Leveraging Knowledge Graphs for Web-Scale Unsupervised Semantic Parsing

Larry Heck, Dilek Hakkani-Tür, Gokhan Tur

Microsoft Research

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

Abstract

The past decade has seen the emergence of web-scale structured and linked semantic knowledge resources (e.g., Freebase, DBPedia). These semantic knowledge graphs provide a scalable "schema for the web", representing a significant opportunity for the spoken language understanding (SLU) research community. This paper leverages these resources to bootstrap a web-scale semantic parser with no requirement for semantic schema design, no data collection, and no manual annotations. Our approach is based on an iterative graph crawl algorithm. From an initial seed node (entity-type), the method learns the related entity-types from the graph structure, and automatically annotates documents that can be linked to the node (e.g., Wikipedia articles, web search documents). Following the branches, the graph is crawled and the procedure is repeated. The resulting collection of annotated documents is used to bootstrap web-scale conditional random field (CRF) semantic parsers. Finally, we use a maximum-a-posteriori (MAP) unsupervised adaptation technique on sample data from a specific domain to refine the parsers. The scale of the unsupervised parsers is on the order of thousands of domains and entity-types, millions of entities, and hundreds of millions of relations. The precision-recall of the semantic parsers trained with our unsupervised method approaches those trained with supervised annotations.

Index Terms: semantic parsing, semantic web, semantic search, dialog, natural language understanding

1. Introduction

Spoken language understanding (SLU) has seen considerable advancements over the past two decades [1]. While understanding language is still considered an unsolved problem, a variety of practical goal-oriented SLU systems have been built for limited domains. These systems aim to automatically identify the intent of the user as expressed in natural language, extract associated arguments or slots, and take actions accordingly to satisfy the user's requests. In such systems, the speaker's utterance is typically recognized using an automatic speech recognizer. Then the intent of the speaker is identified from the recognized word sequence using a SLU component. Subsequent to the SLU processing, a dialog or task manager interacts with the user to help the user achieve their desired task.

Most state-of-the-art SLU systems are based on statistical methods such as conditional random field (CRF) semantic parsers [2, 3] and discriminatively trained intent and domain detectors (e.g., [4],[5]). But recently, as more SLU systems are being deployed, statistical methods have been shown to have their limitations. State-of-the-art statistical SLU systems require tasks to be limited in scope; the SLU is performed over a small number of narrowly defined, known domains, with handcrafted domain-dependent schemas (ontologies). In addition, high accuracy of statistical SLU methods rely on *supervised* training instances (i.e., the instances are manually labeled with the true domains, intents, slots). These characteristics of statistical SLU systems have forced developers to spend considerable energy crafting one domain at a time and ultimately limit the ability of the systems to scale in breadth of domains and external knowledge sources, as well as remain flexible to changes in task definition.

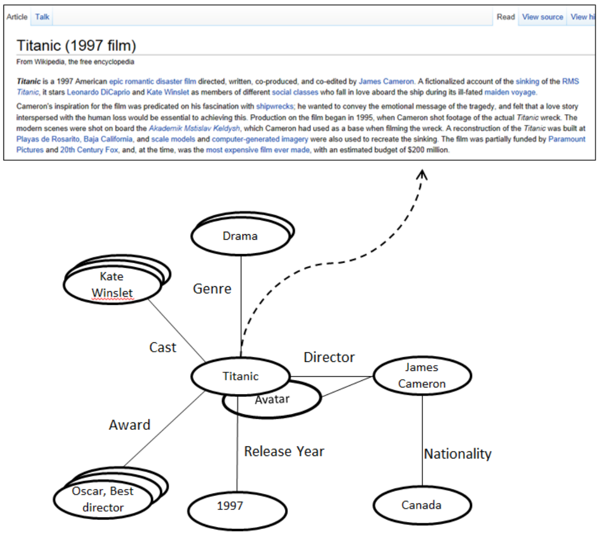
As a result, there has been an increased level of research over the past several years to address these limitations. New *lightly supervised* and *unsupervised* training methods rely on side information to automatically provide training labels (domain, intent, and slots). An example is our previous work on leveraging web search query click logs for mining additional training data and enriching classification features. With these methods, we have seen significant reductions in domain [6], intent [7], and slot filling [8] error rates.

