# Rapidly Building Domain-Specific Entity-Centric Language Models Using Semantic Web Knowledge Sources

*Murat Akbacak, Dilek Hakkani-Tür, Gokhan Tur*

Microsoft, Sunnyvale, CA USA

{murat.akbacak,dilek,gokhan.tur}@ieee.org

## Abstract

For domain-specific speech recognition tasks, it is best if the statistical language model component is trained with text data that is content-wise and style-wise similar to the targeted domain for which the application is built. For state-of-the-art language modeling techniques that can be used in real-time within speech recognition engines during first-pass decoding (e.g., N-gram models), the above constraints have to be fulfilled in the training data. However collecting such data, even through crowd sourcing, is expensive and time consuming, and can still be not representative of how a much larger user population would interact with the recognition system. In this paper, we address this problem by employing several semantic web sources that already contain the domain-specific knowledge, such as query click logs and knowledge graphs. We build statistical language models that meet the requirements listed above for domain-specific recognition tasks where natural language is used and the user queries are about name entities in a specific domain. As a case study, in the movies domain where users' voice queries are movie related, compared to a generic web language model, a language model trained with the above resources not only yields significant perplexity and word-error-rate improvements, but also presents an approach where such language models can be rapidly developed for other domains.

**Index Terms**: speech recognition, language modeling, knowledge graphs, query click graphs, name entities, semantic web

## 1. Introduction

With advances in automatic speech recognition (ASR), spoken language understanding (SLU), and machine learning technologies (and rapid proliferation of mobile devices, especially smart phones), server-based and embedded speech and multi-modal applications have emerged. These range from simpler applications where speech recognition is followed by a known task such as voice search or messaging, to more complex systems such as conversational understanding (CU) systems as used in personal assistants.

In CU systems, at each turn, a user's speech is recognized, and then semantically parsed into a task-specific semantic representation of the user's intention [1]. For training, a domain-specific ASR system, the usual practice is to collect "enough" in-domain data to represent the use cases in terms of both content and style. This data is then used for building domain-specific language models for better speech recognition. However, collecting such data is time-consuming and expensive, and collected/transcribed data is often not representative of how a much larger user population would interact with the recognition system. Hence, the training data for language models need to be replaced/enhanced with real data as soon as it is available.

Most CU systems depend on the application and environment (such as mobile vs. TV) for which they have been designed. The back-end functionality of task-specific databases and knowledge bases typically define the scope of the target domain. Input queries to a CU system typically seek an answer to a question, such as *find the movies of a certain genre and director*, perform an operation, such as *play a movie*, or *reserve a table at a restaurant*, or aim to navigate in the dialog, such as *go back to the previous results*. The first two types of queries (which are similar to informational and transactional queries of web search), mainly include domain entities, their relations with other entities or their attributes. These relationships are likely to be included in back-end knowledge repositories, for example, the structured semantic knowledge graphs of the emerging semantic web, such as Freebase [2].

Inspired by earlier work on using knowledge graphs for SLU, in this paper, we propose to exploit the domain-specific semantic web knowledge sources to rapidly bootstrap language models for ASR. The two main sources include web search query click logs and semantic knowledge graphs. Query click logs are often represented as bipartite graphs that connect search queries with clicked uniform resource locators (URLs) with frequencies of joint occurrence. The knowledge graphs are sets of triples indicating a relation between two entities (e.g., *Avatar - directed_by - James Cameron*), compiled into a graph structure. Such knowledge-bases are more and more popular in the semantic search and parsing communities as described in detail below. However, such information is in graph format and not in natural language, and hence its use is limited for language modeling purposes.

In this study, we focus on movies domain where there is vast amounts of semantic web knowledge sources. More specifically, we employ web knowledge resources such as query click logs and knowledge bases to build language models that capture the content and style for the domain, as well as popularity information in a specific domain where users' queries are formed around name entities. We show that it is possible to build in-domain language models which outperform a generic language model using this approach. Furthermore, one can extend this work to exploit other semantic web sources such as corresponding web documents or snippets.

In Section 2 we present the related work on rapid language model development and conversational understanding. Section 3 provides an overview of these web knowledge sources, and how they have been used previously for understanding tasks. This is followed, in Section 4, with a discussion of how such sources can be utilized for building language models for a speech recognition task. The experimental setup and results are presented in Section 5 for the movies domain. Finally we conclude with a discussion and future work.

## 2. Related Work

Several previous studies looked at the rapid construction of language models [3, 4, 5]. In the context of rapid language modeling from web resources, Bulyko *et al.* proposed a methodology for collecting text data from the Web to match the style and topic of target application, and experimented with mixture models for meeting and lecture speech recognition [6].

