KNOWLEDGE GRAPH INFERENCE FOR SPOKEN DIALOG SYSTEMS

Yi Ma[⋆1] *Paul A. Crook*[†] *Ruhi Sarikaya*[†] *Eric Fosler-Lussier*[⋆]

[⋆] The Ohio State University, Columbus, Ohio 43210, USA
[†] Microsoft Corporation, Redmond, Washington 98052, USA

ABSTRACT

We propose Inference Knowledge Graph, a novel approach of remapping existing, large scale, semantic knowledge graphs into Markov Random Fields in order to create user goal tracking models that could form part of a spoken dialog system. Since semantic knowledge graphs include both entities and their attributes, the proposed method merges the semantic dialog-state-tracking of attributes and the database lookup of entities that fulfill users' requests into one single unified step. Using a large semantic graph that contains all businesses in Bellevue, WA, extracted from Microsoft Satori, we demonstrate that the proposed approach can return significantly more relevant entities to the user than a baseline system using database lookup.

***Index Terms*—** Knowledge graph, spoken dialog system, Markov Random Fields, linked big data

1. INTRODUCTION

Within the field of spoken dialog systems (SDSs) for task orientated conversations, the problem of accurately tracking the user's goal (*e.g.* finding restaurants that satisfy a number of user constraints) has received considerable attention in the literature [1, 2]. One promising branch of research has focused on the statistical modelling of uncertainties and ambiguities encountered by dialog managers (DMs) due to Automatic Speech Recognition (ASR) and Spoken Language Understanding (SLU) errors, and ambiguity in natural language expressions. Included among the most successful statistical approaches are graphical models, *e.g.* [3, 4, 5]. In such models a DM computes a probability distribution over the set of possible user goals, referred to as its *belief*, and acts with respect to the distribution rather than the most likely goal. To make the update of such probabilistic graphical models tractable, the graphs are commonly factorized [4, 6] and approximate inference methods are applied. Approximate inference methods range from partitioning of probability distributions and applying handwritten transition likelihoods/update rules [5] to highly factorizing the graphical model and applying an inference update method such as loopy belief propagation or blocked Gibbs sampling [3, 4]. In all these approaches, the aim is typically to track the user's goal in terms of attributes that can be used to describe and look up actual entities of interest in some underlying database. For example, in a restaurant search scenario the DM will track the cuisine and location that the user is requesting (*e.g. Italian* restaurants in the *downtown* area) and use a separate database lookup to inform itself as to whether entities matching these requirements exist and return results to the user.

With the emergence of conversational personal assistants on mobile devices, *e.g.* Siri, Google Now and Microsoft's Cortana, there has been a surge of interest in exploiting web search resources, especially the large Resource Description Framework (RDF) semantic knowledge bases (also known as a *semantic knowledge graphs*), to reduce the manual work required in expanding SDSs to cover new domains, intents or slots [7, 8, 9, 10]. An example of a popular and well known semantic knowledge graph is Freebase [11]. A semantic knowledge graph represents information using triples of the form subject-predicate-object where in graph form the predicate is an edge linking an entity (the subject) to its attributes or another related entity, see Figure 1. Other semantic knowledge graphs are Facebook's Open Graph, Google's Knowledge Graph and Microsoft's Satori. The latter two contain information covering many domains (people, places, sports, *etc.*) and underpin the entity related results that are generated by Google's and Microsoft Bing's search engines, *e.g.* a search on "Leonardo da Vinci" will display related art, such as the Mona Lisa, and also other famous artists.

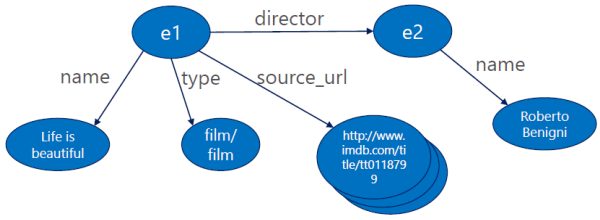

Fig. 1. Part of a semantic knowledge graph representing the relationships, described as RDF triples, between the entities Roberto Benigni (e1) and the film "Life is Beautiful" (e2).