In our recent work, we have begun to exploit the combination of statistical approaches with methods inspired by the deep semantic methods from the AI community. These methods are made possible by the emergence of semantically rich representations of knowledge graphs (the so-called *semantic web*) created by the large web search companies. Prior work leveraging semantic graphs for conversational system did not leverage the relational structure of the graphs, but rather used them as database resources to construct named lists of entities [9, 10]. In [11], we exploit the semantic structure of the web pages users visited when completing tasks. In [12], we used semantic graphs for unsupervised intent detection. And in [13], we leverage semantic graphs for unsupervised entity relation classification.

In this paper, we focus on entity extraction. We leverage web-scale semantic graphs to bootstrap a web-scale semantic parser with no requirement for semantic schema design, no data collection, and no manual annotations. We align the knowledge populated in the semantic graph with the related documents, and transfer entity annotations. We use these to bootstrap models and further improve them by combining with gazetteers mined from the knowledge graphs and adapting them to the target domain with a MAP-style algorithm. Section 2 provides background on techniques that we leverage in the paper. The new unsupervised semantic parsing methods are described in Sections 3 to 5. Finally, in Section 6, we present experimental results showing the performance of the new approach.

2. Refining Gazetteers with Web Search

Gazetteers (entity lists), such as lists of movie or actor names, are important features for spoken language understanding. They provide coverage and recall for domains that have large lists of specific entity instances. For example, gazetteers can be used to represent the list of the 11M music release track titles in a knowledge base such as Freebase.



Titanic (1997 film)

From Wikipedia, the free encyclopedia

Titanic is a 1997 American epic romantic disaster film directed, written, co-produced, and co-edited by James Cameron. A fictionalized account of the sinking of the RMS *Titanic*, it stars Leonardo DiCaprio and Kate Winslet as members of different social classes who fall in love aboard the ship during its ill-fated maiden voyage.

Cameron's inspiration for the film was predicated on his fascination with shipwrecks; he wanted to convey the emotional message of the tragedy, and felt that a love story interspersed with the human loss would be essential to achieving this. Production on the film began in 1995, when Cameron shot footage of the actual *Titanic* wreck. The modern scenes were shot on board the *Akademik Mstislav Keldysh*, which Cameron had used as a base when filming the wreck. A reconstruction of the *Titanic* was built at Playas de Rosarito, Baja California, and scale models and computer-generated imagery were also used to recreate the sinking. The film was partially funded by Paramount Pictures and 20th Century Fox, and, at the time, was the most expensive film ever made, with an estimated budget of $200 million.

Figure 1: An example portion of the Freebase semantic graph related to the movie domain

While gazetteers yield high recall in entity extraction, they can often reduce precision. This is because large named entity lists often contain many ambiguous, confusable, or incorrect phrases. For example, lists of recent movies will contain the named entity "Up" (the 2009 American 3D computer-animated comedy). But an entry such as "Up" is likely confusable with the direction "Up" and could generate many false positive matches.

Leveraging our previous work, we employ a method to construct gazetteers that significantly improve precision while maintaining much of the recall of unprocessed named entity lists [14]. The method learns from user clicks in web search logs by comparing click distributions of an entity to aggregate click distributions of random phrases. This yields a cross-entropy quality score for each entity. A threshold is then applied to remove all low scoring entries of the list. This method can be applied to any typed list that generates a reasonably large set of web search activity. Experiments on large-scale movie and restaurant entity exraction in spoken language understanding show 10-15% relative improvements in the F-measure.

3. Unsupervised Data Mining with Knowledge Graphs

Our approach to web-scale unsupervised learning in spoken language understanding is based on a graph crawling algorithm over large-scale semantic knowledge graphs. Semantic graphs are defined by a schema and composed of nodes and branches connecting the nodes. The nodes represent entity-types. An example from Freebase is shown in Figure 1, where nodes represent core entity-types for the movie domain. Domains in Freebase span the web, from "American Football" to "Zoos and Aquariums"[1]. The branches that connect the nodes reprepresent relations between the entity-types. These are called *Properties* in Freebase. As shown in Figure 1, relations include examples like "Director", "Cast", and "Release Year".

