A Weakly-Supervised Approach for Discovering New User Intents from Search Query Logs

Dilek Hakkani-Tür, Asli Celikyilmaz, Larry Heck, Gokhan Tur

Microsoft Research

dilek@ieee.org, asli@ieee.org, gokhan.tur@ieee.org, larry.heck@ieee.org

Abstract

State-of-the art spoken language understanding models that automatically capture user intents in human to machine dialogs are trained with manually annotated data, which is cumbersome and time-consuming to prepare. For bootstrapping the learning algorithm that detects relations in natural language queries to a conversational system, one can rely on publicly available knowledge graphs, such as Freebase, and mine corresponding data from the web. In this paper, we present an unsupervised approach to discover new user intents using a novel Bayesian hierarchical graphical model. Our model employs search query click logs to enrich the information extracted from bootstrapped models. We use the clicked URLs as implicit supervision and extend the knowledge graph based on the relational information discovered from this model. The posteriors from the graphical model relate the newly discovered intents with the search queries. These queries are then used as additional training examples to complement the bootstrapped relation detection models. The experimental results demonstrate the effectiveness of this approach, showing extended coverage to new intents without impacting the known intents.

Index Terms: spoken language understanding, graphical models, search query click logs, intent discovery.

1. Introduction

Spoken language understanding (SLU) models aim to automatically capture and tag the semantic frames that include user intents and related concepts in real human to machine dialogs [1]. State-of-the-art SLU models are trained with examples collected for each task domain and manually annotated according to a semantic schema, often designed for each domain and task. The cycle of schema design, data collection and annotation is cumbersome and time-consuming. Rather than applying all these tedious steps, we propose a different method that utilizes the semantic space already defined and populated in a knowledge graph, such as the structured semantic knowledge graphs of the emerging semantic web, for example, *Freebase*[1].

Spoken queries to a dialog system may be classified as *informational*, *navigational*, and *transactional* in a similar way to the taxonomy for web search [2]. While informational queries seek an answer to a question, such as "*find the movies of a certain genre and director*", navigational queries aim to navigate in the dialog, such as "*go back to the previous results*", and transactional queries aim to perform an operation, such as "*play a movie*", or "*reserve a table at a restaurant*". Answers to informational queries are likely to be included in knowledge repositories. Hence the ontology of the user intents for informational queries can be formed based on the semantic web ontologies [3, 4], such as the ontology of *Freebase* or *schema.org*[2]. Furthermore, the populated knowledge in the graph can be used to mine examples that include surface forms of entities and their relations in natural language [5, 6]. In our previous work [6], for each relation type in the graph, we leveraged the complete set of entities that are connected to each other with the specific relation, and searched these entity pairs on the web. We used the snippets that the search engine returns to create natural language examples and used those as the training data for each relation. We further refined the annotations of these examples using the knowledge graph itself and iterated using a bootstrap approach.

When users are interacting with a dialog system, relations invoked in their utterances are predicted by relation detection. These can then be used to create requests in query languages (for example, in SPARQL[3] Query Language for RDF) to the knowledge graph, to create logical forms for natural language utterances [7], or to constrain slot filling and intent detection for SLU according to the invoked relations.

While the navigational intents can usually be shared across ontologies of similar dialog system applications, the ontology of user intents for transactional queries are usually defined by dialog system designers and developers, and are mainly driven by the capabilities of the back-end applications. For Internet search queries, they can also be mined from search queries [8].

In this paper, we propose a novel Bayesian hierarchical graphical model that employs search query click logs (QCL) to enrich the information included in the bootstrapped models by discovering *new* user intents, such as transactional ones. Our motivation is that while these intents are not represented in the knowledge graphs, they may have appeared in search queries. Furthermore, users of these queries click on sites that are related to their intents. For example, while a site may provide reviews of a restaurant (e.g., urbanspoon.com), another one may allow for booking tables (e.g., opentable.com). Hence, we use the clicked URLs as additional information associated with user intents in the graphical model.

Furthermore, some of the relations in the knowledge graph may also have appeared in the search query logs. We employ the bootstrap models to detect the queries that include these known relations, and use their estimated intent labels as partial, automated supervision during the training of the graphical models.

