Probabilistic Enrichment of Knowledge Graph Entities for Relation Detection in Conversational Understanding

Dilek Hakkani-Tür, Asli Celikyilmaz, Larry Heck, Gokhan Tur, and Geoff Zweig

Microsoft

{dilek,asli,larry.heck,gokhan.tur}@ieee.org, gzweig@microsoft.com

Abstract

Knowledge encoded in semantic graphs such as Freebase has been shown to benefit semantic parsing and interpretation of natural language user utterances. In this paper, we propose new methods to assign weights to semantic graphs that reflect common usage types of the entities and their relations. Such statistical information can improve the disambiguation of entities in natural language utterances. Weights for entity types can be derived from the populated knowledge in the semantic graph, based on the frequency of occurrence of each type. They can also be learned from the usage frequencies in real world natural language text, such as related Wikipedia documents or user queries posed to a search engine. We compare the proposed methods with the unweighted version of the semantic knowledge graph for the relation detection task and show that all weighting methods result in better performance in comparison to using the unweighted version.

Index Terms:Semantic knowledge graphs, entity types, spoken language understanding, relation detection, spoken dialog systems.

1. Introduction

Semantic knowledge graphs, such as Freebase[1], encode factual world knowledge in triples of a pair of entities and their relation, for example, the triple

`<"Avatar",directed_by,"James Cameron">`

encodes that one of the directors of the movie *Avatar* is *James Cameron*. Such knowledge encoded in the semantic graphs in the form of entities and their relations can be useful for spoken language understanding (SLU) when detecting them in natural language utterances and determining their types and relation. For example, lists of entities of a specific type (such as gazetteers of movie names and actor names) have been commonly used for interpretation of natural language user queries in spoken dialog systems [2, 3, 4]. Entity lists/gazetteers can be formed from triples in semantic graphs, mined from the web or provided by third parties. When used as is, such lists usually introduce noise to spoken language understanding, as for example, commonly used terms could also be entities (for example, *Up*, and *Holiday Inn* are movie names). Previous work by [3] used an approach based on search queries and clicked URLs to assign a weight to each entity term to estimate if a term is more commonly used in natural language as an entity or not. They have shown the benefit of these weights for the slot filling and domain detection tasks. However, many entities are ambiguous as they are involved in many types of relations in the encoded knowledge. For example, in the domain of movies, many actors also direct or produce movies; or in the domain of organizations, the founder of a company is frequently the CEO of the same company or is on its board of directors.

When user requests are mapped to queries in a query language such as SQL or SPARQL, in addition to knowing that a sequence is an entity, it is also necessary to specify its type or relations it invokes. For example, for an utterance such as "I want to see something with Brad Pitt", the common understanding would be movies where *Brad Pitt* acted in. The query to the back-end knowledge source for this utterance would require specifying him as the actor, even though he may be listed in several types of roles, such as an actor or producer of movies.

In this paper, we tackle the problem of entity type ambiguity and investigate three methods for introducing weights to semantic graph entity types, with the end goal of improving interpretation of natural language user queries to a conversational interaction system. The first method requires a fully populated semantic knowledge graph (i.e. all entities and their relations with others are specified) and hence provides an upper bound. It estimates the posterior probability of each entity type based on the entries in the semantic graph. The second and third methods require a large text corpus, and are both based on the assumption that, entity types that are used more frequently in natural language for a specific entity should be weighted higher during SLU. Hence, these methods rely on instances of each entity in natural language text, such as Wikipedia documents and web search queries, and do not require a complete knowledge graph where all relations of all entities are marked. The second method uses a seeded version of latent Dirichlet allocation (SLDA) [5] where entity types in the original lists are used as prior information. The third method, which we call as *seeded feature propagation* (SFP), is based on representing entities and entity types as context vectors, where context features are propagated from unambiguous entities to entity types, and then similarity between each entity and entity type is computed and normalized to obtain probabilities.

We show the contribution of each entity weighting method on the SLU relation detection task. Specifically, our goal is to find all relations invoked in a user utterance (i.e., `film.directed_by` in "James Cameron movies") as well as to convert them to search queries to the back-end knowledge sources. In our previous work, we proposed methods to bootstrap relation detection models from web documents [6] and discovering new relation types from large text corpora [7]. Semantic knowledge graphs have also been used for SLU semantic parsing tasks in dialog systems [8, 9, 10, 11] and question answering [12, 13, among others].

