EXPLOITING THE SEMANTIC WEB FOR UNSUPERVISED SPOKEN LANGUAGE UNDERSTANDING

Larry Heck *Dilek Hakkani-Tür*

Microsoft Research

larry.heck@ieee.org dilek@ieee.org

ABSTRACT

This paper proposes an unsupervised training approach for SLU systems that leverages the structured semantic knowledge graphs of the emerging Semantic Web. The approach creates natural language surface forms of entity-relation-entity portions of knowledge graphs using a combination of web search retrieval and syntax-based dependency parsing. The new forms are used to train an SLU system in an unsupervised manner. This paper tests the approach on the problem of intent detection, and shows that the unsupervised training procedure matches the performance of supervised training over operating points important for commercial applications.

***Index Terms*—** spoken language understanding, intent detection, structured knowledge-based search, semantic web

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.

Early work that led to most modern SLU systems include the DARPA project called Airline Travel Information System (ATIS) [2]. In the original ATIS project, the task consisted of spoken queries on flight-related information. An example utterance is *"I want to fly to Boston from New York next week"*. In this case, much of the SLU problem was reduced to the problem of extracting task specific arguments in a given (semantic) frame, such as Destination and Departure Date. And another aspect of SLU in ATIS was on the problem automatic intent classification. While the primary intent (or goal) was *Flight*, users also expressed other intents such as *Ground transportation* or *Airplane specifications*.

Both statistical and knowledge-based approaches were used in the ATIS project. Most of the knowledge-based approaches (e.g., [3]) originated from the field of artificial intelligence (AI). These methods leveraged deep semantics and relied heavily on rules and formal semantic interpretations. The interpretations mapped sentences into their *logical forms*: a context-independent representation of a sentence covering its predicates and arguments (intents, slots). For example, if the sentence is *"John loves Mary"*, the logical form would be *Love(John, Mary)*. An important special case of logical forms is an entity-relation representation. As defined in the 1997 W3C Resource Description Framework (RDF), entity-relation representations can play an equivalent role to logical forms in many semantic frame-based language understanding tasks.

While the concept of using semantic frames in ATIS were motivated by the case frames of AI, the intents and slots were very specific to the target domain. In addition, labeled/annotated example queries in sufficient volume for experimentation were made available for the task. Both of these factors enabled the SLU researchers to employ well established statistical classification methods for intent determination and slot filling(e.g., [4]). As a result, while the initial SLU systems in the ATIS project employed knowledge-based approaches, the statistical approaches began to dominate over time. For the *targeted language understanding* tasks typical of ATIS, these machine-learning methods proved to be more accurate and flexible than purely knowledge-based approaches.

But statistical methods have been shown to have their limitations as well. 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 hand-crafted domain-dependent schemas (ontologies). In addition, high accuracy of statistical SLU methods rely on *supervised* training patterns (i.e., the patterns 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 of statistical systems. 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 [5], intent [6], and slot filling [7] error rates.

In our most 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. In [8], we exploit the semantic structure of the web pages users visited when completing tasks.

In this paper, we extend this idea in several ways. Rather than relying on clicked web pages, we exploit the recent emergence of the large-scale semantic graphs in the web search community. Specifically, we more directly leverage the structure of the semantic graphs to automatically specify a semantic representation for SLU. The se-

978-1-4673-5126-3/12/$31.00 ©2012 IEEE

mantic representation guides the creation of entity-relation patterns that are used to mine natural language (NL) surface forms from the web. These NL surface forms are used to enrich the original semantic graph; the forms are attached to each knowledge concept and associated relations on the graph. Finally, these surface forms are used to automatically train SLU systems in an unsupervised manner that cover the knowledge represented in the semantic graph. In this paper, we demonstrate the utility of the approach for SLU on the problem of (unknown) intent detection.

