

# USING A KNOWLEDGE GRAPH AND QUERY CLICK LOGS FOR UNSUPERVISED LEARNING OF RELATION DETECTION

Dilek Hakkani-Tür Larry Heck Gokhan Tur

Microsoft Research  
Mountain View, CA, USA

## ABSTRACT

In this paper, we introduce a novel statistical language understanding paradigm inspired by the emerging semantic web: Instead of building models for the target application, we propose relying on the semantic space already defined and populated in the knowledge graph for the target domain. As a first step towards this direction, we present unsupervised methods for training relation detection models exploiting the semantic knowledge graphs of the semantic web. The detected relations are used to mine natural language queries against a back-end knowledge base. For each relation, we leverage the complete set of entities that are connected to each other in the graph with the specific relation, and search these entity pairs on the web. We use the snippets that the search engine returns to create natural language examples that can be used as the training data for each relation. We further refine the annotations of these examples using the knowledge graph itself and iterate using a bootstrap approach. Furthermore, we exploit the URLs returned for these pairs by the search engine to mine additional examples from the search engine query click logs. In our experiments, we show that, we can achieve relation detection models that perform about 60% macro F-measure on the relations that are in the knowledge graph without any manual labeling, resulting in a comparable performance with supervised training.

**Index Terms**— semantic web, spoken language understanding, knowledge graph, search query click logs, multi-class classification

## 1. INTRODUCTION

Spoken dialog queries to a dialog system may be classified as *informational*, *transactional*, and *navigational* in a similar way to the taxonomy for web search [1]. Informational queries seek an answer to a question, such as “find the movies of a certain genre and director”, transactional queries aim to perform an operation, such as “play a movie”, or “reserve a table at a restaurant”, and navigational queries aim to navigate in the dialog, such as “go back to the previous results”. Answers to informational queries are likely to be included in knowledge repositories, such as the structured semantic knowledge graphs of the emerging semantic web, for example, *Freebase*<sup>1</sup>. Hence the ontology of user intents for informational queries can be formed based on the semantic web ontologies, such as the ontology of *Freebase* or *schema.org*<sup>2</sup>. 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 [2]. Navigational intents can usually be shared across ontologies of similar dialog system applications.

<sup>1</sup><http://www.freebase.com>

<sup>2</sup><http://www.schema.org>

Utterance: *find me recent action movies with brad pitt*

Intent	Find_Movie
Release_Date	<i>recent</i>
Genre	<i>action</i>
Actor	<i>brad pitt</i>

**Table 1.** An example utterance with semantic template.

As the ontologies of the semantic web can be used to bootstrap ontologies for dialog system applications, one can also use the populated knowledge in the graph to mine examples that include surface forms of entities and their relations in natural language. For example, for a pair of related entities, one can enhance the link of the relation in the knowledge graph with a set of natural language patterns that are commonly used to refer to that relation. Such patterns can be useful to train models for various language processing tasks, such as spoken language understanding (SLU). SLU in human/machine spoken dialog systems aims to automatically identify the intent of the user as expressed in natural language and extract associated arguments or slots [3] towards achieving a goal. The output of an SLU system is typically normalized and interpreted into a SQL-like structured query language or third party API. Historically, intent determination has emerged from the call classification systems (such as the AT&T How May I Help You [4] system) after the success of the early commercial interactive voice response (IVR) applications used in call centers. On the other hand, the slot filling task originated mostly from non-commercial projects such as the DARPA (Defense Advanced Research Program Agency) sponsored Airline Travel Information System (ATIS) [5] project.

Such semantic template filling based SLU systems with intent determination and slot filling tasks rely on a semantic space, usually dictated by the target application. When statistical methods are employed, in-domain training data is collected and semantically annotated for model building and evaluation. An example utterance with semantic template is shown in Table 1.

In our previous work, we showed the use of web search queries and search query click logs with the knowledge graph to bootstrap SLU slot filling models [6]. Furthermore, we used snippets returned from web search for pairs of related entities to bootstrap intent detection models in order to catch previously unseen in-domain intents [7].

