Exploiting the Semantic Web for Unsupervised Natural Language Semantic Parsing

Gokhan Tur, Minwoo Jeong, Ye-Yi Wang, Dilek Hakkani-Tür, Larry Heck

Microsoft, USA

Abstract

In this paper, we propose to bring together the semantic web experience and statistical natural language semantic parsing modeling. The idea is that, the process for populating knowledge-bases by semantically parsing structured web pages may provide very valuable implicit annotation for language understanding tasks. We mine search queries hitting to these web pages in order to semantically annotate them for building statistical unsupervised slot filling models, without even a need for a semantic annotation guideline. We present promising results demonstrating this idea for building an unsupervised slot filling model for the movies domain with some representative slots. Furthermore, we also employ unsupervised model adaptation for cases when there are some in-domain unannotated sentences available. Another key contribution of this work is using implicitly annotated natural-language-like queries for testing the performance of the models, in a totally unsupervised fashion. We believe, such an approach also ensures consistent semantic representation between the semantic parser and the backend knowledge-base.

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

1. Introduction

Natural language understanding (NLU) in goal-oriented dialog systems aims to automatically identify the domain and intent of the user, as expressed in natural language (NL), and to extract associated arguments or slots [1]. The pioneering DARPA-sponsored ATIS (Air Travel Information System) Project [2], for example, has focused on slot filling, for the airline domain. In this task, users request flight information, such as *I want to fly to Boston from New York next week*. In this case, understanding was reduced to the problem of extracting task specific arguments in a given frame-based semantic representation involving, for example, *Destination* and *Departure Date*. Another example of semantic parsing from the movies domain is presented in Table 1. While the concept of using semantic frames is motivated by the case frames used in artificial intelligence research, in this instance the slots are very specific to the target domain and finding target values within automatically recognized spoken utterances can be difficult due to automatic speech recognition errors and poor modeling of natural language variability. For these reasons, spoken language understanding researchers employed known classification methods for filling frame slots from the application domain using a given training data set and performed comparative experiments. These approaches used generative models such as hidden Markov models [3], discriminative classification methods [4, 5, 6] and probabilistic context free grammars [7, 8].

The state-of-the-art approach for training NLU models relies on supervised machine learning methods. An exhaustive survey of intent determination and slot filling methods can be found in [1]. These models require a large number of in-domain sentences which are semantically annotated by humans, a very expensive and time consuming process. Additionally, NLU models require in-domain gazetteers (such as city, movie, actor, or restaurant names) for better generalization. However, populating and maintaining these gazetteers, which are typically very dynamic and need constant maintenance, requires a significant amount of manual labor and typically semi-automated knowledge acquisition techniques are employed. For instance, Li et al. exploited query click logs leveraging domain-specific structured information for web query tagging [9] and built semi-supervised models using the derived labels. Wang et al. leveraged structured HTML lists to automatically generate gazetteers [10]. These gazetteers were then used to improve slot filling models. Liu et al., extending this approach, proposed automatically populating gazetteers from the web queries to be used in slot filling [11]. Starting from a seed gazetteer, they mined query click logs to expand it using a generative model. They learned target websites where users clicked 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. In our previous work we also used statistical methods to weigh the gazetteer entries [12]. However, 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.

Utterance	*show me recent action movies by spielberg*
Domain:	Movie
Genre:	*action*
Date:	*recent*
Director:	*spielberg*

Table 1: An example utterance with semantic annotations.

In our previous work [13], we proposed extending these gazetteer population techniques to mine unannotated training data for semantic parsing. Instead of stripping out the context words found in candidate entities, they were used to train slot filling models. This also eliminates the need to maintain gazetteers and this method can be combined with any available annotated data to improve performance.

In this paper, we significantly extend this idea, also exploiting the semantic structure of the web pages the users clicked for their queries. We see this as an initial effort towards bringing together the extensive, yet complementary semantic web experience with statistical natural language semantic parsing. For instance, a query like "*showtimes for hugo by scorsese*", resulting in a click to the IMDB web page of this movie, can be easily parsed for not only the movie name but also the director, exploiting the semantic structure of the landing web page. This

is also the case for many other movie-related web sites such as rottentomatoes.com or netflix.com, and also for queries belonging to other domains, such as restaurant queries going to urbanspoon.com, book queries going to barnesandnoble.com, or financial queries going to finance.yahoo.com. This implicit annotation provided by the semantic web enables us to better mine queries, build better bootstrap slot filling models, and finally evaluate them.