Intent detection aims to find utterances that are in the domain of the dialog system, but are not covered by the current application. For example, while a spoken dialog system that aims to provide information to users about movies may know about movie directors, it may not know which awards are granted to which movies. So, detecting an utterance such as ”Did Avatar get any of the Oscars?” as such an utterance is the scope of intent detection. While the frequency of such utterances depends on the scope of the application as well as the user interface design, in human-initiative dialog systems, they are expected to appear frequently.

2. EXPLOITING THE SEMANTIC WEB

The methods developed in this paper rely on the extensive complementary literature on the semantic web [9, 10] and semantic search [11]. In the 1997 W3C Resource Description Framework (RDF), a simple yet powerful triple-based representation was used for the Semantic Web. A triple typically consists of two entities linked by some relation. As stated previously, this is an important special case of the well-known predicate/argument structure of logical forms used in the AI community. For example, *Directed By (Avatar,James Cameron)* is included in the knowledge graph to state that *“the director of the movie Avatar is James Cameron”*.

As RDFs became more popular, triple stores (knowledge-bases) covering various domains like *Freebase*[1] emerged. However, to cover the entire web, the immediate bottleneck was the development of a global ontology to represent all domains. Efforts to manually build an “Ontology of Everything” include Cyc [12]. A more recent and promising effort is being driven by companies with large-scale web search engines (Microsoft, Google) and adopted by many academic institutions. This effort is based on the initiative *schema.org*[2]. This initiative provides a collection of schemas (ontologies) that webmasters can use to semantically and uniformly markup their pages. The collection is growing everyday, with new contributions from many sources. Search engines like Bing, Google, and Yandex have adopted the ontology of schema.org and are leveraging it to support semantic search that will eventually grow to the scale of the web. An example RDF segment pertaining the artist Yo-Yo Ma is shown in Figure 1. 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 web pages. 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.

2.1. Enriching Semantic Graphs with NL Surface Forms

Given the structured knowledge-bases of the web include entities (e.g., movies, organizations, restaurants, etc.) and relations (e.g., director, founder, menu), our goals for SLU are twofold: (1) enrich these knowledge bases with natural language surface forms and (2) use these surface forms to train statistical models in SLU systems. Examples of surface forms are shown in Table 1.

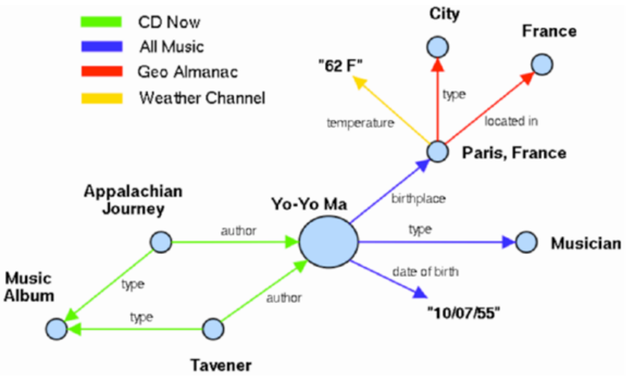
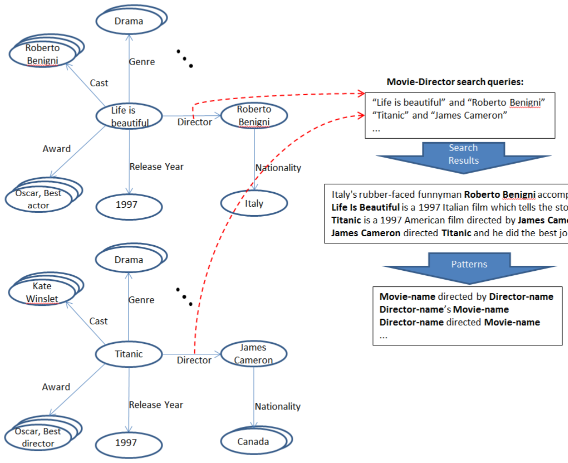


Fig. 1. A segment of a semantic web pertaining to Yo-Yo Ma [9]



Movie-Director search queries:

“Life is beautiful” and “Roberto Benigni”
“Titanic” and “James Cameron”
...