In the next sections, we first describe the semantic resources used and the relation detection task. Then, we present methods for estimating weights for entities and their types. We show results comparing all methods for both unsupervised and supervised learning for the relation detection task.

2. Semantic Graphs

The Semantic Web is a collaborative movement aiming at converting the unstructured and semi-structured documents into a structured semantic network [14, 15, 16]. In 1997, W3C first defined the Resource Description Framework (RDF), a simple yet very powerful triple-based representation for the semantic web. As RDFs became more popular, triple stores (referred as semantic knowledge graphs) covering various domains have emerged, such as Freebase [1] and YAGO2 [17]. In 2008, W3C proposed the SPARQL RDF query language to retrieve and manipulate the data in knowledge bases. In this work, we target mapping natural language spoken queries addressed to a conversational agent into SPARQL queries to Freebase. In our approach, this task includes linking entities in user queries to the semantic graph entities and determining the relations that need to be included in the SPARQL query as described in the next section.

3. Relation Detection for Language Understanding

Relation detection task was originally formulated as part of Message Understanding Conferences (MUC) [18], and evaluated through NIST knowledge base population tracks [19], and aims to detect and classify instances of relations, where a relation is defined as a meaningful connection between two entities [20]. Since then, there has been a wide body of work, mainly related to detecting relations between two candidate entities in natural language text using supervised and unsupervised classification based methods [21, 22, 23, 24, 25, 26, among others] and relation discovery from natural language text [27, 28, among others].

In this paper, we propose a relation detection task for spoken language understanding that aims to detect all relations invoked in user utterances, with the goal of automatically forming SPARQL queries that correspond to user utterances. Each relation implies the presence of a triple in the

USER UTTERANCE:	SPARQL QUERY (simplified):
Show me horror movies by Steven Spielberg	SELECT ?movie WHERE { ?movie film.genre "Horror". ?movie film.directed_by "Steven Spielberg". }
Who is the director of Avatar?	SELECT ?person WHERE { ?movie film.name "Avatar". ?movie film.directed_by ?person. }

Figure 1: Example user utterances with their SPARQL mapping.

SPARQL query to a back-end knowledge source, and the entity values in these triples are then resolved using the words of the utterance. For example, in Figure 1, relation detection aims to find detecting film.genre and film.directed_by relations, given the utterance "Show me horror movies by Steven Spielberg". Once these relations are detected, two triples are invoked: `<?movie, film.genre, ?genre>` and `<?movie, film.directed_by, ?director>`. The values of the variables in these triples, such as `?movie` and `?genre`, are looked up in the utterance and included in the query to the back-end knowledge source. An advantage of this approach is that, even when the value of a variable is not extracted because of a system error, the movies would be listed with their genre information, enabling better error recovery strategies for conversational systems.

We treat relation detection as a multi-class, multi-label (i.e. each utterance can invoke more than one relation) classification problem, where the goal is to find the most probable relations given a user utterance.

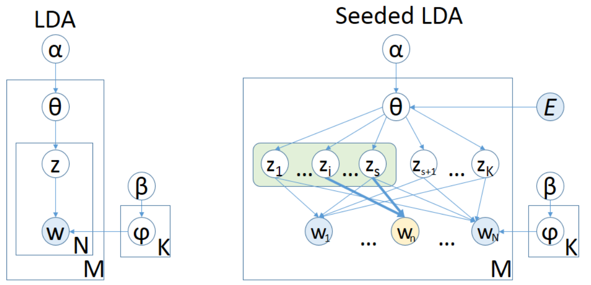

Figure 2: $z_1, \ldots, z_s$ represent entity types from the semantic graph. Yellow background shows entity tokens (i.e., w_n), thick links show entity type information encoded in the semantic graph (i.e., w_n is marked in the graph as an entity of two types z_i and z_s). E is the prior knowledge added as labeled random variable to the graphical model.

4. Approach

We investigate three methods for weighting entity types: one based on a fully populated semantic graph, and two based on natural language text corpora.