Work on utilising semantic knowledge graphs in SDSs has largely focused on automating the collection of language corpora aimed at training SLU models, *e.g.* intent detection [8], relationship detection [9] and entity detection [7]. Our approach in this paper solves a different problem. We con-

[1] Work conducted while interning at Microsoft.

sider whether it is possible to specify a transformation from an existing semantic knowledge graph to a graphical model, specifically a Markov Random Field (MRF) [12, Chp.8], over which it is possible to maintain a probability distribution representing the likely state of the dialog. Such an approach, if possible, would reduce the manual effort involved in the design and development of statistical DMs in that manual factorization of the graphical model to suit the domain would be avoided through simply inheriting the existing factorization of the semantic knowledge graph.

The remainder of the paper is set out as follows. Related work is described in more detail in Section 2; Section 3 uses a toy knowledge graph to illustrate the proposed approach. Section 4 describes the Inference Knowledge Graph algorithm. Section 5 the experimental setup and results and Section 6 concludes and discusses future work.

2. PREVIOUS WORK

To our knowledge, transforming semantic knowledge graphs to graphical models for the purpose of dialog state tracking is a new area of work, however, it is constructive to contrast our approach with related works in the areas of dialog management and exploiting large semantic knowledge graphs.

The Bayesian update of dialog state (BUDS) dialog system [4] uses a highly factorized graphical model to maintain a belief distribution reflecting the state of the dialog. This distribution is updated using the approximate inference method of loopy belief propagation [13]. The factorization of the graph for BUDS is manually designed with the weights on the graph being updated using the Natural Actor Belief Critic [14] learning algorithm, based on training with simulated users. The graphical model maintains a distribution over slots (attribute values) that exist in the Tourist Information domain. Entities, such as restaurants, are looked up from a separate database. In contrast, our approach automatically creates a factored graph from the existing semantic knowledge graph and embeds the entities within the factored graph.

A number of approaches have been presented that use semantic knowledge graphs to help automate the construction of SDSs. Many of the approaches utilize the entity and relationship knowledge encoded in the graph to collect training data in a distantly supervised manner, *e.g.* filtering web browser query-click logs using the existence of matching entity names or relationships in the semantic graph [7], or collecting natural language surface forms for relationships that exist within the graph by composing web search queries [8]. Likelihoods of the existence of intents or slots within a domain or of the relationships between entities have also been computed from semantic knowledge graphs [9, 10] or used to seed latent Dirichlet analysis that is applied to another corpora, such as Wikipedia [9]. The data thus collected is used to build various conversational and SLU models, *e.g.* intent, slot, entity and relation detection models. This contrasts with our approach as we attempt to use the graph structure at runtime rather than deriving training data for off-line development of models. One approach [7] learns CRF models that use previous turn context to perform entity extraction and entity type labeling. While the entity extraction model's output is superficially similar to that presented in this paper, the target output is different. Our aim is to track the entities that correspond to the user's overall goal (which is a function of the whole dialog) and not the accurate identification of entities that occur within each utterance in the dialog.

The simplifying assumptions made in this initial experiment mean that the proposed approach somewhat resembles spreading activation approaches for information retrieval [15], which have been applied to web retrieval [16] and large semantic knowledge graphs [17]. Our approach differs in the use of a MRF representation and Mean Field Variational inference. These techniques allow our approach to be potentially far more expressive in the probabilistic relationships that we encode in the MRF. For example, while in the currently implementation we have a one-to-one mapping between *edge* factors in the MRF and semantic graph *edges*, and these factors have equal potentials in both directions, our future plans include exploring more sophisticated mappings between the two graphs including MRF factors that have unequal potentials or which cover multiple semantic graph edges. This will allow the MRF to more accurately capture the interactions between entities and attributes.

Although the Dialog State Tracking Challenge 2 & 3 [2] would be an ideal test bed for our approach, a semantic knowledge graph covering the tourist and restaurant domains in Cambridge UK was not readily available.

3. A TOY EXAMPLE

We start from a simple toy example to deduce the generalized algorithm. Imagine in a city with only three restaurants (`teletype font` indicates a node in the graph and the attribute nodes – *Cusine* and *Price* – are linked to their corresponding restaurant nodes as shown in Figure 2.):

1. Restaurant `Wild Ginger` is `Expensive` and serves `Asian Fusion` food
2. `John Howie Steak House` is also `Expensive` and serves `American` cuisine
3. `McDonald's` serves `American` fast food and is `Cheap`

We assume in the first turn the user said '*I want an **expensive American** restaurant*.' and suppose the ASR correctly recognized the spoken utterance and the SLU component identified the value for attribute type *Price* is `Expensive` and the value for attribute type *Cuisine* is `American`.