The posterior probabilities from the graphical model relate the newly discovered intents with the search queries. These queries, paired with their intent clusters are then used as additional training examples to complement the bootstrapped relation detection models. The experimental results demonstrate the effectiveness of this approach, showing extended coverage to new intents without impacting the known ones. While the in-

[1] http://www.freebase.com

[2] http://www.schema.org

[3] http://www.w3.org/TR/rdf-sparql-query/

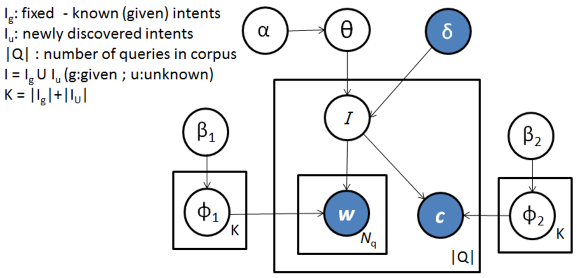

I_g: fixed - known (given) intents
I_u: newly discovered intents
|Q| : number of queries in corpus
$I = I_g \cup I_u$ (g:given ; u:unknown)
$K = |I_g|+|I_U|$
α
θ
δ
I
β1
β2
ϕ1
ϕ2
K
w
c
N_q
K
|Q|

Figure 1: Click Intent Model (CIM). Shaded circles indicate observed variables, i.e., words w, clicks c and the intent prior δ.

tent discovery part of our work is inspired from the work of [8], to the best of our knowledge, using the knowledge graph and query click logs to capture relational information about the entities in natural language utterances that entail user intents is a novel approach for conversational understanding systems.

In Section 3, we describe the click intent model (CIM) that aims to discover new user intents from search logs. Then, we explain how to bootstrap intent detection models using populated knowledge graphs, and extend the coverage of bootstrapped models with the newly discovered intents. In order to show the effectiveness of our approach, we present experimental results in section 5.

2. Related Work

Clustering of user utterances to discover semantic categories has been studied both in the context of spoken dialog systems and web search research. [9] examined two algorithms based on mutual information and Kullback-Leibler distance to cluster word sequences in utterances into concepts for slot filling. [10] used predicate and arguments from semantic role labeling to cluster unlabeled user utterances into user intents. [11] analyzed human-human interactions to identify subtasks to use in human-machine dialog management. [12] used a language modeling approach to cluster the acoustics from logged calls by their estimated semantic intents. In their work, each cluster was presented to an application developer who validated the cluster and its suggested language model and then updated the dialog application. More recently, [13] used an unsupervised, non-parametric Bayesian approach to cluster user actions in a spoken dialog corpus. On the web search side, there are a few studies similar to ours that aim to cluster web search queries to discover user intents. [14] represented search queries with their word n-grams, and employed agglomerative clustering. [15] represented the queries using their clicked URLs. [8] is the closest to our work in that they also used both words and clicked URLs within a graphical model framework. Our work is novel in two aspects: first, we introduce automatic supervision into the graphical model by way of the known intent labels from a prior bootstrapped model. Second, we use the probability distributions of each utterance over known and newly discovered intent clusters obtained from the graphical model as additional information to a spoken dialog system to study the effectiveness of the discovered clusters in understanding the user utterances.

3. Click Intent Model

We propose Click Intent Model (CIM), a hierarchical semantic clustering model that aims to capture the latent intent variables of each utterance. It is based on the assumption that the words in the user queries as well as the URLs that the users clicked on after issuing the query are related to the user's intent. We represent each query q as a vector of N_q words w_q, each of which are chosen from a vocabulary V of size $|V|$. Each query q has a clicked URL, $c_q \in C$, from the set of URLs C associated with it, and there are $|C|$ unique URLs in our corpus. In our graphical model, sampling an intent is therefore influenced by the lexical items, namely the words in a given utterance w_q, as well as the clicked URL c_q. The click information per query is an observed variable, as shown in the plate diagram in Figure 1.

Any given query is associated with a latent intent variable and we assume a fixed size K for the set of intents. Given the intent variable, the words w_q and the click c_q of the query q are generated. When users are issuing search queries, they have an intent in mind that includes a set of words as constraints. Note that, in our graphical model, we assume that the clicks and words are independent given the intent variable enabling their generation separately. In addition, although the intent variable is defined as a latent variable, there are intent classes that are known a *priori*, such as the relations covered in the knowledge graph ontology. Therefore, the users' intents can either be one of the known (given) intents, I_g, or a new one yet to be discovered, I_u, which can be sampled from any intent class. If an example query-click pair is already labeled (automatically using a bootstrap model or manually) with a user intent, we know that its intent is one of the intents in I_g, i.e., $\delta_q = k$; but if it is not labeled, then the query's intent can be any of the intents, known or new (unknown). Thus, we use a switch variable δ_q for each query to control this step (see Algorithm 1). To handle language variability and discover the hidden intents of utterances, we use the information extracted from the knowledge graph as prior information encoded with the switch variable. To this end, we classify all search queries with the models bootstrapped using the knowledge graph [6] and use the labels of the ones classified as a known intent label with a high posterior probability.