4.1. Entity Type Weights from the Populated Semantic Graph (SG-weighted)

For each entity surface form, e_i in the semantic graph, we count the number of times the specific entity is observed with a specific type, t_j, denoted by $C(e_i, t_j)$ in the populated knowledge. Here, by type, we mean that the entity was observed as the object of a relation associated with the type. For example, if an entity was observed as an object of "film.directed_by" relation, we consider that the entity was used as type "film.director". Then, we estimate a weight for each entity and type pair as:

$$P_{SG}(t_j|e_i) = \frac{C(e_i, t_j)}{\sum_{k \in T} C(e_i, t_k)}$$

where T denotes the set of all possible entity types. For example, if "Brad Pitt" is listed as a producer of 15 movies and as starring in 45 movies, a weight of 0.25 is assigned to "film.producer" and 0.75 is assigned to "film.actor".

4.2. Seeded LDA (SLDA-*-weighted)

Mixture modeling of documents into topical semantic clusters is proven to be effective for SLU tasks, where the goal is finding the global aspects such as domain, topic, or intent of a given utterance [29, 30, 31]. Our seeded topic modeling approach is based on the semi-supervised LDA presented in [32, 33], which is depicted in Fig. 2 graphically in comparison to the standard LDA. The prior information we use for the seeded LDA (SLDA) is defined by the input entity type clusters, while capturing the entity type distributions in our dataset. In the figure, z represent the latent entity type variables and w represent the ngram tokens in sentences, where words corresponding to entities are joined into a single token (i.e., "brad pitt" is mapped to "brad_pitt".)

Our model views each sentence invoking a mixture of entity types. We inject the entity information we obtain from the input list of entities and their types as prior knowledge into the SLDA model as labeled latent entity types. Specifically, we obtain a matrix of entity-type by entity realizations matrix, $\mathbf{E} \in I^{t \times s}$. This matrix contains for each t different n-grams (entity realization) obtained from the knowledge graph, a binary vector $\vec{e}_{\mathbf{n}}$={0,0,1,1,...,0,0} per each s different entity type, i.e., $\vec{e}_{n=1,..,t} \in I^s$. Thus, in Figure 2, the blank circles indicate latent variables, whereas blue and yellow filled circles indicate known variables. $\mathbf{E}$ is the prior knowledge injected as binary matrix and s is the number of latent entity types corresponding to known entity clusters (depicted as the green plate on the graph).

In our model, we attribute each ngram (in a given document) to a possible semantic entity type. We also would like to build a more focused model, where there is a one-to-many map between the semantic entity classes and latent topics. To achieve this, we use an informative prior during Gibbs sampling, which pulls word-entity relations from the entity relation matrix. Specifically, at training time, we provide prior entity relation matrix $\mathbf{E}$, for ngrams, which we know *a priori* that they correspond to one or more entity types in our corpus. For example, an actor name such as "*brad pitt*" in a given sentence, "..*in the movies directed by brad pitt*.." can be sampled from "film.actor" type (one of the designated s types) or it could also be sampled from "film.producer" type (also one of the designated s topics).

We sample the ngrams which do not exist in the entity relation matrix, i.e., $\vec{e}_{\mathbf{n}} \in \mathbf{E}$, from those topics designated for the semantic entity classes, namely the topics corresponding to entity types that get value 1 in the $\vec{e}_{\mathbf{n}}$ vector. Similarly, for the unlabeled ngrams, we let the algorithm decide which topic that word should belong to.

During model training and inference, we use this entity type matrix as restrictive information when generating each word in each sentence. We reserve s number of latent topics $z_1, \ldots, z_s$ to sustain a correspondence between the latent topics and the semantic labels (entity types) as shown in the graph representation of SLDA. The rest of the topics, $z_{s+1}, \ldots, z_K$ may or may not correspond to any entity type in our corpus.

After training the SLDA model, we calculate a probability of each type, given an entity, based on the final entity type assignments of the training corpus.

4.3. Seeded Feature Propagation (SFP-*-weighted)

SFP forms initial entity type cluster representations from unambiguous entities similarly to earlier work on word sense disambiguation [34] and computes similarity of each ambiguous entity to these clusters. This method represents each entity e_i with a vector E_i of $|V|$ terms, where in this case, V denotes the set of ngrams in the vocabulary of the corpus. Each term in this vector could be a frequency of the term, or other term weights. The entities can also be represented as entity embeddings [35, 36]. In this work, we compute term weights from the context where these entities occur in the large corpus.