The first step is to convert the semantic knowledge graph into a MRF factor graph by introducing factor potentials over nodes and edges in the original graph.[2] Every node in the knowledge graph becomes a binary (*on* or *off*) random variable x_n that indicates how likely the node represents the

[2]The backbone of the semantic knowledge graph in this example has the shape of a chain but larger graphs will contain multiple loops.

Table 1. Pairwise edge potentials (Node 1 is connected with an undirected edge to Node 2).

Node1/Node 2	on	off
on	$P > 1$	1
off	1	$P > 1$

user's goal, *i.e.* is *on*. Potential functions f_i in the resulting MRF are defined as follows. No prior bias is imposed on nodes and their potentials $f_i(x_n)$ are set such that they are equally likely to be *on* or *off* when no evidence is observed. Pairwise edge potentials $f_j(x_n, x_m)$ are defined as in Table 1. For $P > 1$ two nodes connected by an edge are more likely to be the same state, *i.e.* both *on* or *off*, than in opposite states. This encourages *on* or *off* status to propagate from the evidence nodes to all the other connected nodes in the MRF.

The second step is to create evidence nodes `American'` and `Expensive'` (where the symbol `'` is used to indicate an evidence node) and append them to the corresponding original attribute graph nodes `American` and `Expensive` respectively as shown in Figure 2. Since evidence nodes `American'` and `Expensive'` are observed, they are clamped *on*. With clamped evidence nodes, the new graphical model becomes a conditional MRF. Once the evidence nodes from the current user turn have been attached to the MRF, conditional inference is performed and the marginal probability $p(x_n) = \sum_{\mathbf{x} \setminus x_n} p(\mathbf{x})$ for every variable node x_n in the MRF is computed[3]; where $\mathbf{x} \setminus x_n$ is the set of all $\mathbf{x}$ with x_n excluded and $p(\mathbf{x}) = \Pi_j f_j$ is the joint distribution. We use the UGM toolkit [18] for inference. Lastly, all the nodes in the MRF are ranked based on $p(x_n)$ and the top K entity nodes (*e.g.* restaurant nodes) are presented back to the user.

Figure 2 shows the marginal probability heat map for the likelihood of each node being *on* after exact inference.[4] If we order the restaurants based on their marginal probabilities, we can see that `John Howie Steak House` has the highest marginal likelihood of being *on* compared to the other two restaurants `Wild Ginger` and `McDonald's`. This aligns with what we expect: observations `American'` and `Expensive'` are combined together to raise the probability of `John Howie Steak House` being *on*, even though `Wild Ginger` is also `Expensive` and `McDonald's` also serves `American` food.

At this point we are done with the first user turn and the same process (the above-mentioned steps) can be repeated for the second user turn and so on. The evidence nodes will accumulate in the MRF as the dialog proceeds.

Even though this is a toy example, it demonstrates a proof of concept of transforming a semantic graph into a MRF and performing inference which results in a set of marginals that can represent the likely user's goal state. A dialog policy could then be trained to act with respect to the distribution of marginal values (similar to BUDS SDS[4] whose policy is trained with respect to marginals).

[3]This can be done efficiently using message passing approaches.

[4]The graphical model for this example is simple enough to allow exact inference.

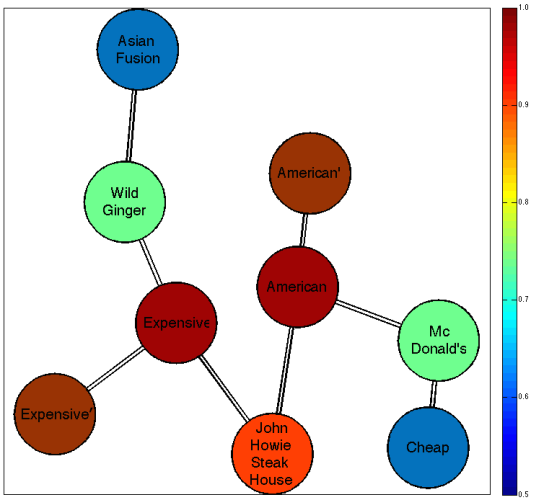

Fig. 2. A heat map of marginal probability for the value *on* for the conditional MRF after inference on first user turn with evidence nodes `American'` and `Expensive'` appended. The hotter the color, the more likely the node will be *on* (each node can take two values – *on*/*off* – with equal prior probability). The color of the evidence nodes is brown – the hottest – because, as observations, their probability of being *on* is clamped to 1.0 throughout inference.