In this work, instead of trying to align our SLU semantic space with the knowledge graph, we “only” rely on the semantic space dictated in the knowledge graph for informational user requests and aim to identify knowledge graph relations invoked in user’s utterances. The invoked relations can then be used to create requests in query languages (for example, in SPARQL<sup>3</sup> Query Language for RDF) to

<sup>3</sup><http://www.w3.org/TR/rdf-sparql-query/>

the knowledge graph, to create logical forms for natural language utterances [8], or to constrain slot filling and intent detection for SLU [3] according to the relations in the knowledge graph invoked by user's utterances. While this change is radical from SLU point of view (and contrary to the existing SLU literature), this actually is a simpler paradigm to implement, scales to the many knowledge graph domains and languages naturally, enables a wide variety of unsupervised SLU training approaches, and, by definition, guarantees consistency with the back-end information sources, hence results in more direct SLU interpretation.

In the next section, we will briefly give a very high overview of the emerging Semantic Web. Then in Section 3, we present the unsupervised relation detection approach. Section 4 presents the experiments in the framework of a conversational understanding system along with discussion of the results.

## 2. SEMANTIC WEB

In this study, we assume the knowledge graph defines the semantic space for the SLU model to be built. Such an approach relies heavily on the extensive complementary literature on the semantic web [9, 10] and semantic search [11]. In 1997, W3C first defined the Resource Description Framework (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 or knowledge graphs) covering various domains have emerged, such as *Freebase*. 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* [12], 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 [10]. A commonly used ontology is provided in *schema.org*, with consensus from academia and major search companies like Microsoft, Google, and Yahoo. An example RDF segment pertaining the movie *Life is Beautiful* is shown in Figure 2. One can easily see from the graph that this *drama* movie was directed by *Roberto Benigni* in 1997, which is also described by the following two triples:

Life is Beautiful	<i>Director</i>	Roberto Benigni
Life is Beautiful	<i>Release_Date</i>	1997

These semantic ontologies are not only used by search engines, which try to semantically parse them, but also by the authors of the in-domain web pages (such as *imdb.com*) for better visibility. While the details of the semantic web literature is beyond the scope of this paper, it is clear that these kinds of semantic ontologies are very close to the semantic ontologies used in goal-oriented natural dialog systems and there is a very tight connection between the predicate/argument relations and intents, as explained below.

## 3. 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 WWW, and further mine related queries from the search query click logs. We refine the annotations of the

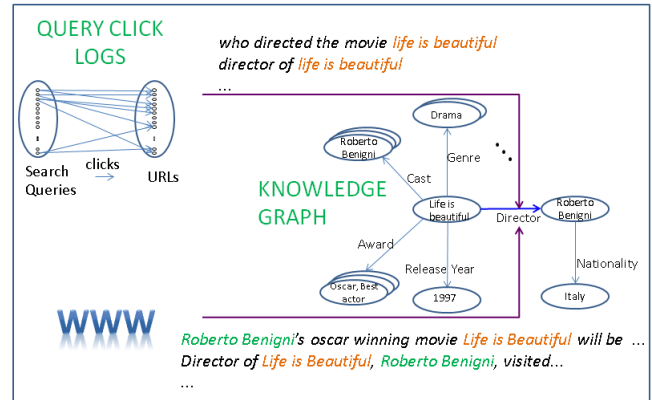


Fig. 1. Mining patterns from the world wide web and the search query click logs, guided by the knowledge graph.

mined examples via two methods that rely on other related entities on the knowledge graph and bootstrapping.

As in our earlier work [7], we extract all possible entity pairs in a given domain that are connected with a specific relation from the knowledge graph, and mine patterns used in natural language realization of that relation using web search<sup>4</sup>. We train relation detection models using the mined patterns. While the patterns are guaranteed to contain the searched entity pairs, they may also include other entities that are related. In order to refine the annotation of these patterns, we follow two approaches: one of them is based on the knowledge graph, and uses the other relations and entities in the graph to match the words in the mined pattern. The other one is a bootstrap method, and iteratively uses the trained model to find more relations in the mined examples. In addition to patterns mined from search results, we also enrich our training data by extracting queries that click on the web sites that contain the entity pairs. Figure 1 shows an overview of our approach, part of the knowledge graph (KG) and the mining of related patterns from the world wide web (WWW) and the search query click logs.