Italy's rubber-faced funnyman **Roberto Benigni** accomplishes ...
Life Is Beautiful is a 1997 Italian film which tells the story of a ...
Titanic is a 1997 American film directed by **James Cameron**...
James Cameron directed **Titanic** and he did the best job you...

Movie-name directed by **Director-name**
Director-name’s **Movie-name**
Director-name directed **Movie-name**
...

Fig. 2. Extracting surface forms from semantically structured KB.

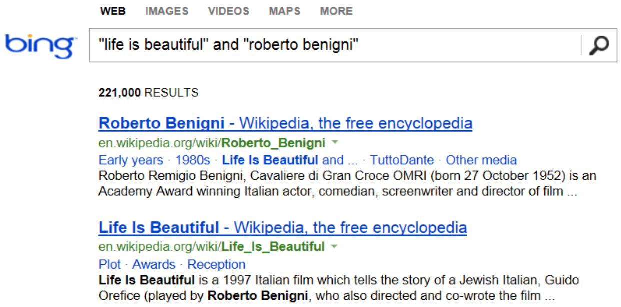
Entity	Relation	Entity	NL Surface Form
COMPANY	Founder	PERSON	COMPANY is founded by PERSON
			PERSON, founder of COMPANY
			Who is the founder of COMPANY
			Which company is PERSON a founder
MOVIE-NAME	Director	PERSON	MOVIE-NAME directed by PERSON
			PERSON’s MOVIE-NAME
			The critically acclaimed movie MOVIE-NAME directed by PERSON

Table 1. Example NL surface forms from structured KBs.

Figure 2 shows a semantically structured knowledge-base in graph form, and the process used in this study for extracting NL surface forms. Given an entity in the graph, the process starts by forming web search queries through a simple conjunction with related entities. For example, the entity *“Life is Beautiful”* (a movie) is related to the entity *“Roberto Benigni”* (a director), so the formed web search query is *“Life is Beautiful” and “Roberto Benigni”*. Figure 3 shows a portion of the search results page for the query *“Life is Beautiful” and “Roberto Benigni”*.

Once the web search queries are formed, the process retrieves the top-N most relevant documents from the web using a standard search engine (e.g., Bing, Google). The summarized captions or snippets of the retrieved documents on the search results page are

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



"life is beautiful" and "roberto benigni"

221,000 RESULTS

Roberto Benigni - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Roberto_Benigni**
Early years · 1980s · **Life Is Beautiful** and ... · TuttoDante · Other media
Roberto Remigio Benigni, Cavaliere di Gran Croce OMRI (born 27 October 1952) is an Academy Award winning Italian actor, comedian, screenwriter and director of film ...

Life Is Beautiful - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Life_Is_Beautiful**
Plot · Awards · Reception
Life Is Beautiful is a 1997 Italian film which tells the story of a Jewish Italian, Guido Orefice (played by **Roberto Benigni**, who also directed and co-wrote the film ...

Fig. 3. Example search results for the query *"Life is Beautiful" and "Roberto Benigni"*.

used as the source of NL surface forms. In this study we employ the Berkeley Parser [13], a state-of-the-art parser trained from a treebank following a latent variable approach by iteratively splitting non-terminals. We use the LTH Constituency-to-Dependency Conversion toolkit[3] to form dependency parses from the output parse trees. Once the sentences returned by Bing search are dependency parsed, the algorithm picks the smallest dependency sub-tree that includes the two entities of the branch on the semantic graph. For example, Figure 4 shows the dependency parse of the sentence *"James Cameron directed Titanic and he did the best job you could ask for"*, and the word sequence corresponding to the smallest sub-tree including the two entities *James Cameron* and *Titanic*. From the knowledge repository, we know that one of the entities is the Director-name, and the other is the Movie-name. The candidate pattern from this sentence is obtained by replacing these entities with their entity tokens.