In the next section, we present our slot filling system. Then in Section 3, we present the proposed approach in detail. Then in Section 4, we present the experiments and results.

2. Semantic Parsing

Following the state-of-the-art approaches for slot filling [5, 6, among others], we use discriminative statistical models, namely conditional random fields, (CRFs) [14], for modeling. More formally, slot filling is framed as a sequence classification problem to obtain the most probable slot sequence:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\, p(Y|X)$$

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 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 [14]:

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

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.

In this study, we follow the popular IOB (in-out-begin) format in representing the data as shown below.

3. Unsupervised Slot Filling Model Building and Evaluation

The state-of-the-art slot filling approaches rely on semantically annotated natural language data and/or semantic grammars, based on carefully designed semantic templates. While there have been a number of generic natural language understanding semantic frames, such as FrameNet [15], for "targeted" or goal-oriented understanding tasks, researchers have preferred to define task-specific templates (optionally using generic frames as feedback or features). In the case of the well-known ATIS system, the project participants contributed to the design of the semantic template, which consists of slots like departure city, arrival city, airline, date, and time. From a few thousand sentences, they came up with 79 different slots, which can be clustered into 44 unique categories (e.g., destination city or destination airport code can be grouped together).

When a new slot filling model needs to be built for a new domain, the typical first step is to design the semantic template

Figure 1: A IMDB movie web page

based on some seed sentences, defining the scope of the task. For example, if the domain is movies, one can think of actor, director, movie name, genre, or release date as potential slots. Assuming one slot filler per atomic domain (ignoring hierarchical domains), the problem we tackle in this paper is whether we can build bootstrap slot filling models instantly without any manual effort. This problem consists of following sub-problems:

- how to define a semantic template for that domain
- how to mine relevant data
- how to annotate that data
- how to evaluate the resulting model

While this is basically a holy grail problem of NLU, we believe it is doable to a certain extent. The rest of this section describes the algorithms tackling each of these sub-problems.

3.1. Semantic Parsing of Target Web Pages

Our approach relies heavily on structured web pages for the target domain, as mentioned above. Figure 1 shows the IMDB web page for the movie *Hunger Games*, for example. It is straightforward to semantically parse pages like this using simple patterns to extract key information about the movie.

Since there is no domain or task specific semantic template ready, the solution proposed here relies on the extensive complementary literature on the semantic web [16, 17] and semantic search [18]. In 1997, W3C first defined the Resource Description Framefork (RDF), a simple yet very powerful triple-based representation for the semantic web. A triple typically consists of two entities linked by some relation, similar to the well-known predicate/argument structure. An example would be `directed_by(Avatar,James_Cameron)`. As RDFs became more popular, triple stores (referred as knowledge-bases) covering various domains have emerged, such as freebase.org. However, as the goal is to cover the whole web, the immediate bottleneck was the development of a global ontology that is supposed to cover all domains. While there are some efforts to manually build an *Ontology of Everything* like Cyc [19], the usual practice has been more suitable for Web 2.0, i.e., anyone can use defined ontologies to describe their own data and extend or reuse elements of another ontology [17]. A commonly used ontology is provided in schema.org, with consensus from academia and major search companies like Microsoft, Google,

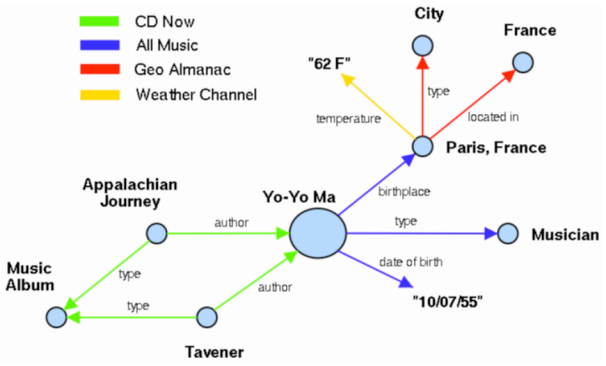

Figure 2: A segment of the semantic web pertaining Yo-Yo Ma (from [16])

and Yahoo. An example RDF segment pertaining the artist *Yo-Yo Ma* is shown in Figure 2. One can easily see that he was born in Paris in 1955, and is an author of the music albums, *Tavener* and *Appalachian Journey*.