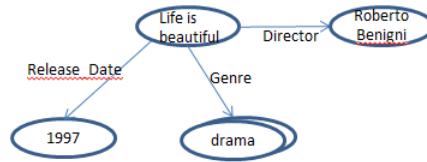
Our approach for mining examples guided by relations in the knowledge graph is similar to [13], but we directly detect relations invoked in user utterances, instead of parsing utterances with a combinatory categorial grammar [14]. Furthermore, we enhance our data with web search queries which are inquiring similar information as dialog system users.

### 3.1. Relation Detection

Relation detection aims to determine with relations in the part of knowledge graph related to the utterance domain has been invoked in the user utterances. For example, figure 2 shows two example utterances 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 both user requests contain the same “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.

<sup>4</sup>such as with <http://www.bing.com>

Sample from the relevant part of the knowledge graph:



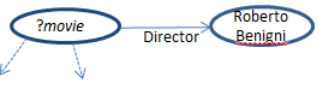

Sample user utterances:	<i>Show me movies by Roberto Benigni</i>	<i>Who directed Life is Beautiful?</i>
Corresponding relation on the knowledge graph		
Request in query language (simplified for demonstration):	SELECT ?movie { ?movie Director "Roberto Benigni". }	SELECT ?director { "Life is Beautiful" Director ?director. }
Request in logical form (simplified):	$\lambda y. \exists x. x = \text{"Roberto Benigni"} \wedge \text{Director}(x, y)$	$\lambda x. \exists y. y = \text{"Life is beautiful"} \wedge \text{Director}(x, y)$

Fig. 2. Knowledge graph, natural language queries and their interpretation according to the knowledge graph.

### 3.2. Mining Examples from the Web

Assume  $S_{ab}$  is the set of all snippets returned for the pair of entities  $a$  and  $b$  via web search<sup>5</sup>. We choose a subset of  $S_{ab}$ ,  $M_{ab}$ , that include snippets with both entities:

$$M_{ab} = \{s : s \in S_{ab} \wedge \text{includes}(s, a) \wedge \text{includes}(s, b)\}$$

where  $\text{includes}(x, y)$  is a binary function that has a value of 1 if string  $x$  contains  $y$  as a substring. One approach is using the complete strings of the snippets for each relation as training examples. However, the snippets can be lengthy and contain irrelevant information. Hence, as described in detail in [7], we parse the returned snippets with the Berkeley Parser [15], a state-of-the-art parser trained from a treebank following a latent variable approach by iteratively splitting non-terminals. Then we convert the output parse trees to dependency parses using the LTH Constituency-to-Dependency Conversion toolkit<sup>6</sup> [16]. We then pick the smallest dependency sub-tree that includes the two related entities that were searched, and use the word sequence in that sub-tree. This allows for clipping of irrelevant parts of the snippets while keeping the words that realize the relation in the snippet.

### 3.3. Refining Annotations of Example Snippets

While we require the snippets to include the two entities, and hence possibly invoke the relation between them, some snippets may invoke more than one relation. This may be because some entities are connected with more than one relation, and some entities are related to other entities as well. For example, the snippet *A Florida Enchantment is a silent film directed by and starring Sidney Drew* is mined as a training example for the "Director" relation, but it includes the movie "Cast" and "Genre" relations as well. This is because *A Florida Enchantment* is connected to *Sidney Drew* with more than one relation ("Director" and "Cast"), and the movie is linked to a genre, which is also invoked in this example. To refine the annotations of such examples that include more than one relation, we use

<sup>5</sup>In this work, we use Bing search engine and download the top 10 results for each entity pair

<sup>6</sup>[http://nlp.cs.lth.se/software/treebank\\_converter/](http://nlp.cs.lth.se/software/treebank_converter/)

two algorithms: The first one relies on the knowledge base, and retrieves all properties of the searched entities. For example, for the snippets mined for the "Director" relation in Figure 1, we form a list of entities and relations, that includes "Roberto Benigni" as "Cast", "Drama" as "Genre", "1997" as "Release\_Year", and "Oscar, Best actor" as "Award". These are then searched in the example and if a matching string is found, then the matching relation is added to the annotations of this example.

The second refinement approach is a bootstrap method, and relies on training a relation classifier with the mined data and their annotations. This classifier is then used to label the examples with more relations, in a way similar to Yarowsky's bootstrap algorithm [17]. In this step, only relations,  $r$ , with a high probability of appearance in the utterance,  $u$ , are included. Hence we optimize a threshold  $t$  for finding  $r$  with  $P(r|u)$  according to the classifier on a development data set.