After the patterns are extracted using their dependency parses, a score, $s(p)$, for each pattern, p, is computed using the following equation:

$$s(p) = P(rel_1|p) - P(rel_2|p)$$

where $P(rel_i|p)$ is the probability of the i^{th} most probable relation given the pattern. The goal is to extract the set of most distinguishing patterns for the specific type of entity relation. Hence, the patterns with highest scores are assigned to their most probable relations.

2.2. Unsupervised Learning from Structured Knowledge

Traditional training of SLU systems requires queries to be manually labeled (domain, intent, and slots). But this supervised training limits the breadth of the SLU semantic coverage. Leveraging the enriched semantic graphs described in the previous section, we can automatically infer SLU labels on training data. With the automatic labels, we can train SLU systems in an unsupervised manner. Given the breadth of available semantic graphs (e.g., Freebase), we can greatly expand the coverage of domains, intents, and slots of the SLU system. Each branch of the semantic graph provides additional coverage for the SLU system, and the training proceeds by crawling through the graph until the knowledge-base is completely traversed. Given the unsupervised, automated nature of our proposed method, when a new structured knowledge source becomes available, the system can learn the new knowledge, i.e., natural language patterns that are frequently used when realizing the relation of entity pairs, in a *push-button* manner. These patterns can then be used to generate or mine additional training data or as features for machine learning. In this work, we take the approach of exploiting sentences/search snippets that include common patterns as additional training data for enhancing the data for known intents, and creating data for unknown intents. This provides an important methodology to scale conversational understanding systems to the scale of the web.

3. INTENT DETECTION

In spoken language understanding, an area of active research is developing methodologies to treat users' utterances which cannot be handled by the application or back-end system. This problem arises because systems are typically designed to operate over limited and narrow domains, which is often required to achieve accurate speech recognition and understanding. Depending on the experience of the user and the effectiveness of the dialog design, users may provide utterances that cannot be handled by the application or back-end system.

As a result, language understanding systems usually process the input query with multiple passes, with the first pass used to detect whether or not the system can satisfy the user's intent, i.e., what the user wants to accomplish. This processing step is called *intent detection*. For example, in the restaurant domain with the user query *"I'd like to make a reservation at Xahn Restaurant"*, the system must determine whether or not it can satisfy the intent of making a restaurant reservation.

3.1. Methodology

The basic components of an intent detector are shown in Figure 5. The input query (spoken or text) is processed to extract features that convey intent information. The intent detector employs a method similar to a likelihood ratio test to distinguish between two hypotheses: the input query is a known intent (target) or not known (non-target). Features extracted from the input query are evaluated against a model representing the known intents, generating a score. Similarly, a score is computed against a model representing all other unknown (background) intents. For a likelihood ratio test, a final score,Λ, is computed as a ratio of likelihoods from each model (or difference in the log domain). This score is then compared to a threshold, θ: if greater, then make an "Accept" decision (the intent is known) and if less, then "Reject" the intent as unknown.

$$\Lambda(X) = \max_z \{S_z\} - S_b \tag{1}$$

where S_z is the score of the input query against the z-th intent model, and S_b is the score of the background model (BGM).

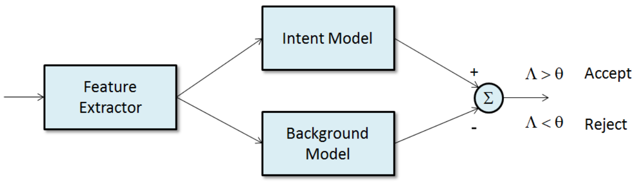


Fig. 5. Intent Detector

3.2. Intent Modeling

The intent model is composed of models for all of the known intents, as shown in Figure 6. In this paper, we use a discriminative Boosting toolkit called icsiboost [14] to train N one-versus-others classifiers for each of the N known intents. The toolkit implements a form of