Two previous studies targeted spoken dialog interactions [7, 8]. Akbacak *et al.* mined seed set of domain-representative actions and concepts, as well as action-concept pairs from in-domain knowledge sources that are not stylistically suitable to train language models directly (e.g., frequently asked questions, product manuals, website content, etc.) [7]. Initial seed list of actions and concepts are used to mine more actions, concepts, as well as action-concept pairs from web documents using syntactic and shallow-semantic parsers. Hakkani-Tür *et al.* uses semantic parsing to capture semantic relationships in target domain, as well as past domains to find domain independent conversational sequences and merge the two [8]. These two studies are the most relevant ones to our work since they also do not rely on any in-domain collected text data, and they aim to capture and employ semantic relationships for language modeling or understanding tasks. Here, we employ existing semantic web sources to capture domain semantics, and focus more on mining relevant sentences from web resources.

One of the first research studies to build conversational understanding systems from web documents, AT&T's WebTalk, relied on mining information from structured information in the form of tables and frequently asked questions (FAQs) [9]. In that work, user utterances were mapped to FAQs and system responses were formed accordingly. Other studies mainly focus on using web search query logs to bootstrap and improve semantic parsing of user utterances. For selecting domain-related URLs, Hakkani-Tür *et al.* relied on manually defining a small set of domain-related base URLs (such as imdb.com and rottentomatoes.com for the movies domain) [10, 11]; Hillard *et al.* used domain-specific entities in the semantic graph [12], and Wang *et al.* used both [13]. More recently, on a larger scale, knowledge graphs that are mined from the web or constructed manually were used to bootstrap and improve conversational understanding domain and intent detection and slot filling tasks. For example, Tur *et al.* employed queries clicked to entity indicator web pages (e.g., *James Cameron* web page at imdb.com) [14, 15]. Heck and Hakkani-Tür used Wikipedia pages of entities to automatically annotate them with information in the knowledge graph [16]. For speech understanding, another thread of work mainly focuses on searching entities and entity pairs on the web to train speech understanding grammars [17].

## 3. Overview of Web Knowledge Sources

In this section we provide an overview of the web knowledge sources, mainly query click logs and knowledge graphs. One can envision other sources such as corresponding documents or snippets, using the proposed approaches here without loss of generality.

### 3.1. Knowledge Graph

The Semantic Web is a collaborative movement aiming at converting unstructured and semi-structured documents into a structured semantic network [18, 19, 20]. 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 is `directed_by(Avatar,James_Cameron)`. As RDFs became more popular, triple stores (referred to as knowledge-bases) covering many 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 covers all domains. While there are some efforts to manually build an *Ontology of Everything* like Cyc [21], 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 [19]. A commonly used ontology is provided in schema.org, with consensus from academia and major search companies like Microsoft, Google, and Yahoo. A segment of the semantic knowledge graph about the movies domain is shown in Figure 1. In this figure, each node corresponds to an entity such as a movie (e.g., "Titanic") or a person (e.g., "James Cameron"). Each arc and its label denotes a relation (e.g., "directed_by") between the connected nodes.

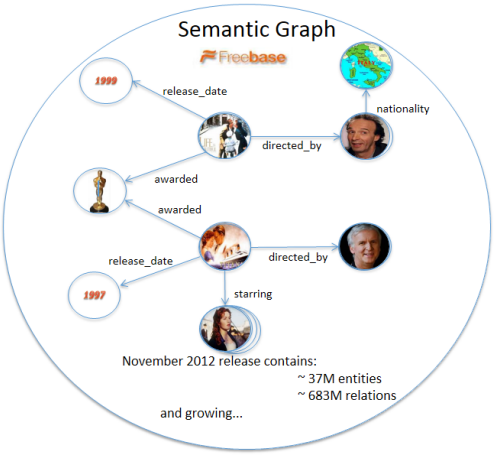


Figure 1: A segment of the semantic knowledge graph from the movies domain, showing the entities and their relations.

