination [22], our domain detection approach relies on using icsiboost[2], an implementation of the AdaBoost.MH algorithm, a member of the boosting family of discriminative classifiers [23].

To measure domain detection performance, we compute error rate (ER), that is the percentage of utterances that are not assigned the correct domain category, and F-measure, that is the harmonic mean of recall and precision. Table IV lists error rates only when $P(URL_i|x_j)$ over the list of URLs ($URL_i, i = 1, ..., 1000$) is used as features (excluding any lexical features) . The first row (majority class) lists the error rate when the most frequent domain (*other*) is assigned to each

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

	Approach			
Domain	1	2	3	4
WR1	89.5%	91.8%	91.0%	92.0%
WR2	91.1%	94.6%	94.3%	95.6%
WU1	98.9%	100%	100%	98.9%
WU2	96.8%	97.2%	96.9%	97.2%
Other	91.7%	94.6%	94.5%	94.8%
Overall	92.5%	94.6%	94.1%	95.0%

TABLE VI
F-MEASURES FOR EACH DOMAIN, WITH EACH APPROACH (NOTE THAT THE 4 APPROACHES ARE DESCRIBED IN TABLE V).

example. The first column lists error rates averaged over all examples in the test set, the second and third columns list those for the NL utterances, and query-like utterances, respectively. Adding QCL features with full utterances significantly reduces the error rate on query-like utterances, but does not change the error rate on NL utterances (as only a small subset of them can be retrieved from the query logs); on the other hand, adding QCL features mined with transformed utterances reduces the error rate on NL utterances by 8.9% absolute. Merging the logs for both the full and transformed forms of the utterances results in the lowest error rate on the test set.

Table V lists error rates when features computed from search query click logs are added to word n-grams as features. Similar to the previous set of results, using features from query click logs results in significant reductions in error rate, though the results are mixed, as word n-grams are useful for NL utterances.

Using word n-grams, in addition to the ID of the lowest KL divergence domain as a feature resulted in an error rate of 5.7% on the test set, which is better than word n-grams alone, but not as good as using individual probabilities as features with boosting.

Finally, Table VI provides an analysis of what is happening in each domain. The overall F-measure results are similar to error rates, we get 2.1% absolute improvement in error rate when features for full utterances are used, and 0.4% more improvement when features with transformed utterances are also added. For utterances belonging to the domains not related to the web (WU1 and WU2), the F-measure does not change much across different experiments, as expected. However, the F-measure significantly improves for utterances that belong to web-related domains when features related to clicked URLs are included. The average F-measure for the web-related domains increases from 90.3% to 93.8% using QCL features with the proposed utterance-to-query transformation approach.

VI. DISCUSSION ON OVER-TRANSFORMATION

We extend the idea of transformations in two other ways: we used only the head nouns of the user utterances, as well as all the named entities (extracted from manual annotations). For example, transforming *show me weather in los altos tomorrow morning* to *weather los altos*. This increased the coverage of the user utterances in the query click logs to 72.2%, but the error rate on the test set increased to 6.94%. Manual examination of the errors suggests an issue with over-simplifying

Approach	Overall ER	ER on NL Subset	ER on Query-like Subset
1: Word 1,2,3-grams (n-grams)	7.0%	5.6%	9.7%
2: n-grams + Full Utterance QCL feats (A)	5.2%	5.2%	5.0%
3: n-grams + Transformed Utterance QCL feats (B)	5.7%	5.9%	5.3%
4: n-grams + (A+B)	4.9%	5.4%	3.8%

TABLE V
ERROR RATES WHEN WORD N-GRAMS AS WELL AS FEATURES COMPUTED FROM QCL ARE USED FOR DOMAIN DETECTION.

user utterances into very generic keyword searches. Similarly, we removed all stop-words learned from the training data using frequency and salience measures (similar to finding domain-independent salient phrases) in the utterances simplified by syntax-based transformation. This resulted in a coverage of 65.5%, but also a similar increase in the error rate, to 6.04%. These results indicate that, while transforming natural language utterances to a search query style helps in retrieving clicked URLs, over-simplification results in queries from unrelated domains, and hence suggests employing a more conservative transformation approach.

VII. CONCLUSIONS

We presented methods to exploit the query click logs to improve domain detection in a multi-domain SLU system, in order to provide extra features via the syntax-based transformation of input sentences into a style similar to queries. The experimental results show significant error rate reductions using discriminative classification algorithms. This approach especially improves the performance of web-related user utterances and utterances that are already in the styles of search queries, as expected.