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

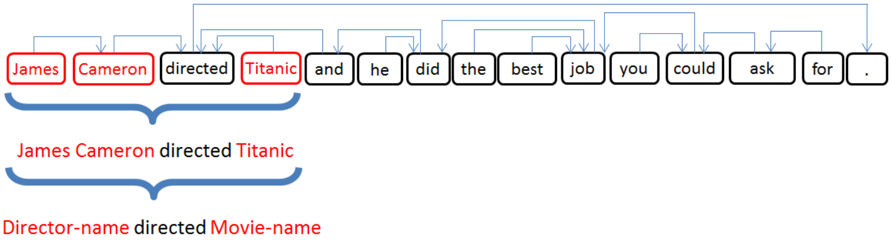

James
Cameron
directed
Titanic
and
he
did
the
best
job
you
could
ask
for
.
James Cameron directed Titanic
Director-name directed Movie-name

Fig. 4. Extraction of a pattern from a given sentence using dependency parsing. The two entities searched are shown in red fonts.

Adaptive Boosting [15] which uses one-level decision trees to implement a greedy search over weighted linear combinations of weak classifiers. The modeling of intents are completed over discrete word N-grams.

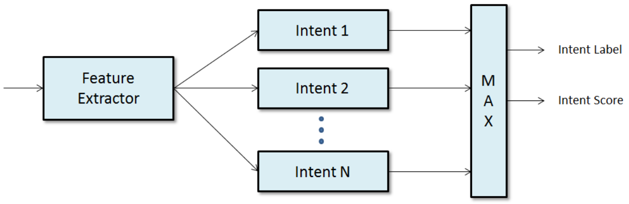


Fig. 6. Intent Model

At runtime, the input query is evaluated against the weak classifiers, each of which assign a score to the query for each intent, and the top scoring intents are assigned to the query.

3.3. Background Modeling

3.3.1. Out-of-Domain Detection

For cases where the utterance is out of the domain, such as *"where is the closest hotel"* to a movies application, the intent detection problem is referred to as *out-of-domain* (OOD). This problem has been studied in previous literature and several effective methods have been established (for example in [16]). The most common approach for OOD detection is to simply train a background model on queries from intent classes *other than* those from the target domain. It has been shown that the performance of the detector is relatively insensitive to the choice of the non-target domains as long as the distribution and quantity of domains and queries are relatively large. And given the readily available queries from non-target domains, high performing OOD detectors are relatively straightforward to develop.

3.3.2. In-Domain, Unknown Intent Detection

When the users' utterance is in-domain but still not known or covered by the capabilities of the application, the system needs to detect this condition and respond appropriately. We will call this lesser known and more difficult problem *in-domain unknown* (IDU) intent detection. An example of IDU detection is when the user queries an application built to support movie theater showtimes with *"what was the box office revenue for avatar"*. In this example, the utterance is in the domain of movies but the system was not programmed to understand *"box office revenue"*. When IDU queries are detected, often the best system response is to indicate to the user that system understood the domain of the request but was not able to complete the action (e.g., *"I'm sorry. I don't know the box office revenue for avatar"*).

Compared to OOD detection, IDU detection is typically more difficult. The challenges arise because many of the differentiating features (e.g., lexicon representing entities, relations, actions) between target and non-target utterances in OOD detection are no longer useful for IDU; many of these features now overlap between target and non-target utterances. For example, words such as "hotel" and entity-types such as HOTEL-NAME and HOTEL-AMENITIES may not be present in a movie theater show times application. But the word "movies" is likely to overlap in queries about a movie's box office revenue and movie theater show times. This ambiguity presents a significant and important challenge to commercial spoken language understanding systems.

3.4. Performance

Targeted language understanding systems typically complete a first pass on the query and decide whether to accept or reject it. The decision is made by comparing the system's confidence score to a threshold. If the score exceeds the threshold, accept the utterance for further processing, otherwise reject it. Rejections are often followed by a reprompt or some other dialog with the user to clarify their intent. The rejection may be due to various factors, including excessive noise in the speech signal or linguistic ambiguity or uncertainty.