P controls the spread of influence between nodes. Its value is dependent on the graph size and structure, and the accuracy errors in approximate inference. In this work it was manually tuned to ensure the effect of observations spread further than immediate neighboring nodes whilst avoiding uniform saturation of large network cliques due to the presence of some highly interconnected nodes.

4. INFERENCE KNOWLEDGE GRAPH ALGORITHM

We formally propose the complete Inference Knowledge Graph (IKG) algorithm as follows:

1. Convert the semantic knowledge graph into a MRF:
   (a) Every node in the knowledge graph becomes a binary random variable
   (b) Define potential functions over the nodes and edges
2. Append evidence nodes to the MRF:
   (a) Map SLU slot values to corresponding nodes
   (b) Clamp the evidence nodes to be *on* (when the user provides information or confirms) or *off* (when the user issues a negation)
3. Perform exact or approximate inference on the MRF to calculate the marginal probability for each node
4. Sort all the nodes by their marginal likelihood of being *on* and return top K entity nodes to the user
5. Apply some transition function $\mathcal{T}$ to project network marginals into the next turn[5] and repeat from 2.

[5]In this paper we assume an identity function resulting in accumulation of observations over turns.

Table 2. Evaluation results – average fraction of annotated SLU slots covered by top K business entities.

Top K(=1,2,3,5,7) Entities	Transcription SLU		ASR 1-best SLU		ASR N-best SLU
	Baseline	Inference Knowledge Graph	Baseline	Inference Knowledge Graph	Inference Knowledge Graph
Top 1	0.675	**0.694**	0.620	**0.638**	**0.634**
Top 2	0.676	**0.698**	0.620	**0.642**	**0.640**
Top 3	0.676	**0.700**	0.620	**0.644**	**0.644**
Top 5	0.676	**0.702**	0.622	**0.645**	**0.646**
Top 7	0.676	**0.703**	0.622	**0.646**	**0.647**

5. EXPERIMENTAL SETUP AND EVALUATION

If the resulting MRF is to form part of a DM, the distribution of marginals that is induced by the attachment of evidence nodes and inference step should track the user's intended goal during dialogs. As entities are embedded in the MRF, our expectation is that a list of the entities ordered by their associated marginal likelihoods will match against the requirements specified by the user. To automatically measure how closely the ordered list of entities match the user's requirements we adopt a surrogate measure for relevance that is motivated by the availability of annotated dialogs. For each dialog we collect the complete set of annotated slot values mentioned in all turns. Then for the top K entities generated by the MRF at the end of each dialog, we collect the set of attribute values associated with those entities. We then compute the fraction of annotated slot values that are matched by entity attribute values. In performing the matching we use a manually constructed dictionary to canonicalize slot values. This same dictionary is used to canonicalize slot values output by the SLU in order to attach them to the graph as evidence nodes.

To test our algorithm we extract a semantic subgraph from Satori that contains all the businesses in Bellevue, WA along with their attributes. The extracted graph has 43,208 nodes (18,142 business entity nodes and 25,066 attribute nodes) and 108,288 edges. There are 8 attribute types and each attribute type can have multiple values, as illustrated below:

Atmosphere: `Lunch Spot`, `Family-friendly Dining`, `Date Spot`, `Romantic`, ...

Cuisine: `American`, `Café`, `Sandwiches`, `Fast Food`, `Japanese`, ...

Price: `Cheap`, `Expensive`, ...

We evaluated our system using dialogs of real users interacting with Cortana – a personal assistant on Windows mobile phone – during the week of June 30th to July 4th, 2014. Since the selected subgraph only contains businesses in Bellevue, WA, we only used dialogs that are business or location related queries and either mention no absolute location or the absolute location provided by the user contains the keyword 'Bellevue'. This gave us a total of 12,607 dialogs, of which 6,647 are spoken dialogs, the remainder being typed input. We test three conditions; *(i)* using the complete set of typed and human *transcribed* spoken dialogs as input to the SLU, or for the 6,647 spoken dialogs *(ii)* use the ASR 1-best or *(iii)* ASR N-best output as input to the SLU.

We compare the IKG results against a database lookup baseline. From the results shown in Table 2 we can see that all graph methods outperform the baseline significantly (with $p < 0.05$). Although using ASR N-best does not further improve the system performance, it demonstrates that the IKG method has the ability to resolve noisy input. The occurrence of slots in the dialogs that cannot be matched by entity attributes, such the relative distance slot *nearby* as in *'find the **nearest** café'*, limits the maximum score that is achievable when using this measure.