### 3.2. Query Click Graph

Large-scale search engines such as Bing or Google log more than 100M queries per day. Each query in these logs has an associated set of URLs that are clicked after the users entered the query. Such search query click logs are usually represented as a bipartite graph where each query belongs to the set of queries $Q$ and each URL belonging to the set of URLs $U$ is represented as a node, as shown in Figure 2. Directed arcs connect query $q_i \in Q$ and URL $u_j \in U$, if a user who types $q_i$ clicks on $u_j$. 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 via random walk algorithms [22]. Transition probabilities between these two sets of nodes can be computed by normalizing the frequencies of the click events, where $C(q_i, u_j)$ denotes the number of times $u_j$ was clicked on after query $q_i$ was issued. As an example,

$$P(u_j|q_i) = C(q_i, u_j) / \sum_{k \in U} C(q_i, u_k)$$

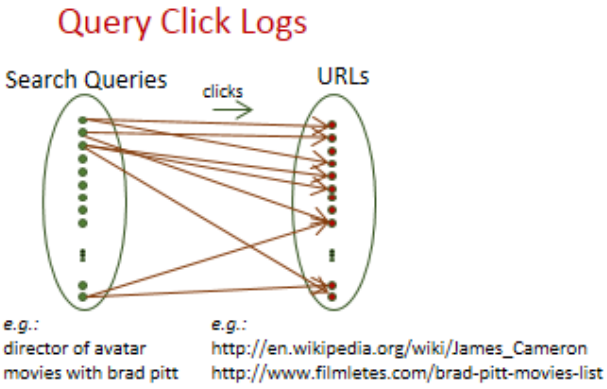


Figure 2: Search query click logs can be represented as a bipartite graph with weighted arcs from queries to URLs.

## 4. Proposed approach

For domain-specific conversational understanding tasks, it is best if the statistical language model for speech recognition component is trained with text data that is content-wise and style-wise (e.g., natural language word sequences not only containing entities but also carrier phrases around the entities) similar to the targeted domain for which the application is built. For state-of-the-art language modeling techniques that can be used in real-time within speech recognition engines during first-pass decoding (e.g., N-gram models), the above constraints have to be fulfilled in data that is used to train these models. The standard method to build any statistical system is to gather as much in-domain data as possible to capture required characteristics, and this has been the common practice in language modeling. However, this does not scale very well for many cases, such as very specific tail domains (e.g., fly-fishing) or specialized forms of head domains (e.g., ancient books). Furthermore, it is a time-consuming and very involved process before one can test-drive a dialog system for the desired domain. Even when a large amount of data is collected through crowdsourcing, the collected data can be artificial and might lack diversity in style and content since it is not trivial to monitor all inputs from crowd-sourcing subjects and introduce constraints to increase diversity. In other words, collected data can still be not representative of how a much larger user population would interact with the recognition system.

Here, we propose to capture domain relevant text data from query click graphs and knowledge graphs. We start with an entity list from a knowledge graph. More specifically, the target domain is initially modeled via only an entity list. This can be considered as a seed list to model the target domain. For example, in this paper we focus on movies domain as a test domain and for this domain entity list consists of actor/actress names, movies titles, etc. Next step is to mine web search query logs to find queries containing these entities. Here, an exact match is applied between the entity list and query logs so as not to expand too quickly on the query set size. Using the query click graph, we identify set of URLs that are clicked on when mined queries are issued by web search engine users. Our goal is to find domain-representative URLs so that later we can walk back (from URLs to queries) on the query click graph to expand on the set of queries we can use for language model training. At this stage, we start using both entity lists and URLs to model target domain. To find domain representative URLs, one can compute the probability of a click on a particular website ($url_i$) given an entity list. This can be done by aggregating the counts of clicks received by a particular website for queries that are coming from the entity list ($DomainSeedSet$ or $DSS$ in below equation), and then dividing this by total number of clicks received for all websites in the context of seed entity list:

$$p(url_i|DomainSeedSet) = \frac{clicks(url_i|DSS)}{\sum_j clicks(url_j|DSS)}$$

where clicks is the sum of all clicks that a particular website received over all queries in the seed query list. An alternative and more effective approach is to identify set of domain representative URLs in a more discriminative way. This can be achieved by introducing a large set of random queries ($RandomQuerySet$ or $RQS$) and calculating domain representativeness score as a log-likelihood ratio between $p(url_i|DomainSeedSet)$ and $p(url_i|RandomQuerySet)$ where

$$p(url_i|RandomQuerySet) = \frac{clicks(url_i|RQS)}{\sum_j clicks(url_j|RQS)}$$

and $RandomQuerySet$ is used to create a background model. Instead of using $p(url_i|DomainSeedSet)$ as a domain representativeness score, we calculate

$$r_{url_i} = \frac{logprob(p(url_i|DomainQuerySet))}{logprob(p(url_i|RandomQuerySet))}$$

for every $url_i$ to determine how well a specific URL represents the target domain.