Similarly, we represent entity types as vectors, T_j of $|V|$ terms. The type vectors are formed by propagating context features from all the entities that are unambiguously marked as having a single type in the semantic graph:

$$T_j = \sum_k E_k$$

where E_k are the vectors of the entities that are marked only as type t_j in the original entity type list.

Then, the probability of being in type t_j for an entity e_i is computed as:

$$P_{VS}(t_j|e_i) = \frac{exp(SIM(E_i, T_j))}{\sum_{m \in d_i} exp(SIM(E_i, T_m))}$$

where SIM denotes a similarity measure, and we use cosine similarity in the experiments. The set d_i is the set of all types assigned to entity e_i in the original entity type list.

Query Statistics	Training	Test
No. with SPARQL annotations	3,338	1,086
% with no relation (i.e. entity only)	10.1%	9.1%
% with 1 relation	70.4%	69.2%
% with 2 relations	10.2%	10.7%
% with 3 or more relations	1%	1.6%
% not covered by graph	8.3%	9.4%

Table 1: Relation detection data sets used in the experiments.

In this work, we do not perform iterations that update entity type vectors as ambiguous entities are associated with them as in [34], but our method can be extended as such.

5. Experiments

5.1. Data Sets

All our experiments use list of entities from the publicly available Freebase semantic knowledge graph. The list includes 820K entities of 78 entity types, including movie names, actors and release dates.

For estimating entity type weights with SLDA and SFP methods, we use two large corpora: Wikipedia movie domain documents (wiki) and movie domain search queries from Bing search query click logs (QCL) from a period of 3 months (October-December, 2013). The Wikipedia film domain document set contains 592K sentences and 10.6 million words. These documents were segmented into sentences using the Splitta tool [37] and used in training SLDA (SLDA-wiki) and SFP (SFP-wiki). The resulting entity dictionary includes weights over different entity types for the 182K entities observed in the Wikipedia documents. From QCL, we only use queries whose users clicked on movie domain Wikipedia web pages, and excluded queries that are shorter than 8 words, as there may not be enough context in such queries. The resulting data set contains 540K queries and 5.1 million words. Similarly, we used queries in training SLDA (SLDA-QCL) and SFP (SFP-QCL). Only 45K of the Freebase entities were observed in these queries, hence the SLDA and SFP methods estimate weights for only this subset.

We experimented with the formed dictionaries and associated entity type weights in relation detection experiments. The relation detection data sets are crowd-sourced utterances addressed to a conversational agent and are described in Table 1. Both the training and test sets were manually annotated with SPARQL queries, which we used to extract the relation annotations. Similar to a categorization of web search queries [38], we created a categorization of the conversational agent queries. Note that, in such natural language query data sets, the distributions of entity bearing, and entity+relation bearing user utterances are very different than those of web search queries. Around 10% of the training and test sets do not include invocation of a relation. These are mainly queries with just an entity (such as "Brad Pitt"). Most (around 70%) of the queries include only a single relation (such as, `film.genre` in "find funny movies"). The data sets also include movie domain queries that are not covered by the semantic graph (i.e., the invoked relations are not encoded in the semantic graph, such as "add this movie to my queue").

5.2. Results

We train a multi-class, multi-label classifier that estimates relations for each user utterance using icsiboost [39]. We extract word unigrams, bigrams, and trigrams as classification features. When a dictionary is available, we match ngrams in the example

Method:	Unsupervised			Supervised
Training Data:	None		Unlabeled	Labeled
Data Set:	Training	Test	Test (1 iter)	Test (Avg. 3)
No Dict.	-	-	-	84.1%
Unweighted Dict.	20.2%	21.2%	-	82.3%
SG-weighted Dict.	35.6%	34.6%	*38.3%*	*85.4%*
SLDA-wiki-weighted Dict.	34.6%	34.0%	**37.3%**	**84.9%**
SLDA-QCL-weighted Dict.	31.2%	30.9%	32.3%	84.7%
SFP-wiki-weighted Dict.	32.2%	30.9%	35.7%	**85.0%**
SFP-QCL-weighted Dict.	28.4%	31.5%	32.8 %	84.5%

Table 2: F-measure results of unsupervised learning experiments. *-wiki-* and *-QCL-* are experiments where Wikipedia documents and web queries are used for training, respectively. The semantic graph upperbounds and the best results are indented and boldfaced.

utterance with entities. When an ngram in the utterance matches an entity, we use the highest weighted type for the matching entity and its weight as additional features. For the unweighted dictionary, we use presence of all matching entity types as binary features. For evaluation, we compute relation detection F-measure.