Detection tasks trade off two types of errors: miss and false acceptance. A miss, or false rejection, occurs when the system incorrectly misses a "target" query; one that should have been understood by the system (e.g., in domain). False acceptances occur when the system incorrectly accepts a "non-target" query and decides the query is valid (in domain and known) when, in fact, the query should have been rejected as OOD or IDU.

To tradeoff the two error types, a single performance number is inadequate, and a performance curve over multiple operating points is used. One approach used for this purpose is ROC (Receiver Operating Characteristic) curves. The ROC curve plots the probabilities over the two error types on a linear scale. Another approach that has gained traction and been adopted in the speech processing community is the DET (Detection Error Tradeoff) curve [17]. The DET curve plots the probabilities of miss versus false accept on a normal deviate scale: the x- and y-axes are scaled non-linearly by the standard normal deviates between the score distributions for targets and non-targets, yielding tradeoff curves that are more linear than ROC curves.

For intent detection, the operating point on the DET curve is typically selected to minimize a cost measure, where the cost combines the separate costs (e.g., to the user experience) of the two error types. This can be written as

$$\mathrm{C} = \mathrm{C}_{miss} \cdot \mathrm{P}_{Tgt} \cdot \mathrm{P}_{miss|Tgt} \quad + \quad \mathrm{C}_{fa} \cdot \mathrm{P}_{NonTgt} \cdot \mathrm{P}_{fa|NonTgt} \quad (2)$$

where the total cost, C, is a probabilistic combination of the cost of a miss C_{miss} of a target intent, and the cost of a false accept C_{fa} of

Query	Intent
“create a list of the top ten banks by employees”	Find Company (General)
“what is the price of your common preferred and adjustable stock”	Find Stock Information
“show how much money was spent by Microsoft on advertising”	Find Finances
“what are analysts saying about investing in the Coca Cola company stock”	Find News
“can you tell me about the sales revenue from the last quarter”	Find Revenue
“which cell phone model had the largest number of complaints in 2011”	Find Products
“show the highest paid tech CEO and his salary versus company revenue”	Find Leadership
“chart Apple’s sales for last year”	Find Annual Sales
“show me any history and info on the treasurer of Dell”	Find People
“find me all the overseas offices for Apple and rank them by highest market cap then by liabilities”	Find Location

Table 2. Example queries and intents from the business domain.

a non-target intent. Here, the prior probabilities of a target and non-target are given as P_{Tgt} and P_{NonTgt}, respectively, and the curve of operating points that represents the quality of the intent detection is given by the likelihood measures $P_{miss|Tgt}$ and $P_{fa|NonTgt}$.

4. EXPERIMENTS

We considered the problem of intent detection in the business domain. Example queries for the 10 *known* intent classes of the business domain are shown in Table 2. We define 27 other intents to represent *unknown* classes. To study the effectiveness of our unsupervised training methods of this paper, we train separate intent classes for the 10 *known* target intent classes shown in Table 2, and then a train a single background model (BGM) in an unsupervised manner to represent the remaining unknown non-target classes.

The training data for the intent models consists of 4,032 queries distributed across the 10 known intent classes of Table 2, with the *Find Company (General)* class having the most tokens (35.8%) and *Find Location* having the least (3.1%). For each set of models, we use icsiboost to discriminatively train the intent detector. Specifically, all detectors use bigram features, and are trained with 1000 Boosting iterations with the default smoothing value of 0.5.

The test data for the IDU detection experiments consists of 899 queries covering both the 10 known intent classes as well as the other (unknown) 27 intents. The distribution of queries over the known intent classes in the test data was approximately the same as the training data.

Each query for both training and testing was processed with a named-entity recognizer (NER). The NER used a multi-pass, longest string match method against a large collection of entities from Microsoft Bing’s enriched version of freebase.org.