When the initial entity list is large, duplicate memberships of the same entity in different entity lists used for different domains or categories in the knowledge graph can occur. Similar to assigning weights to each URL, we can also assign a weight for each entity in the original entity list and this weight can represent domain representativeness of an entity. Hillard *et al.* proposed a method based on the cross-entropy difference between cross-entropy of an entity's URL distribution against entity list's URL distribution and the cross-entropy of same entity's URL distribution against random query list's URL distribution [12]. The resulting score should be high for entities that have unambiguous membership in the list and therefore is a good representative of the target domain, and low for entities that have ambiguous or incorrect membership. In the case of the movie title list from Freebase, "The Dark Knight" is a phrase that uniquely references a movie, but the list also contains the title "Hotel" (a small movie from 2003) that has meaning in many other contexts. In our proposed approach, we use these entity weights to prune our original entity list to keep highly domain representative entities and recalculate

$$r_{url_i} = \frac{logprob(p(url_i|PrunedDomainSeedSet))}{logprob(p(url_i|RandomQuerySet))}$$

to obtain more reliable domain representativeness scores for each URL. Although we observe improvements after manually checking domain representativeness scores for a large set of URLs, this does not have significant impact on language modeling experiments. After obtaining a sorted list of URLs using their domain representativeness scores, we apply a threshold to keep the top $N$ URLs. The last step in the proposed approach is to walk backwards on the query click graph to the query side of the graph using the domain representative URL list and mine domain matching queries to train a domain-specific language model.

| Language Model | Data Size | Perplexity | OOV | WER |
|---|---|---|---|---|
| (a) All QCL queries | >300M | 194 | 0.35% | 37.3% |
| (b) Queries hitting KG URLs | 6M | 150 | 2.12% | 37.9% |
| (c) QCL domain-matching subset (top-15 URLs) | 20M | 127 | 0.23% | 33.6% |

Table 1: Perplexity and Word Error Rate (WER) based evaluation of different language models trained from (a) all Query Click Logs (QCL) vs. (b) queries from QCL hitting Knowledge Graph (KG) URLs that represent domain entities vs. (c) subset of queries from QCL selected using domain representativeness score, $r_{url_i}$, of URLs via graph walking (a hard threshold is applied on $r_{url_i}$ resulting in URL set sizes of $N = 15$)

# 5. Experiments

## 5.1. Data Sets

Experiments are performed using a conversational understanding system for the entertainment domain, with real users. In this domain, users issue voice 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*". For all experiments, we use the Freebase knowledge graph, which is publicly available [2]. The test set includes 1.4K such utterances, where the average utterance length is 4.3 words. On this test set, we evaluate a baseline approach where a generic language model trained from all queries is used, and we compare the performance of two proposed approaches:

1. Domain URLs from a knowledge graph are used to mine queries from Query Click Logs and then a language model is trained from these queries,
2. The proposed approach presented in Section 4 is used to identify set of URLs to model the target domain by calculating domain representativeness scores for each URL.

We use both perplexity, out-of-vocabulary (OOV) rate, and word error rate (WER) metrics. We should point out that due to differences in vocabulary sizes in some experiments perplexity numbers are sometimes not directly comparable, making WER the main metric for comparison purposes.

Queries are mined from Bing search engine query click logs from over a 2 year time period. The large random set of generic queries includes >300M queries, and the smaller set where we use the proposed approach to identify domain-representative URLs and mine corresponding queries, which are subset of all queries, includes 20M queries. In addition to these two systems, we also use set of URLs from a knowledge source where domain information is available, as presented in Section 4. For movies domain, using the URLs from a knowledge graph, we mined 6M queries. Using these three query sets, a separate language model is trained for each set, and the resulting language models are evaluated on the in-domain test set via recognition experiments. For language model training, we use trigram language models trained using the Knesser-Ney smoothing technique.

## 5.2. Results

Table 1 presents results for baseline and proposed approaches. We performed perplexity and Word Error Rate (WER) based evaluation of different language models trained from three query sets as shown in Table 1 at the top of this page.