### 3.4. Tying Search Logs with the Knowledge Graph

Large-scale search engines such as Bing or Google log more than 100M queries per day. Each query in the log has an associated set of URLs that are clicked after the users entered the query. This user click information could be used to find queries that are highly related to the contents of the clicked URLs, as well as queries that are related to each other. In our previous work, we showed the use of query click logs for mining examples and features for the SLU domain detection task [18, 19].

The search results for the related entity pairs also include URLs of the snippets that contain the two entities. Hence, we have access to the set of URLs,  $U_{ab}$ , that include the snippets in  $M_{ab}$ . Similar to the previous work, we search the Bing search engine query click logs for the queries whose users click on the URLs in  $U_{ab}$ . For each URL, we only use the most frequent 10 queries that include one of the entities of interest. We use these queries with the labels of the relation as training examples for relation detection.

## 4. EXPERIMENTS

We treat relation detection as a multi-class, multi-label (i.e. each utterance can invoke more than one relation) classification problem, and use icsiboost [20], a Boosting based classifier, with word unigrams, bigrams and trigrams as features.

#### 4.1. Data Sets

For training, we only use the patterns mined in an unsupervised way from web search and query logs. We use 7 entity pairs from the knowledge graph for experimenting with examples related to movie search, hence the relations include: director, star, release date, language, genre, country, and rating. We extract snippets related to each pair from web search results, and filter the snippets further to include only the ones that include the two entities. After cleaning, we end up with 178K patterns. The development data set is used to tune the thresholds for F-measure computation, and contains 1,200 utterances of 20 relations, some of which are not in the data mined from the knowledge graph, such as movie reviews or duration, and some utterances are not informational, but include transactional intents (such as “play trailer”). 66% of the development set utterances contain one of the 7 mined relations. The unseen test set also contain 1,200 examples, and 64% of these examples include one of the 7 relations.

#### 4.2. Evaluation Measures

In the experiments, we report two F-measures on the test set. *Targeted Macro-F* is the macro-averaged F-measure for the 7 relations for which we mined data. *Micro-F* is the relation detection F-measure on the test set examples, when all the 20 categories in the data set are considered.

#### 4.3. Results and Discussion

Table 2 shows the results from our experiments. As a baseline when no labeled training data is available, we assign all utterances only the majority relation (*Majority Class* experiment), which is the “Director” relation for the development set. The *Full Snippets* experiment uses  $n$ -grams of the complete snippet sequence (i.e. all of set  $M_{ab}$ ), and the *Patterns* experiments use the snippets clipped using dependency parses. (1 iter) refers to a single iteration of the bootstrap algorithm, and all models are improved when the bootstrapping method is applied to refine and extend the labels of training examples. Further iterations for bootstrapping are not reported in the table, as they did not result on any improvement on the development set after the first pass.

Patterns enriched with KG refers to the first approach of refining the annotations using the other entities in the knowledge graph. As seen from the results, all methods show improvement over using full snippets as the training examples. Refining the annotations with bootstrap method improves using single relation as the annotation of the example as well. Marking additional entities on the mined examples with the KG seems to not help, after examining examples, we noted that this may be due to the noise introduced, and integrating it with a classification method that uses context may be helpful to improve these results further.

Search queries by themselves, even after applying bootstrapping didn’t result in as good performance as the search snippets. This may be due to the different nature of the search queries, as often times search queries only include the entities or exclude function words that may be related to the relation.

The combination experiments refer to combining the estimation of the “Patterns from Snippets (1 iter)” model and the “Search Queries (1 iter)” model. Upper bound refers to using the correct relations that were picked by these two models (and hence is a cheating upper bound to show the room for improvement), and W-Voting refers to interpolating the decisions from the two models with weights optimized on the development set.

As seen in these results, by simply using full snippets from search results, we can obtain significantly better F-measure results

	Micro-F	Targeted Macro-F
Majority Class	20.3%	4.2%
Full Snippets	42.5%	55.1%
Patterns from Snippets	44.1%	58.0%
Patterns from Snippets (1 iter)	45.2%	59.6%
Patterns enriched with KG	44.5%	58.0%
Patterns enriched with KG (1 iter)	44.9%	58.9%
Search Queries	31.6%	40.6%
Search Queries (1 iter)	34.7%	43.2%
Combination (Upperbound)	50.2%	62.7%
Combination (W-Voting)	45.5%	59.9%
Supervised	47.6%	59.3%

**Table 2.** F-measure with each method before and after iteration, and their combination.

(both micro and macro) than using the majority class. When we further clip the examples and refine their annotations, we get an additional 9% relative boost on targeted macro F-measure and 7% relative boost on micro F-measure.