Algorithm 1 Click Intent Model - CIM

```
Draw a distribution $\theta{\sim}Dirichlet(\alpha)$, for each $k$=1,...,$K$,
draw $\phi_1{\sim}Dirichlet(\beta_1)$, and $\phi_2{\sim}Dirichlet(\beta_2)$.
**for** each query $q_j$, $j \leftarrow 1, ..., |Q|$ **do**
    –if $\delta_q$=$k$, $I_q = k$, $k \in I_g^{\dagger}$;
      else $I_q = I \sim Multinomial(\theta)$.
    –draw $c_q \sim Multinomial(\phi_2^{I_q})$.
    –**for** words $w_i$ in $q_j$, $i \leftarrow 1, ..., N_j$ **do**
        Sample $w_i{\sim}\ Multinomial(\phi_1^{I_q})$.
    **done**
**done**
```

† δ_q prior enables deterministic assignment of an intent to a query q

The generative process of our click intent model - CIM is presented in Algorithm 1. Each query q is associated with a known or unknown corpus-intent distribution θ. We use prior information represented with the δ_q parameter to determine whether to sample the query from known or unknown intent topics. Specifically, if the information about the intent of the query is known a *priori*, i.e., $\delta_q = k$, then for that query q, the intent topic is not sampled but deterministically assigned to that intent topic, $I_q = k$, $k{\in}I_g$. Each intent I is represented as a distribution over $c = 1..K$ clicks. In the CIM model, we assume that each utterance has an intent variable and a clicked-URL. Each intent variable generates a click and each word in a given query according to the multinomial intent-click distribution $\phi_2^{I_q}$ and intent-word distribution $\phi_1^{I_q}$.

3.1. Inference and Learning

The goal of inference is to predict the intent of a given query. The CIM model has K intent distributions over clicks represented with ϕ_2 and intent-word distributions ϕ_1 for each query as well a corpus variable θ indicating the intent distributions of the given queries. Previous studies [16, 17] show that the choice of inference method has negligible effect on the probability of test documents or inferred topics. Thus, to model the posterior distribution, we use a Markov Chain Monte Carlo method, specifically, Gibbs sampling. For each query q, if there is no prior information about its intent is available, we sample an intent given the rest of the clicks, words and the hyper-parameters as follows:

$$p(I_q = k|\mathbf{c}, \mathbf{w}_q, I_{-q}, \alpha, \beta_1, \beta_2, \alpha) \propto$$

$$\frac{n_q^k + \alpha}{(|Q| - 1 + K\alpha)} * \frac{n_c^k + \beta_2}{n_{(.)}^k + |C|\beta_2} * \prod_{i=1}^{N_q} \frac{n_{w_i}^k + \beta_1}{n_{(.)}^k + |V|\beta_1} \tag{1}$$

where n_q^k is the number of queries assigned to a semantic class k excluding the query q, and n_c^k is the number of times c is assigned to intent class k, and $n_{w_i}^k$ is the number of times word w_i is assigned to class k. $(.)$ indicates sum over the object, i.e., query, words in a query or clicked-URLs.

4. Relation Detection Models

Relation detection aims to determine which relations in the part of the knowledge graph related to the utterance domain has been invoked in the user utterances. For example, Figure 2 shows a part of the knowledge graph ontology that includes a set of related entity types (part a), and two example user utterances (part c) that invoke the "*Director*" relation in the knowledge graph, and basically request one of the two entities connected with this relation. The queries to the back-end for such user requests contain the "*Director*" relation. Hence, the detection of the relation as being invoked in the utterance is necessary for formulating the query to the back-end. The formulation of the complete query to the back-end requires detection of the invoked entities in the user's utterance, in addition to detecting the graph relations that are invoked. While we treat these as two separate tasks in this work, they can also be modeled jointly.

The graph ontology covers several relations between entity pairs, however, some relations may be missing. Furthermore, intents (mostly transactional intents) that operate on single entities, such as "*playing a movie*" are not represented in this framework. In this paper, we aim to identify such intents from the search query click logs, and enrich the graph and the SLU based on the graph with them (Figure 2, part b).