Our new unsupervised graph crawling algorithm is summarized as a sequence of the following 6 steps.

1. **Initialize the Crawl**
   The procedure starts by selecting an initial node (entity-type) of the graph. We will refer to this node as the "Central Pivot Node" (CPN). The CPN should be one of the primary entity-types of the domain of interest.
2. **Retrieve Sources of NL Surface Forms**
   From an instance entity of the CPN, retrieve documents related to the entity instance. As illustrated in Figure 1, we could retrieve the Wikipedia page for the movie "Titanic", as well as documents from other sources such as documents returned from a web search of the entity surface form (SF). These documents are used as sources of natural language surface forms of the CPN's entity.
3. **Annotate with 1st-Order Relations**
   Form a gazetteer from the 1st-Order relations of a specific entity instance of the CPN. For the graph of Figure 1, the gazetteer for the CPN "Titanic" includes the movie name and 56 surface forms of entities from the movie's 1st-Order relations (with 5 of the relations shown in the figure). This results in a small, high-precision gazetteer. Use this gazetteer to automatically annotate the sentences from the retrieved documents, where the sentences are extracted using available sentence extraction/splitting methods[15]. The annotation is achieved by using a greedy, longest-string pattern match. These annotations will be used as "truth" labels for subsequent (statistical) training passes.
4. **Extract Features With Large-Scale Entity Lists**
   For the CPN and each of its K related properties, enumerate all possible entity instances and form large-scale gazetteers. For the example in Figure 1, this yields a gazetteer of all 275K movies (the CPN) and 56 gazetteers for all actors, directors, etc. in the movie industry. We use the web search-based refining methods described in Section 2 to increase the quality of these lists (cleaning, sorting, and removing ambiguous entries).

   With each of the K gazetteers, use the same longest string matching algorithm as used in the previous step and re-annotate the documents of the CPN. But, in this step, the (lower precision, higher recall) annotations will be used as features in subsequent (statistical) training passes.
5. **Annotate with Higher-Order Relations**
   Extending to higher order relations (2nd, 3rd, etc.), repeat Steps 3 and 4. Higher order relations refer to chained relationships of entities. In Figure 1, the director James Cameron has a 1st-Order relation and his nationality has a 2nd-Order relationship to the movie Titanic

   $$Movie \rightarrow Director \rightarrow Nationality$$

   Sentences from the documents of the original CPN are re-annotated with the entities of the higher order relations in a multi-pass approach. While there are a number of stopping criteria from research on the general area of graph crawling algorithms[16, 17], the most direct and simplest is to set a threshold on the convergence of SLU quality (e.g., F-measure) using a held-out development dataset.
6. **Crawl Graph to Select New CPN and Repeat**
   From the initial CPN, crawl to (or enumerate) each related entity-type, select a new CPN, and repeat the procedure in Steps 1-5.

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

4. Modeling Entities for Semantic Parsing

The data mining procedure of the previous section provides high precision-recall labels for a large set of sentences over a broad set of entities *without* supervision. Using the Wikipedia source alone, we can generate millions of auto-annotated sentences that then can be used to bootstrap statistical semantic parsers. With this volume of data, we can bootstrap state-of-the-art statistical methods for entity extraction.

For this work, we frame the entity extraction (slot filling) task as a sequence classification problem to obtain the most probable entity sequence:

$$\hat{Y} = \operatorname*{argmax}_{Y} P(Y|X)$$

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 from the set of slots/concepts C. Following the state-of-the-art approaches for entity extraction (e.g., [2, 3]), we use discriminative conditional random fields (CRFs)[18] for modeling.

CRFs are shown to outperform other classification methods for sequence classification [1], since the training can be done discriminatively over a sequence. The baseline model relies on word n-gram based linear chain CRF, imposing the first order Markov constraint on the model topology. Similar to maximum entropy models, in this model, the conditional probability, $P(Y|X)$ is defined as [18]

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

with the difference that both X and Y are sequences instead of individual local decision points given a set of features f_k (such as n-gram lexical features, state transition features, or others) with associated weights λ_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.