These semantic ontologies are not only used by search engines, which try to semantically parse them, but also by the authors of these pages for better visibility. While the details of the semantic web literature is beyond the scope of this paper, it is clear that these kinds semantic ontologies are very close to the semantic ontologies used in goal-oriented natural dialog systems.

3.2. Mining Natural Language In-Domain Data

The standard method to build any statistical system is to gather as much in-domain data as possible. However, this does not scale very well for many cases, such as very specific tail domains (e.g., fly-fishing and specialized forms of head domains (e.g., ancient books). Furthermore, it is a time-consuming and very involved process before one can quickly test-drive a dialog system for the desired domain.

In this paper, we propose bringing together the semantic web experience and statistical natural language semantic parsing modeling. The idea is to leverage search engine resources for semantic parsing of the web pages feeding the target domain triple store. One can also easily replicate a toy version of this approach by using a target web site (such as imdb.com for movies).

The next step is to extract queries matching the already parsed web pages. This is similar to our previous work on exploiting query click logs for semi-supervised semantic parsing. However, now we have much more information about the web pages the queries are hitting, instead of just the domain. Then we semantically annotate the queries given the tags in the web page. For example, a query like "*2012 movie the hunger games by gary ross*" can be parsed using the information provided in Figure 1. Since there are vast numbers of queries, we employ a series of filters to get a representative set of data to train and test the understanding models. These include the elimination of queries which cannot be parsed and which have an untagged non-stopword matching one of the entities in the web page. These are mostly queries with typos, such as "*the hunger games by gary roos*". While one can employ spelling correction techniques for them, there is no need, due to the abundance of queries.

One issue with exploiting these queries is that, they are in query language, somewhat different than natural language. Vast majority of the queries consist of keywords or phrases as expected. Hence, we have explored various versions of this data, i) using all tagged queries (all), ii) using all tagged queries with a stopword which is not tagged to eliminate entity-only queries (like "*avatar cameron*") (NL-like), and iii) using queries which are also grammatical sentences (like "*who directed avatar*") (grammatical).

3.3. Building and Evaluation of Models

The mined queries can then be used to train the models. Furthermore, some of them (probably based on the date of the queries) can be set aside for testing purposes.

In our experimental framework, we have also explored the case where there is some unannotated in-domain data, as we assume a totally unsupervised framework without a semantic annotation guideline. For such a case we have explored a maximum-a-posteriori (MAP) adaptation technique, where the bootstrapped NLU model annotates the data, which can then be used to improve the model. This is similar to our previous work on unsupervised language model training for speech recognition [20].

One nice feature of the proposed approach is that the schema used to parse the web pages is now exactly same as the schema used to parse the natural language input. This also alleviates the problem of interpretation significantly, as there are no mismatches or inconsistencies, which happens frequently with task-specific semantic templates.

4. Experiments and Results

Here we present a proof-of-concept experimental study for the target domain of *movies*. The users present queries about various movies, such as *who is the director of avatar*, *show me some action movies with academy awards*, or *when is the next harry potter gonna be released*. We have only focused on 4 top named slots: movie, actor, director, and character names for the sake of simplicity.

We have set aside a control set of natural language data from this domain, consisting of about 2,700 sentences, with about 300 sentences reserved for testing. This includes about 3,750 slots (about 1,400 movie names). Compare this with the total amount of mined data, consisting of 287,216 queries (with 326,744 slots), after extensive filtering. We have found out that 48,364 of them (with 54,988 slots) are natural-language-like (NL-like), having a stopword in them which is not part of a slot, and 3,925 of them (with 4,046) are grammatical sentences.

Table 3 presents results using these 3 sets of queries for the query test set and the control test set. Seeing that the NL-like set results in better performance on the control set, compared to all or grammatical queries, we have used 20% of those queries for unsupervised test set. We believe this is because this set provides the best of both worlds, while having good coverage, it is more similar to sentences we would expect to get. On the control test set, the unsupervised approach achieved x% (64.26% vs. 57.73%) of the performance obtained using a supervised model, and actually outperforming the supervised model on some slots like the actor name.