4.1. Bootstrapping Relation Detection Models

For bootstrapping relation detection classification, we mine training examples by searching entity pairs that are related to each other in the knowledge graph on the web, and further mine related queries from the search query click logs [6]. We refine the annotations of the mined examples via two methods that rely on other related entities on the knowledge graph and bootstrapping. As in our earlier work [3], we extract all possible entity pairs in a given domain that are connected with a specific relation from the knowledge graph, and mine patterns (such as "*movie* is a film by *person*") used in natural language realization of that relation using web search[4]. We train relation detection models using these mined patterns. We treat relation detection as a multi-class, multi-label (i.e. each utterance can invoke more than one relation) utterance classification problem, and use icsiboost [18], a Boosting based classifier, with word uni-, bi- and trigrams as features. More details on the bootstrap approach can be found in [6].

4.2. Extension to New Intents

Once clusters of new intents are estimated by the CIM, we present the most representative words for each cluster to a human annotator. Similar to [12], there are two aims in this step: (1) select well-formed clusters to include in SLU as new intents, (2) tie the associated intents to the dialog system application (for example, determine the actions the system should take when the new intent is detected in the user utterances).

To extend the coverage of the bootstrap models with the newly discovered intents, we extract the set of search queries that have a high probability of association with the selected clusters. We add these queries with their cluster labels to the training set of relation detection models and retrain.

5. Experiments

Our experiments aim to check the added value of the newly discovered intents for relation detection in a conversational understanding system for the entertainment domain.

5.1. Data Sets

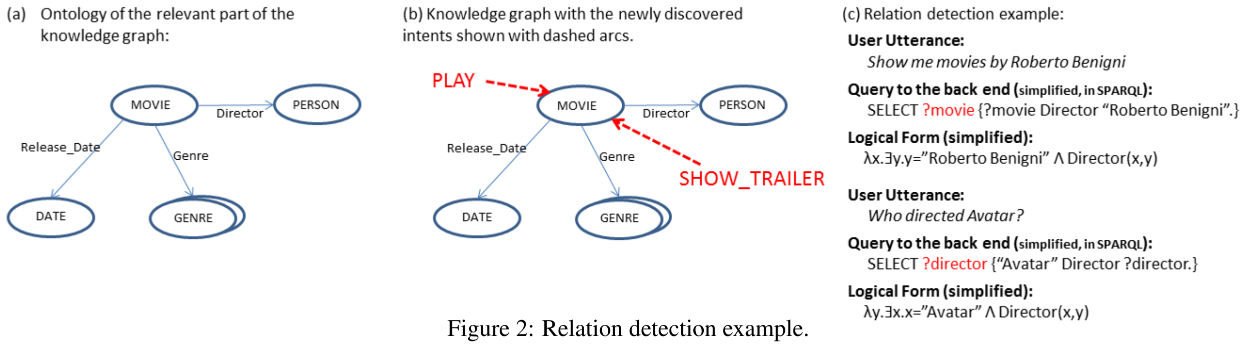
For experiments, we downloaded a set of queries that clicked on a movie related URL from QCL. Each query URL pair is also associated with the frequency of their joint appearance. There are about 81 thousand queries and 8,138 unique URLs in this set. We grouped the URLs into 415 base URLs (such as fandango.com) in order to combine similar web pages related to different entity values, such as movies and actors. We ran CIM with these queries to obtain a set of intent clusters. In these experiments, we set the number of latent variables to 20.

To see the impact of intent discovery on SLU intent detection, we experimented with spoken utterances from a *movies* domain conversational system, where users can, for example, ask to find or play a movie. We split the manually annotated data set into development (6,000 user turns) and test (5,706 user turns) sets. The annotations of these sets include on average 0.98 relation labels per utterance. Note that there are utterances in this data set that only include a movie name without any relations, such as the user request "*gone with the wind*". The bootstrap model was trained with 7 relations from the knowledge graph, which we refer to as "known" intents. These intents cover 47.0% of all the annotated labels in the development and test sets.