The features generated from the data mining procedure of Section 3 include lexical n-gram and gazetteer labeled sequences. Given the induced "truth" labels from Step 3 of the procedure and these features, we train CRF models to extract the entities. The resulting CRF model can then be adapted to the target domain in an unsupervised fashion using a MAP-style method after tagging with their gazetteer entries.

5. Modeling Relations for Semantic Parsing

Extending our previous work on entity-relation modeling [13], we propose a new method for semantic parsing of entities based on the entity-relation-entity structure of the semantic graph. Our methods for relation modeling automatically generate a ranked list of the most important entity-relation-entity patterns through unsupervised mining. We leverage these patterns to induce semantic parsing grammars or templates, and then use the templates to spot entities. In the movies domain, for example, one of the most common templates to relate a movie with its director is:

ent(movie name) → *rel(directed by)* → *ent(director)*

An example sentence matching this template is "the movie Avatar was directed by James Cameron".

In this paper, we use the grammars of the entity-relation-based parser as a final pass after the entity extraction parsing. The intuition behind this approach is that the template matching assumes known entity-relation occurances. We use the procedure described in the previous section to spot high confidence entities, serving as anchor points for the template-based relation grammars.

6. Experiments and Results

To experimentally test the unsupervised learning approaches presented in this paper, we simulate a scenario where a developer seeks to train a SLU system for a NL movie search application (e.g., Netflix). The developer has no prior data, models, annotations, or schema for the movies domain.

The experiments test two conditions represented by a *mined* testset and a *control* testset. The mined dataset is a development corpus derived from Wikipedia. It was constructed by selecting 1000 sentences to be held out from the training dataset (neither the sentence nor the topics were seen in training). This testset represents a *matched* condition to the training data; the testset is derived from the same domain and language style of the training development corpus.

The *control* testset represents the target application of the developer. For the experiments in this paper, we collected 2000 sentences from a movie retrieval application. Example sentences include "Show me movies with James Bond", "Show me movies for a two year old", and "Find me some chick flicks". This testset represents the natural language expressions and style of the target user interactions. Because the procedure boostraps from Wikipedia, the control dataset is a *mismatched* condition to the training data.

To initiate the unsupervised parsing procedure described in Section 3, we select the entity type *Movie* as the starting central pivot node (CPN). For this entity-type, there are approximately 175K movie names and 56 related entity-types for each movie in Freebase. Examples of these entity-types are listed in Table 1 with their corresponding NL surface form.

Entity Type	NL Surface Form
name	"Avatar"
initial_release_date	"12/10/2009"
directed_by	"James Cameron"
produced_by	"Jon Landau", "James Cameron"
written_by	"James Cameron"
music	"James Horner"
starring	"Sigourney Weaver",...

Table 1: Example entity types and NL surface forms from the film domain in Freebase

To complete Step 4 of the the unsupervised procedure in Section 3 "Extract Features With Large Scale Lists", we construct high-quality gazetteers for the initial annotations using the related properties of the CPN. While 56 gazetteers were constructed (corresponding to the related properties of the movie CPN in Freebase), only four types are present in the Control (movie search) test data and are shown in the table : Movie (name), Actor, Genre, and Director. The size of the original entity lists are approximately 175K (Movies), 234K (Actors), 685 (Genre), and 59K (Director). We used the procedure described in Section 2, and experimentally determined the best threshold on the cross-entropy quality score. Table 2 shows results from the the performance of semantic parsing using only the large-scale gazetteers. F-measures are shown for the 1000 sentence testset mined from Wikipedia.The gazetteers were constructed using web search click distributions, sorting according to the