To further understand the gain of our system we divide the 6,647 spoken dialogs into two parts; one part contains all the dialogs where the baseline returns empty results (327 dialogs), the other part includes the rest of the dialogs where baseline returns at least one business entity (6,320 dialogs). We calculate the accuracy of ASR 1-best SLU for both baseline and IKG systems on the two partitions. When the baseline returns at least one result, it and the IKG's scores are equally good. Therefore the gain is due to the IKG gracefully handling dialogs where baseline fails.[6] The IKG on the 327 dialogs ranges from 0.35 (for $K = 1$) to 0.50 (for $K = 5$).

We also note that during a dialog if a user only mentioned a business name, *e.g.*, a restaurant name, the baseline would return only one entity that matches that business. However, the graph would return a list of businesses that share similar attributes in addition to the one mentioned by the user.

6. CONCLUSION AND FUTURE WORK

Using a semantic graph containing all businesses in Bellevue, WA extracted from Microsoft's Satori, we demonstrate a novel approach of remapping semantic knowledge graphs into MRFs which results in a graphical model that can be used to successfully infer the user's goal state in a dialog. We show that the MRF model returns significantly more relevant entities to the user than a database lookup baseline.

Future directions include *(i)* reflecting ASR confidence scores in the factor potentials of evidence nodes, instead of clamping to *on* or *off*, which may improve ASR N-best performance, *(ii)* more sophisticated mappings to the MRF including non-uniform P or learned factor potentials (*e.g.* from dialog corpora) and *(iii)* modeling temporal transitions $\mathcal{T}$, *e.g.* goal change, between dialog turns.

[6] An example of where the baseline can fail is where SLU output contains both `Mexican` and `Japanese`, where one of them is a recognition error.

7. REFERENCES

[1] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black, "The dialog state tracking challenge," in *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2013.

[2] Matthew Henderson, Blaise Thomson, and Jason Williams, "Dialog state tracking challenge 2 & 3," *http://camdial.org/ mh521/dstc/*, 2013.

[3] Antoine Raux and Yi Ma, "Efficient probabilistic tracking of user goal and dialog history for spoken dialog systems," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, 2011, pp. 801–804.

[4] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.

[5] S. Young, M. Gašić, S. Keizer, F. Mairesse, B. Thomson, and K. Yu, "The Hidden Information State model: a practical framework for POMDP based spoken dialogue management," *Computer Speech and Language*, vol. 24, no. 2, pp. 150–174, 2010.

[6] Jason Williams, Pascal Poupart, and Steve Young, "Factored partially observable markov decision processes for dialogue management," in *Workshop on Knowledge and Reasoning in Practical Dialog Systems (IJCAI)*, 2005.

[7] Lu Wang, Larry Heck, and Dilek Hakkani-Tur, "Leveraging semantic web search and browse sessions for multi-turn spoken dialog systems," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, SLP Student Travel Grant Award.

[8] Larry Heck and Dilek Hakkani-Tur, "Exploiting the semantic web for unsupervised spoken language understanding," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 228–233.

[9] Dilek Hakkani-Tür, Asli Celikyilmaz, Larry Heck, Gokhan Tur, and Geoff Zweig, "Probabilistic enrichment of knowledge graph entities for relation detection in conversational understanding," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[10] Ali El-Kahky, Derek Liu, Ruhi Sarikaya, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck, "Extending domain coverage of language understanding systems via intent transfer between domains using knowledge graphs and search query click logs," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014.

[11] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2008, SIGMOD '08, pp. 1247–1250, ACM.

[12] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[13] Judea Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann Publishers Inc., San Fransisco, CA, US, 1998.

[14] Filip Jurčíček, Blaise Thomson, and Steve Young, "Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as pomdps," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 7, no. 3, pp. 6, 2011.

[15] Fabio Crestani, "Application of spreading activation techniques in information retrieval," *Artificial Intelligence Review*, vol. 11, no. 6, pp. 453–482, 1997.

[16] Fabio Crestani and Puay Leng Lee, "Searching the web by constrained spreading activation," *Information Processing & Management*, vol. 36, no. 4, pp. 585–605, 2000.

[17] Michael J Cafarella, Michele Banko, and Oren Etzioni, "Relational web search," in *Technical Report 06-04-02 University of Washington*, 2006.

[18] Mark Schmidt, "UGM: Matlab toolkit for undirected graphical models," `http://www.cs.ubc.ca/~schmidtm/Software/UGM.html`, 2011.