More interesting results were obtained using the NL-like unsupervised test set. While the performances are higher, probably due to much bigger data sizes, we observe that using all queries resulted in slightly better performance for this set.

	Mined Test Set			Control Test Set		
	Movie Name F-Measure	Actor Name F-Measure	All Slots F-Measure	Movie Name F-Measure	Actor Name F-Measure	All Slots F-Measure
Supervised Control Train Set	24.49%	23.57%	23.86%	55.22%	81.25%	64.26%
All	83.94%	87.79%	86.43%	38.39%	89.13%	48.94%
NL-like	79.14%	86.88%	85.25%	47.94%	84.26%	57.73%
Grammatical	48.77%	64.77%	62.34%	11.24%	75.69%	33.63%

Table 2: Unsupervised slot filling performances under various conditions.

	Movie Name F-Measure	Actor Name F-Measure	All Slots F-Measure
Supervised Set	55.22%	81.25%	64.26%
Unsupervised Set	47.42%	83.59%	57.82%
NL-like	47.94%	84.26%	57.73%
NL-like + Unsupervised Set	50.21%	85.47%	60.03%

Table 3: Unsupervised slot filling adaptation performance on control test set, assuming in-domain sentences.

As a final set of experiments we have performed unsupervised MAP adaptation by automatically annotating the control train set using the model trained from NL-like queries. This is for simulating the scenario when there are some in-domain sentences but there are no semantic annotations or guidelines. This well-known technique has improved the performance of slot filling on the control test set significantly, covering 35% of the difference between the supervised and unsupervised model performances (57.73% to 60.03% F-Measure).

5. Conclusions

We have presented an initial study towards bringing together the semantic web experience and statistical natural language semantic parsing modeling. We mined search queries hitting the structured web pages to semantically annotate them and built statistical unsupervised slot filling models. We have presented results using a natural-language-like query set and a control test set for assessing the performance of the models. Furthermore, we have presented MAP adaptation for further improving these models in case when there are some in-domain unannotated data is available.

We believe, such an approach also ensures consistent semantic representation between the semantic parser and the backend knowledge-base. Note that, since the schema used will be limited to the semantic web, there may be some other (probably unnamed) slots where the model builder may wish to augment. These may include descriptive slots, such as "*movies that will keep me up all night*", or providing extra constraints, such as "*show me action movies in my instant queue sorted in alphabetical order*". Handling such queries in a scalable fashion is part of our future work.

6. Acknowledgments

We would like to thank Ashley Fidler for providing us the control data set and revising this paper.

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] 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.

[3] 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.

[4] 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.

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

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

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

[8] 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.

[9] 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.

[10] Y.-Y. Wang, R. Hoffmann, X. Li, and J. Szymanski, "Semi-supervised learning of semantic classes for query understanding: From the web and for the web," in *Proceedings of the CIKM*, Hong Kong, 2009.

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

[12] 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 the Interspeech*, Florence, Italy, 2011.

[13] 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 the Interspeech*, Florence, Italy, 2011.

[14] 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.

[15] C. J. F. J. B. Lowe, C. F. Baker, "A frame-semantic approach to semantic annotation," in *Proceedings of the ACL - SIGLEX Workshop*, Washington, D.C., April 1997.

[16] S. A. McIlraith, T. C. Sun, and H. Zeng, "Semantic web services," *IEEE Intelligent Systems*, pp. 46–53, 2001.

[17] N. Shadbolt, W. Hall, and T. Berners-Lee, "The semantic web revisited," *IEEE Intelligent Systems*, pp. 96–101, 2006.

[18] R. Guha, R. McCool, and E. Miller, "Semantic search," in *Proceedings of the WWW*, Budapest, Hungary, 2003.

[19] D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 32–38, 1995.

[20] D. Hakkani-Tür, G. Tur, M. Rahim, and G. Riccardi, "Unsupervised and active learning in automatic speech recognition for call classification," in *Proceedings of the ICASSP*, Montreal, Canada, May 2004.