	Manual Transcriptions					ASR				
	Movie	Actor	Genre	Director	All	Movie	Actor	Genre	Director	All
Supervised (Lex. + Gazetteers)	51.25%	86.29%	93.26%	64.86%	66.53%	45.15%	82.56%	88.58%	58.59%	60.96%
Supervised (Lex. Only)	46.44%	80.22%	92.83%	52.94%	61.72%	39.21%	74.86%	86.21%	45.36%	54.10%
CRF Lex.	0.19%	9.67%	0.00%	62.39%	5.61%	0.20%	9.67%	0.00%	57.14%	5.27%
Gazetteers	69.69%	50.70%	15.76%	2.63%	51.14%	59.66%	47.78%	11.80%	2.82%	43.88%
CRF Lex. + Gaz.	1.96%	72.35%	4.73%	79.03%	31.94%	1.74%	69.76%	3.57%	75.00%	30.77%
CRF Lex. + Gaz. + Adapt	71.72%	58.61%	29.55%	77.42%	60.38%	55.74%	62.70%	30.95%	73.21%	54.69%
CRF Lex. + Gaz. + Adapt + Rel.	-	-	-	**84.62%**	**61.02%**	-	-	-	**80.67%**	**55.40%**

Table 3: F-Measure performance of the unsupervised methods developed in this paper

cross entropy of gazetteer entity surface forms versus 100K random web search queries. F-measure results are shown for the Top $N\%$ of scored gazetteer entities. The best results are obtained with the Top 60% gazetteer, yielding an unsupervised F-measure of 32.84% on the 1000 sentence Wikipedia testset.

	Movie	Actor	Genre	Director	All
Top 20%	29.51%	3.03%	20.00%	17.14%	16.52%
Top 40%	30.21%	30.00%	5.52%	27.03%	23.22%
Top 60%	**28.57%**	**56.59%**	**6.62%**	**40.00%**	**32.84%**
Top 80%	14.90%	59.09%	6.58%	47.62%	24.56%
Top 100%	6.66%	31.54%	12.31%	44.12%	13.14%

Table 2: F-measure performance of semantic parsing using gazetteers only.

While the results in Table 2 are promising, it should be noted that these are for matched conditions only: training and testing on Wikipedia sentences. To explore the effect of mismatched training-testing conditions, Figure 2 shows results from the unsupervised annotations obtained with the 1st-Order relations of Step 3 of Section 3, followed by training the CRF semantic parser. Learning curves are shown on the F-Measures for varying amounts of Wikipedia training data for the matched *Mined* and mismatched *Control* testsets. Only lexical trigram features are used. The unsupervised F-measures are compared to supervised annotations (approximately 61.72% F-Measure). As can be seen, the impact of the mismatched training on Wikipedia and testing on the movie search application is significant. There is a drop of more than 25% in F-measure when training with Wikipedia sentences and testing with movie search queries, compared to training and testing with Wikipedia sentences.

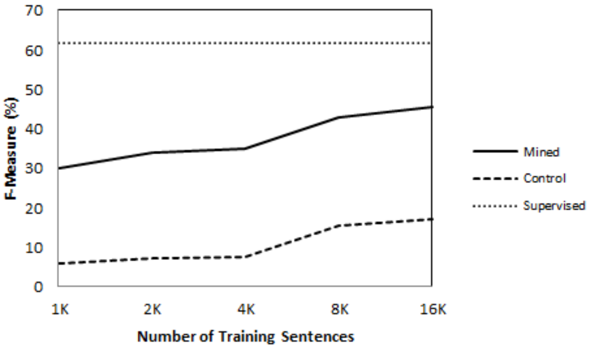


Figure 2: Mined dataset

Table 3 summarizes the F-Measure performance of the unsupervised methods developed in this paper. All conditions shown are for the Control dataset, with the first set of results on the left from manual transcriptions, and the second set on the right from automatic speech recognition (ASR) @18.5% word error rate. The results are shown for 1st-Order relations only.

The row labeled "CRF Lex." shows results for the unsupervised CRF models trained with lexical features only. With the exception of the Director class, the results highlight the lack of robustness of the models to the mismatched training-testing (Wikipedia vs. movie search) case. Given the relative robustness of the gazetteer-only annotation models in the next row, labeled as "Gazetteers", the results suggest that the CRF training is "overfitting", learning to associate surrounding words/phrases in Wikipedia to specific entities.