The last row of the table shows results with supervised training using 2,334 examples, manually labeled with one of the 7 relations. While the micro F-measure is 2.1% better with supervised training in comparison to the best unsupervised result, macro F-measures from both methods are about the same, showing comparable performances. This is intuitive since the unsupervised learning is pivoted around the entities and relations of the knowledge graph, providing guidance for mining data and hence modeling.

## 5. CONCLUSIONS

We presented an unsupervised relation detection approach that relies on mining the world wide web by searching for pairs of entities extracted from a knowledge graph that are connected by a specific type of relation. Performance has enhanced by clipping search snippets to extract patterns that connect the two entities on a dependency parse tree results and refining the annotations of relations according to other related entities on the knowledge graph. We show that our proposed unsupervised learning method performs comparable to a supervised training method which uses manually labeled training examples, requiring design, collection, and annotation of natural language data,

This approach is easily applicable by using freely available resources such as *Freebase* and *Bing*. However, the biggest advantage is that, such an approach naturally aligns semantic parsing and interpretation with the target knowledge graph, and the whole model can be built around pivot entities (such as the movie name, as presented in this study) and the corresponding relations. It scales to other domains and languages, pushing the burden from natural language semantic parsing to knowledge base population, which can be achieved using available structured knowledge sources such as IMDB or Wikipedia. Any in-domain data can further be exploited for better performance using supervised or unsupervised adaptation methods.

The future work includes extending this approach beyond relation detection to slot filling and semantic interpretation, closing the loop in a conversational understanding system, and demonstrating its use for multi-lingual natural language understanding.

**Acknowledgments:** We thank Ashley Fidler for her help with creating the development and test data sets, and Umut Ozterem for discussions.

## 6. REFERENCES

- [1] Andrei Broder, "A taxonomy of web search," in *ACM SIGIR Forum*, 2002, pp. 3–10.
- [2] Thomas Lin, Patrick Pantel, Michael Gamon, Anitha Kannan, and Ariel Fuxman, "Active objects: Actions for entity-centric search," in *Proceedings of World Wide Web Conference (WWW-12)*, Lyon, France, 2012, pp. 589–598.
- [3] Gokhan Tur and Renato De Mori, Eds., *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, John Wiley and Sons, New York, NY, 2011.
- [4] A. L. Gorin, G. Riccardi, and J. H. Wright, "How May I Help You?," *Speech Communication*, vol. 23, pp. 113–127, 1997.
- [5] 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.
- [6] 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*, Portland, Oregon, 2012.
- [7] Larry Heck and Dilek 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.
- [8] Luke Zettlemoyer and Michael 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.
- [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 the 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] Jayant Krishnamurthy and Tom M. Mitchell, "Weakly supervised training of semantic parsers," in *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, Jeju Island, Korea, 2012.
- [14] Mark Steedman, *Surface Structure and Interpretation*, The MIT Press, 1996.
- [15] S. Petrov and D. Klein, "Learning and inference for hierarchically split PCFGs," in *Proceedings of the AAAI*, 2007.
- [16] Richard Johansson and Pierre Nugues, "Extended constituent-to-dependency conversion for English," in *Proceedings of NODALIDA 2007*, Tartu, Estonia, May 25-26 2007, pp. 105–112.
- [17] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 1995, pp. 189–196.
- [18] Dilek Hakkani-Tür, Larry Heck, and Gokhan Tur, "Exploiting query click logs for utterance domain detection in spoken language understanding," in *ICASSP*. IEEE, 2011, pp. 5636–5639.
- [19] Dilek Hakkani-Tür, Gokhan Tur, Rukmini Iyer, and Larry Heck, "Translating natural language utterances to search queries for slu domain detection using query click logs," in *ICASSP*. IEEE, 2012.
- [20] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet, "Icsiboost," <http://code.google.com/p/icsiboost>, 2007.