5.2.1. *Unsupervised Learning*

We first experiment with fully unsupervised learning, which simply matches all word ngrams in user utterances to the list of entities. For each matching entity, the highest weight type is added to the list of relations for the given utterance. For the unweighted dictionary, we use all types of all the matching entities. The first two result columns of Table 2 show results from these experiments on relation detection training and test set examples. The third column shows results from unsupervised learning experiments in the presence of unlabeled training examples. In these results, the annotations from the first result column are used as labels of the training examples, and a classifier is trained using those. All methods improve significantly in comparison to the unweighted entity and entity type dictionaries. The best F-measure we achieve using fully unsupervised learning is with the SG-weighted entity types. The F-measure with SLDA-wiki method, which does not require a fully populated semantic graph is very close to the SG-weighted method.

5.2.2. *Supervised Learning Experiments*

We also experiment with supervised learning, where a labeled set of examples are used for training the relation detection classifier. Since our data sets are small, we experimented with 3 random ways of splitting the training and test sets and averaged results over those. Final column of Table 2 shows F-measure from these experiments. When labeled training data is available, the use of an unweighted dictionary may even hurt the relation detection F-measure (82.3%) over use of word ngrams only (84.1%). All methods for weighting entity types improve over using ngrams only. Figure 3 shows F-measure with varying sizes of labeled training sets (where at each point, 3 random subsets of training examples are used, resulting in averaging over 9 experiments). As seen, over all training subsets, SG-weighted and SLDA-wiki perform the best.

5.3. Discussion

The Wikipedia documents include 22.2% and search queries include 5.5% of all the entities. We tried adding the remaining entities to the learned dictionaries with weights from the SG-weighted dictionaries, however did not achieve any improvements in performance over using the SG-weighted dictionary by itself. Similarly, we analyzed if the weights in one of the dictionaries were better than others. We reduced the list of entities in the SG-weighted dictionary to the ones that were observed in the training corpora (hence included in the other methods). We obtained only small improvements from this over the full dictionary, showing that filtering infrequent entities (as found from large text corpora) from the full set of entities in the knowledge graph may help relation detection.

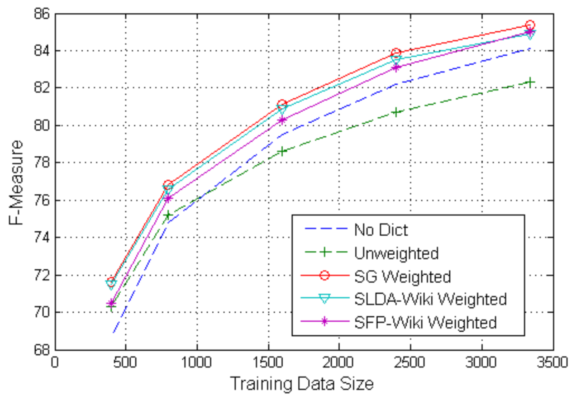

Figure 3: F-measure learning curves over varying training data set sizes, from supervised classification experiments. Experiments are performed over 3 different training and test subsets, and 9 F-measures are averaged for each point in the plot.

Note that, the relation detection approach in this paper focuses on entities in user utterances and relations implied by them. Hence, relations not implied by entities, such as the `film.directed_by` in the second example of Figure 1 are not targeted in this work. Such relations can be found by our previous approach that mines patterns from web documents, using pairs of entities [6]. Our future work includes joining the two approaches.

6. Conclusions

We present three methods for computing entity type weights to enrich semantic knowledge graph entities with probabilistic weights. We show the use of such weights in the SLU relation detection task. All three methods result in improvements in relation detection performance. Even when a fully populated semantic graph is not available, our proposed methods perform significantly better relation detection than using no dictionary or an unweighted dictionary.