To address the mismatch case, we completed an additional unsupervised training iteration by composing a new training dataset with labels from: (a) CRF Lex. + Gaz model (1000 Wikipedia sentences) and (b) the Gaz only model (2000 movie search sentences). The unlabeled movie search sentences were from a held-out development dataset, separate from the testset. The results of this adaptation procedure are shown in the row "CRF Lex. + Gaz. + Adapt". The procedure significantly improves the F-measure for the Movie and Genre classes, as well as improves the overall F-measure by 29% absolute.

Finally, we performed one last pass on the annotations of the Control (movie search) testset using our new relation modeling approach described in Section 5. In the experiments shown in the last row of Table 3, the method automatically induced grammars for the "directed by" relation. This yielded an increase of over 7% in F-Measure for the Director class (manual and ASR transcriptions), and an increase of approximately 0.5-1% overall.

With the combination of all unsupservised methods of this paper, we nearly match the F-measures of supervised training. We get 61.02% and 55.40% for manual and automatic transcriptions, respectively. This is within 5.5% of the performance achieved with supervised training.

7. Summary and Conclusions

We present an new approach for unsupervised semantic parsing with semantic knowledge graphs with no requirement for semantic schema design, no data collection, and no manual annotations. We develop a graph crawling algorithm for data mining, and two entity extraction approaches: a CRF-based method with unsupervised MAP adaptation, and a relation model with induced entity extraction grammars. The combined methods give F-measures of 61.02% and 55.40% for manual and automatic transcriptions, respectively, which is within 5.5% and comparable to the performance achieved with supervised training.

Future work will experimentally investigate the impact of higher-order knowledge graph relations on semantic parsing. Also, we will extend the methods for semantic parsing based on relation modeling to target a greater number of the relations of the semantic graph.

8. References

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

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

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

[4] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet, "icsiboost," http://code.google.come/p/icsiboost, 2007.

[5] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[6] Dilek Hakkani-Tür, Larry Heck, and Gokhan Tur, "Exploiting web search query click logs for utterance domain detection in spoken language understanding," in *Proceedings of ICASSP*, 2011.

[7] Asli Celikyilmaz, Dilek Hakkani-Tür, and Gokhan Tur, "Leveraging web query logs to learn user intent via bayesian latent variable model," in *Proceedings of the ICML Workshop on Combining Learning Strategies to Reduce Label Cost*, 2011.

[8] Gokhan Tur, Dilek Hakkani-Tür, Dustin Hillard, and Asli Celikyilmaz, "Towards unsupervised spoken language understanding: Exploiting query click logs for slot filling," in *Proceedings of Interspeech*, 2011.

[9] Masahiro Araki and Kenji Tachibana, "Multimodal dialog description language for rapid system development," in *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*. Association for Computational Linguistics, 2009, pp. 109–116.

[10] M. Araki and D. Takegoshi, "Framework for the development of spoken dialogue system based in collaboratively constructed semantic resources," in *Proceedings of SDTCD (NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data)*, 2012.

[11] Gokhan Tur, Minwoo Jeong, Ye-Yi Wang, Dilek Hakkani-Tür, and Larry Heck, "Exploiting semantic web for unsupervised statistical natural language semantic parsing," in *Proceedings of Interspeech*, 2012.

[12] Larry Heck and Dilek Hakkani-Tür, "Exploiting the semantic web for unsupervised spoken language understanding," in *Proceedings of the IEEE SLT Workshop*, Miami, FL, 2012.

[13] 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.

[14] Dustin Hillard, Asli Celikyilmaz, Dilek Hakkani-Tür, and Gokhan Tur, "Learning weighted entity lists from web click logs for spoken language understanding," in *Proceedings of Interspeech*, Florence, Italy, 2011.

[15] D. Gillick, "Sentence boundary detection and the problem with the u.s.," in *Proceedings of the NAACL*, Boulder, CO, 2009.

[16] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the 7th International World Wide Web Conference*, 1998, vol. 30, pp. 107–117.

[17] J. Cho, H.Garcia-Molina, and L. Page, "Efficient crawling through url ordering," in *Proceedings of the 7th International World Wide Web Conference*, 1998, vol. 30, pp. 161–172.

[18] 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.