Referring to the intent detection cost measure in Equation 2, the prior probability of the known, target intent class P_{Tgt} and unknown intent P_{NonTgt} as well as the cost of errors C_{miss} and C_{fa} determine the operating point on the system’s performance DET curve. To be most illustrative, we will show the entire DET curve. But it should be noted that the portion of the DET curve that is likely more important for many applications is the upper left (lower $P_{fa|NonTgt}$). This is because a false accept typically leads to the system taking an erroneous action, whereas a miss simply generates a reprompt. To highlight this region of the DET curve, we will compute the $P_{miss|Tgt}$ for low $P_{fa|NonTgt}$ for each curve.

The first experiment compares the tasks of OOD and IDU detection. We seek to confirm the supposition that IDU detection represents an inherently more difficult task. Figure 7 shows the performance of the two tasks on the DET curve. The upper right DET curve with the highest error rates is the baseline IDU system trained over the 10 known intent classes with no background model. The OOD detector’s performance is the second curve from the upper right. For comparison, the same system was used for both the OOD and IDU detectors; the only difference was the testset. The IDU testset is described above. For the OOD system, a testset was constructed using 3,627 queries from the Movies domain (e.g., *“when is the second season of vampire diaries coming out”*), along with the same set of in-domain queries as the IDU test. The equal-error-rates (EER), or the operating point where the two detection errors are equal, and the $P_{miss|Tgt}$ at $P_{fa|NonTgt} = 10\%$ for the two tasks are shown in Table 3. As can be seen, the OOD performance is consistently better (lower detection error) across all of the operating points as compared to the IDU system. The EER drops from 36.5% of the IDU detector to 31.3%, and the P_{miss}@Pfa=10% drops by 29.8% (rel.) from 73.4% to 51.5%.

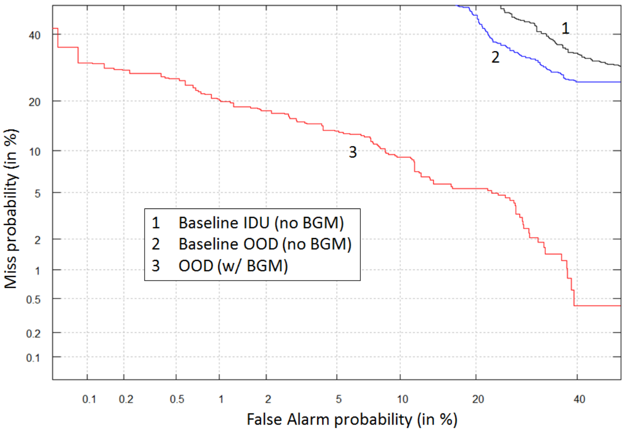


Fig. 7. DET curves for OOD and IDU intent detection

	EER	P_{miss}@Pfa=10%
Baseline IDU (no BGM)	36.5%	73.4%
Baseline OOD (no BGM)	31.3%	51.5%
OOD (w/ BGM)	9.5%	9.1%

Table 3. Comparison of OOD and IDU intent detection

For comparison to OOD systems reported in the literature, we include results on an OOD detector with a background model (BGM). The BGM was trained with 4,732 queries from the other domains (Hotels, Restaurants). Example OOD queries from this testset include *“four star affordable suites of america that have a separate reading area”* and *“find the phone number of bamboo garden bellevue”*. With the BGM, the EER drops to 9.5%, which is comparable to the best OOD reported [16].