7. References

[1] 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.

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

[3] 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.

[4] 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.

[5] D. M. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[6] D. Hakkani-Tür, L. Heck, and G. Tur, "Using a Knowledge Graph and Query Click Logs for Unsupervised Learning of Relation Detection," in *Proceedings of the ICASSP*, 2013.

[7] D. Hakkani-Tür, A. Celikyilmaz, L. Heck, and G. Tur, "A Weakly-Supervised Approach for Discovering New User Intents from Search Query Logs," in *Proceedings of Interspeech 2013*, Lyon, France, 2013.

[8] G. Tur, M. Jeong, Y.-Y. Wang, D. Hakkani-Tür, and L. Heck, "Exploiting semantic web for unsupervised statistical natural language semantic parsing," in *Proceedings of Interspeech 2012*, Portland, Oregon, 2012.

[9] L. Heck and D. 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.

[10] L. Heck, D. Hakkani-Tür, and G. Tur, "Leveraging knowledge graphs for web-scale unsupervised semantic parsing," in *Proceedings of Interspeech, International Speech Communication Association*, 2013.

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

[12] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on Freebase from question-answer pairs," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[13] Q. Cai and A. Yates, "Large-scale semantic parsing via schema matching and lexicon extension," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013.

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

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

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

[17] J. Hoffart, M. Fabian, S. Berberich, and G. Weikum, "YAGO2: a spatially and temporally enhanced knowledge base from wikipedia," in *Commununications of ACM*, 2010.

[18] N. Chinchor, "Overview of MUC-7," in *Proceedings of the Seventh Message Understanding Conference*, 1998, pp. 178–185.

[19] P. McNamee and H. T. Dang, "Overview of the tac 2009 knowledge base population track," in *Text Analysis Conference (TAC)*, vol. 17, 2009, pp. 111–113.

[20] A. Culotta and J. Sorensen, "Dependency tree kernels for relation extraction," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

[21] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel, "A novel use of statistical parsing to extract information from text," in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL)*, 2000, pp. 226–233.

[22] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *The Journal of Machine Learning Research*, vol. 3, pp. 1083–1106, 2003.

[23] R. Bunescu and R. Mooney, "A shortest path dependency kernel for relation extraction," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP)*, 2005, pp. 724–731.

[24] S. Zhao and R. Grishman, "Extracting relations with integrated information using kernel methods," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, 2005, pp. 419–426.

[25] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL*, 2009.

[26] F. Wu and D. Weld, "Autonomously semantifying wikipedia," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM)*, 2007, pp. 41–50.

[27] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction for the web," in *Proceedings of IJCAI*, vol. 7, 2007, pp. 2670–2676.

[28] J. Betteridge, A. Carlson, S. Hong, E. Jr., E. Law, T. Mitchell, and S. Wang, "Toward never ending language learning." in *AAAI Spring Symposium: Learning by Reading and Learning to Read*, 2009, pp. 1–2.

[29] A. Celikyilmaz, D. Hakkani-Tür, and G. Tur, "Multi-domain spoken language understanding with approximate inference," in *Proceedings of Interspeech*, 2011.

[30] A. Celikyilmaz and D. Hakkani-Tür, "A joint model for discovery of aspects in utterances," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 330–338.

[31] D. Lee, M. Jeong, K. Kim, and G. G. Lee, "Unsupervised modeling of user actions in a dialog corpus," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[32] A. Celikyilmaz and D. Hakkani-Tür, "Discovery of topically coherent sentences for extractive summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011, pp. 491–499.

[33] G. Tur, A. Celikyilmaz, and D. Hakkani-Tür, "Latent semantic modeling for slot filling in conversational understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8307–8311.

[34] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics (ACL)*, 1995, pp. 189–196.

[35] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *AAAI*, 2011.

[36] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 926–934.

[37] D. Gillick, "Sentence boundary detection and the problem with the U.S." in *Proceedings of the NAACL*, 2009.

[38] J. Pound, A. Hudek, I. Ilyas, and G. Weddell, "Interpreting keyword queries over web knowledge bases," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 305–314.

[39] B. Favre, D. Hakkani-Tür, and S. Cuendet, "Icsiboost," http://code.google.come/p/icsiboost, 2007.