5.2. Evaluation

We used the 10 most probable terms as estimated by CIM for each cluster to assign it a label. In each case, the human annotator was given 1 minute to validate and select a label for all clusters. In cases where the clusters did not make sense, the annotator was allowed to discard the cluster. The selection of which clusters to use as domain related intent clusters by looking at the most probable terms and assigning them the closest intent from the test set is the only manual supervision step in our experiments. In reality, one can also use the cluster's number as the corresponding intent. However, to be able to evaluate our approach, we need to map these cluster identifiers to the relation categories that were labeled in the development and test sets when applicable, and create new labels for the others. CIM also discovered clusters that make sense for the target domain, but do not necessarily exist in the data sets. An example of such clusters is "*searching a movie theater*". While the search queries include many examples of users looking for a theater, our data sets did not include such an intent. As the builder of the dialog system may not know which user intents may be observed a priori, to be fair, we also included such clusters as new

[4] such as with http://www.bing.com


(a) Ontology of the relevant part of the knowledge graph:
MOVIE
Director
PERSON
Release_Date
Genre
DATE
GENRE
(b) Knowledge graph with the newly discovered intents shown with dashed arcs.
PLAY
MOVIE
Director
PERSON
Release_Date
Genre
SHOW_TRAILER
DATE
GENRE
(c) Relation detection example:

User Utterance:
Show me movies by Roberto Benigni

Query to the back end (simplified, in SPARQL):
SELECT ?movie {?movie Director "Roberto Benigni".}

Logical Form (simplified):
$\lambda x.\exists y.y=\text{"Roberto Benigni"} \wedge \text{Director}(x,y)$

User Utterance:
Who directed Avatar?

Query to the back end (simplified, in SPARQL):
SELECT ?director {"Avatar" Director ?director.}

Logical Form (simplified):
$\lambda y.\exists x.x=\text{"Avatar"} \wedge \text{Director}(x,y)$

Figure 2: Relation detection example.

intents in the training of the relation models.

Table 1 summarizes the results of our experiments on manual transcriptions of the user utterances. We grouped the relations discovered by all models compared in the experiments into "new" and averaged per-class F-measure for each of them. When a specific intent was not discovered by the corresponding model, we still included per-class F-measure for that class in averaging, to have a fair comparison of different models.

There are in total 8 clusters that were discovered by at least one of the clustering models, validated by the human user, and added to the relation detection models. 1 of these (*movie_cast*) is already included amongst the known intents. In the experiments, we included data mined from QCL for this relation in training, as this can enhance the relation models for this category. 4 of the new intents (*movie_review, movie_content, play_movie, play_trailer_of_movie*) were similar to categories covered in the test sets. Similar to [19], we mapped the labels of these clusters to the similar one in the test set for evaluation purposes. 3 of the new intents were not included in the development and test sets (for example, *find_theater*).

The bootstrap model is the best single (i.e. not combined) model used in our previous work [6], referred to as"Pattern from Snippets (1 iter)" in that work. The last row, "Crowd-sup." is a supervised model trained using 8,000 examples that were collected through crowd sourcing and annotated by expert labelers. CIM is the original model that does not include any known relation labels. CIM-W includes labels from the bootstrap models as automatic supervision. CIM-N also lacks relation labels for queries, however, in this one, word sequences corresponding to entities in a gazetteer were canonicalized by replacing them with the entity type (for example, "*show me avatar*" is converted to "*show me movie-name*"). The gazetteers were obtained by taking all entities in the populated knowledge graph, weighing them using query click logs [20], and filtering the ones with low weights. CIM-NW is the model that includes both automatic supervision and entity canonicalization. For the known categories, all models perform better than the supervised baseline with limited training data. The best F-measure on the test set for the known categories is 49.1% as obtained by CIM-NW. This result is also better than the original bootstrap model targeting the known relations, as part of the clusters validated by the human labeler corresponded to known relations. CIM-N resulted in the best average per class F-measure of 36.9% for the newly discovered relations.

The experimental results included were chosen according to the parameters that optimized the performance on the development set. We also tried different values for the following parameters:

Using more examples from QCL: We also experimented with larger query data sets, up to 180K queries from the QCL with the CIM method.

	Dev		Test	
	MF-kn	MF-new	MF-kn	MF-new
Bootstrap	52.1%	11.7%	48.0%	12.0%
CIM	50.5%	27.6%	46.7%	25.4%
CIM-W	50.4%	30.0%	46.6%	27.3%
CIM-N	51.4%	**36.7%**	47.9%	**36.9%**
CIM-NW	**52.6%**	35.6%	**49.1%**	30.0%
Crowd-sup.	42.7%	39.4%	45.5%	42.1%

Table 1: Macro-averaged, per class F-measures for previously known (MF-kn) and newly discovered (MF-new) classes with the bootstrap model, CIM, CIM-N, CIM-NW, and the supervised models.