Given the relative complexity of IDU detection and importance for commercial conversational understanding systems, and the ease of building an effective OOD detector, the primary focus of this pa-

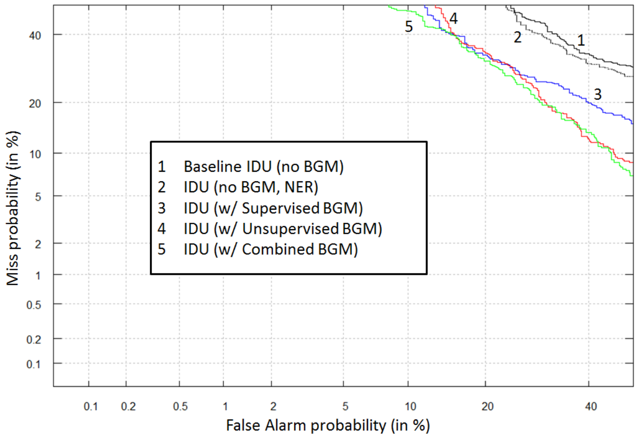


Fig. 8. DET curve of the proposed unsupervised SLU method

per is IDU detection. With the increased ambiguity of in-domain but unknown intent classes, we require higher precision training queries for the BGM. Figure 8 and Table 4 compare four systems: a Baseline IDU detector with no BGM (same baseline as the IDU vs OOD experiments above), a Baseline IDU detector trained with queries processed by the named-entity recognizer as described above, and two IDU detectors with BGMs trained with and without supervised intent class labels. For the unsupervised training, we used 23,561 NL surface forms from the enriched semantic graphs. These were produced with the procedure described in Section 2. The number of retrieved search results (and associated captions) used was $N = 10$. Example queries for the *Find Founder* intent class are shown in Table 1. For the supervised training of the BGM, we used hand-crafted queries, following the same procedure used to create the training queries.

	EER	P_{miss}@Pfa=10%
Baseline IDU (no BGM)	36.5%	73.4%
(no BGM, Named Entity Recognition)	35.1%	72.2%
IDU (w/ Supervised BGM)	26.4%	54.3%
IDU (w/ Unsupervised BGM)	**27.0%**	**53.5%**
IDU (w/ Combined BGM)	**25.2%**	**47.8%**

Table 4. Results of the proposed unsupervised SLU method

As can be seen, the performance of the unsupervised semantic graph-based method developed in this paper approximately matches the performance of the supervised training in the EER and upper left (low P_{fa}) regions of the DET curve. The unsupervised method's EER is 27.0% compared with 26.4% for supervised, and 53.5% at P_{miss}@Pfa=10% for unsupervised training, compared with 54.3% for supervised. The unsupervised EER is 26% better (rel.) than the baseline, and it is not significantly different than supervised (z-test). The supervised BGM does, however, perform significantly better in the high P_{fa} (low P_{miss}) region of the curve. When the supervised and unsupervised training data is combined, the resulting system improves, with an EER of 25.2% and the P_{miss}@Pfa=10% is 47.8%, which is 34.9% better (rel.) than the baseline and significantly better than either the supervised or unsupervised BGM.

5. CONCLUSIONS

This paper proposed an unsupervised training approach for SLU systems that exploits the structure of semantic knowledge graphs from the web. The approach enriched the semantic graph with NL surface forms created from entity-relation-entity portions of the knowledgebase. Realizations of intents were created using a syntax-based dependency parsing method. This paper tested the approach on the problem of intent detection, and showed that the unsupervised training procedure matched the performance of supervised training over operating points important for commercial applications. Future work will extend the approach to intent determination and slot filling.

Acknowledgements: The authors would like to acknowledge Ashley Fidler for the data collection and annotation efforts, and Gokhan Tur for many helpful discussions.

6. 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] 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] J. Dowding, J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran, "Gemini: A natural language system for spoken language understanding," in *Proceedings of the ARPA Workshop on Human Language Technology*, Princeton, NJ, March 1993.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

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

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

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

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

[15] 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.

[16] I. Lane, T. Kawahara, T. Matsui, and S. Nakamura, "Out-of-domain utterance detection using classification confidences of multiple topics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 150–161, 2007.

[17] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," in *Proceedings of the EUROSPEECH*, Rhodes, Greece, 1997.