We applied a hard threshold on $r_{url_i}$ to decide on a URL set of 15 URLs. As you can see from the results, a generic model which is considered as a baseline system in this paper and is trained from all queries, it yields WER of 37.3% on the test set. When we use queries hitting the knowledge graph URLs that represent the domain, we obtain slightly higher WER, and gains on perplexity for in-domain N-grams are suppressed by the increase in OOV rate. This results in a higher WER compared to generic language model. This might be due to the fact that URLs representing domain-specific entities in the knowledge graph are very specific URLs and this results in a low-recall query mining from QCL. Another factor is that queries hitting these KG URLs are not rich in terms of carrier phrases, bringing this LM's recognition performance close to a generic language model's performance. When we use our proposed approach of mining domain-representative URLs and corresponding queries to train a domain-specific language model, we obtain close to 10% relative gains in WER compared to generic language model. In our experiments, language model trained from queries hitting knowledge graph URLs did not result in WER improvements. Yet, further improvements can be observed when a tune set is available in the target domain to interpolate this language model with the language model trained with domain-representative queries.

As future work, after collecting a tuning set, we plan to perform language model interpolation experiments using the language models presented in this paper. In addition, we plan to incorporate the domain-representativeness scores of URLs into language model interpolation weights and interpolate URL-specific language models using the normalized domain-representative URL-specific language models. Similarly, on the query side of the query click graph, similar weighting can be applied to weight N-gram count statistics, and here domain-representative scores of each query from previous section will be used. We also plan to evaluate the proposed approach on other domain-specific scenarios where entity-lists might present different challenges. Finally, we plan to introduce other knowledge sources such as web search snippets and web document content.

# 6. Conclusion and Future Work

We have presented an approach for rapidly building domain-specific language models for speech recognition in conversational understanding systems. Our approach is based on semantic web knowledge resources, mainly focusing on in-domain semantic graph entities and corresponding web search query click logs, to automatically mine in-domain representative data. We show that our approach outperforms a generic web-based language model built from web search queries with close to 10% relative WER gains in the movies domain. The proposed approach can be extended to other semantic web knowledge sources such as corresponding web documents (e.g., Wikipedia entries of the entities), without loss of generality.

# 7. Acknowledgments

We would like to thank Umut Ozertem for providing us the knowledge graph data, and thank Benoit Dumoulin and Larry Heck for many helpful discussions.

## 8. References


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

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of ACM SIGMOD*, 2008.

[3] L. Galescu, E. Ringger, and J. Allen, "Rapid language model development for new task domains," in *Proceedings of the first international conference on language resources and evaluation (LREC)*, 1998, pp. 807–812.

[4] A. I. Rudnicky, "Language modeling with limited domain data," 1995.

[5] P.-C. Chang and L.-S. Lee, "Improved language model adaptation using existing and derived external resources," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2003, pp. 531–536.

[6] I. Bulyko, M. Ostendorf, M. Siu, T. Ng, A. Stolcke, and Ö. Çetin, "Web resources for language modeling in conversational speech recognition," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 5, no. 1, 2007.

[7] M. Akbacak, Y. Gao, L. Gu, and H.-K. J. Kuo, "Rapid transition to new spoken dialogue domains: Language model training using knowledge from previous domain applications and web text resources," in *Ninth European Conference on Speech Communication and Technology (EUROSPEECH)*, 2005.

[8] D. Hakkani-Tur and M. Rahim, "Bootstrapping language models for spoken dialog systems from the world wide web," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[9] J. Feng, S. Bangalore, and M. Rahim, "Webtalk: Mining websites for automatically building dialog systems," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2003, pp. 168–173.

[10] D. Hakkani-Tür, L. Heck, and G. Tur, "Exploiting query click logs for utterance domain detection in spoken language understanding," in *ICASSP*. IEEE, 2011, pp. 5636–5639.

[11] D. Z. Hakkani-Tür, G. Tur, L. P. Heck, and E. Shriberg, "Bootstrapping domain detection using query click logs for new domains." in *INTERSPEECH*, 2011, pp. 709–712.

[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 Interspeech*, 2011, pp. 705–708.

[13] L. Wang, L. Heck, and D. Hakkani-Tur, "Leveraging semantic web search and browse sessions for multi-turn spoken dialog systems," 2014.

[14] 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.

[15] G. Tur, M. Jeong, Y. Wang, D. Hakkani-Tur, and L. Heck, "Exploiting the semantic web for unsupervised natural language semantic parsing," in *Proceedings of Interspeech*. International Speech Communication Association, 2012.

[16] L. Heck and D. Hakkani-Tür, "Exploiting the semantic web for unsupervised spoken language understanding," in *In Prooceedings of the IEEE SLT Workshop*, Miami, FL, December 2012.

[17] I. Klasinas, A. Potamianos, E. Iosif, S. Georgiladakis, and G. Mameli, "Web data harvesting for speech understanding grammar induction," 2013.

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

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

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

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

[22] N. Craswell and M. Szummer, "Random walks on the click graph," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 239–246.