Increasing the number of latent variables: We experimented with 20, 40, and 60 clusters, and found 20 to be the optimum. In the worst case, there can be as many clusters as examples, but this would be similar to the supervised approach of labeling all queries with the relations invoked in them.
Number of queries used in training: We experimented with 2000, 4000, and 6000 utterances representing the newly discovered intents corresponding to the latent intent clusters of the CIM model.

6. Conclusions and Future Work

We presented an unsupervised approach to discover new user intents using a novel Bayesian hierarchical graphical model. The model employs search query click logs to enrich the information extracted from models bootstrapped using populated knowledge graphs and web search. We use the URLs that search users click as implicit supervision in clustering and extend the knowledge graph based on the relational information discovered from this model. The posterior probabilities from the graphical model relate the newly discovered intents with the sear¡ch queries. These queries are then used as additional training examples to complement the bootstrapped relation detection models. We experiment with various automatic ways of marking the queries with information from the knowledge graph. We compare this approach with a supervised learning approach based on crowd-sourced data, and show that the new approach is effective in discovering new user intents and building models with comparable performance, as well as improving the performance on categories previously found in the knowledge graph. As a future work, we plan to sample natural language like queries from the search query logs, instead of using all queries. Furthermore, the current approach does not consider the slot values in each query. The gazetteer-based entity extraction can be extended with a slot filling approach.

Acknowledgments: We would like to thank our colleagues Umut Ozertem and Patrick Pantel for providing useful insights.

7. 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] A. Broder, "A taxonomy of web search," in *ACM SIGIR Forum*, 2002, pp. 3–10.

[3] L. Heck and D. Hakkani-Tür, "Exploiting the semantic web for unsupervised spoken language understanding," in *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT 2012)*, Miami, Florida, 2012.

[4] G. Tur, M. Jeong, Y.-Y. Wang, D. Hakkani-Tür, and L. Heck, "Exploiting semantic web for unsupervised statistical natural language semantic parsing," in *Proceedings of Interspeech 2012*, Portland, Oregon, 2012.

[5] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL*, 2009.

[6] D. Hakkani-Tür, L. Heck, and G. Tur, "Using a knowledge graph and query click logs for u nsupervised learning of relation detection," in *Proceedings of the ICASSP*, 2013.

[7] L. Zettlemoyer and M. Collins, "Online learning of relaxed CCG grammars for parsing top logical form," in *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, Czech Republic, 2007.

[8] T. Lin, P. Pantel, M. Gamon, A. Kannan, and A. Fuxman, "Active objects: Actions for entity-centric search," in *Proceedings of World Wide Web Conference (WWW-12)*, Lyon, France, 2012, pp. 589–598.

[9] A. Chotimongkol and A. Rudnicky, "Automatic concept identification in goal oriented conversations," in *Proceedings of ICSLP*, 2002.

[10] G. Tur, D. Hakkani-Tür, and A. Chotimongkol, "Semi-supervised learning for spoken language understanding using semantic role labeling," in *Proceedings of ASRU*, 2005.

[11] S. Bangalore, G. DiFabbrizio, and A. Stent, "Towards learning to converse: Structuring task-oriented human-human dialogs," in *Proceedings of ICASSP*, 2006.

[12] X. Li, A. Gunawardana, and A. Acero, "Unsupervised semantic intent discovery from call log acoustics," in *Proceedings of ICASSP*, 2005.

[13] D. Lee, M. Jeong, K. Kim, and G. G. Lee, "Unsupervised modeling of user actions in a dialog corpus," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[14] J. Cheung and X. Li, "Sequence clustering and labeling for unsupervised query intent discovery," in *Proceedings of WSDM*, 2012.

[15] J. Yi and F. Maghoul, "Query clustering using click-through graph," in *Proceedings of WWW*, 2009.

[16] I. A. Asuncion, M.Welling, P. Smyth, and Y. Teh, "On smoothing and inference for topic models," in *UAI*, 2009.

[17] H. Wallach, "Structured topic models for language," Ph.D. dissertation, University of Cambridge, 2008.

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

[19] B. Zhang, B. Hutchinson, W. Wu, and M. Ostendorf, "Extracting phrase patterns with minimum redundancy for unsupervised speaker role classification," in *Proceedings of HLT/ACL*, 2010.

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