While the approach relies on the availability of query click logs, one can also use a similar technique using search engine results, assuming that the search engines already utilize query click logs. One key observation is that the sentence simplification approach proposed to improve classification is not necessarily the same as utterance-to-query transformation and that these approaches can be used in parallel. Another observation is that, while a transformation approach is useful, over-simplifying natural language utterances results in too generic queries and noisy features from the click logs.

Our future work involves automatic detection of natural language and keyword-based user utterances to treat them differently, as well as investigating the use of query click logs and different transformation approaches more appropriate for other classification tasks, such as user intent detection.

Acknowledgments: We thank Xiao Li, and Ye-Yi Wang for helpful discussions, and the anonymous reviewers for their valuable suggestions.

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] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," in *Proceedings of SIGIR*, Seattle, WA, USA, 2006.

[3] 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, May 2011.

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

[5] G. Tur, D. Hakkani-Tür, L. Heck, and S. Parthasarathy, "Sentence simplification for spoken language understanding," in *Proceedings of the ICASSP*, Prague, Czech Republic, May 2011.

[6] K. Komatani, N. Kanda, M. Nakano, K. Nakadai, H. Tsujino, T. Ogata, and H. Okuno, "Multi-domain spoken dialogue system with extensibility and robustness against speech recognition errors," in *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, Sydney, Australia, July 2006.

[7] C. Lee, S. Jung, S. Lee, and G. Lee, "Example-based dialog modeling for practical multi-domain dialog system," *Speech Communication*, vol. 51, pp. 466–484, 2009.

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

[9] D. Hillard, A. Celikyilmaz, D. Hakkani-Tür, and G. Tur, "Learning weighted entity lists from web click logs for spoken language understanding," in *Proceedings of Interspeech*, Florence, Italy, 2011.

[10] A. Celikyilmaz, D. Hakkani-Tür, and G. Tur, "Multi-domain spoken language understanding with approximate inference," in *Proceedings of Interspeech*, Florence, Italy, 2011.

[11] X. Li, Y.-Y. Wang, and A. Acero, "Learning query intent from regularized click graphs," in *Proceedings of SIGIR'08: the 31st Annual ACM SIGIR conference on Research and Development in Information Retrieval, Association for Computing Machinery, Inc.*, Singapore, July 2008.

[12] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," CMU CALD technical report, Tech. Rep. CMU-CALD-02-107, 2002.

[13] X. Zhu, "Semi-supervised learning with graphs," PhD Dissertation, Carnegie Melon University, 2005.

[14] X. Li, Y.-Y. Wang, and A. Acero, "Extracting structured information from user queries with semi-supervised conditional random fields," in *Proceedings of the ACM SIGIR*, Boston, MA, 2009.

[15] J. Liu, X. Li, A. Acero, and Y.-Y. Wang, "Lexicon modeling for query understanding," in *Proceedings of ICASSP*, Prague, Czech Republic, 2011.

[16] G. Tur, D. Hakkani-Tür, D. Hillard, and A. Celikyilmaz, "Towards unsupervised spoken language understanding: Exploiting query click logs for slot filling," in *Proceedings of Interspeech*, Florence, Italy, 2011.

[17] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in ATIS?" in *Proceedings of the IEEE SLT Workshop*, Berkeley, CA, 2010.

[18] P. J. Price, "Evaluation of spoken language systems: The ATIS domain," in *Proceedings of the DARPA Workshop on Speech and Natural Language*, Hidden Valley, PA, June 1990.

[19] S. Petrov and D. Klein, "Learning and inference for hierarchically split PCFGs," in *Proceedings of the AAAI*, 2007.

[20] B. Favre, D. Hakkani-Tür, S. Petrov, and D. Klein, "Efficient sentence segmentation using syntactic features," in *Proceedings of the IEEE Spoken Language Technologies (SLT) Workshop*, Goa, India, 2008.

[21] G. Tur, U. Guz, and D. Hakkani-Tür, "Model adaptation for dialog act tagging," in *Proceedings of the IEEE SLT Workshop*, 2006.

[22] P. Haffner, G. Tur, and J. Wright, "Optimizing SVMs for complex call classification," in *Proceedings of the ICASSP*, Hong Kong, April 2003.

[23] 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.