Generating Fine-Grained Open Vocabulary Entity Type Descriptions

Rajarshi Bhowmik and **Gerard de Melo**
Department of Computer Science
Rutgers University – New Brunswick
Piscataway, NJ, USA
{rajarshi.bhowmik, gerard.demelo}@cs.rutgers.edu

Abstract

While large-scale knowledge graphs provide vast amounts of structured facts about entities, a short textual description can often be useful to succinctly characterize an entity and its type. Unfortunately, many knowledge graph entities lack such textual descriptions. In this paper, we introduce a dynamic memory-based network that generates a short open vocabulary description of an entity by jointly leveraging induced fact embeddings as well as the dynamic context of the generated sequence of words. We demonstrate the ability of our architecture to discern relevant information for more accurate generation of type description by pitting the system against several strong baselines.

Figure 1: A motivating example question that demonstrates the importance of short textual descriptions.

1 Introduction

Broad-coverage knowledge graphs such as Freebase, Wikidata, and NELL are increasingly being used in many NLP and AI tasks. For instance, DBpedia and YAGO were vital for IBM's Watson! Jeopardy system (Welty et al., 2012). Google's Knowledge Graph is tightly integrated into its search engine, yielding improved responses for entity queries as well as for question answering. In a similar effort, Apple Inc. is building an in-house knowledge graph to power Siri and its next generation of intelligent products and services.

Despite being rich sources of factual knowledge, cross-domain knowledge graphs often lack a succinct textual description for many of the existing entities. Fig. 1 depicts an example of a concise entity description presented to a user. Descriptions of this sort can be beneficial both to humans and in downstream AI and natural language processing tasks, including question answering (e.g., *Who is Roger Federer?*), named entity disambiguation (e.g., *Philadelphia* as a city vs. the film or even the brand of cream cheese), and information retrieval, to name but a few.

Additionally, descriptions of this sort can also be useful to determine the ontological type of an entity – another challenging task that often needs to be addressed in cross-domain knowledge graphs. Many knowledge graphs already provide ontological type information, and there has been substantial previous research on how to predict such types automatically for entities in knowledge graphs (Neelakantan and Chang, 2015; Miao et al., 2016; Kejriwal and Szekely, 2017), in semi-structured resources such as Wikipedia (Ponzetto and Strube, 2007; de Melo and Weikum, 2010), or even in unstructured text (Snow et al., 2006; Bansal et al., 2014; Tandon et al., 2015). However, most such work has targeted a fixed inventory of types from a given target ontology, many

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 877–888
Melbourne, Australia, July 15 - 20, 2018.

of which are more abstract in nature (e.g., *human* or *artifact*). In this work, we consider the task of generating more detailed open vocabulary descriptions (e.g., *Swiss tennis player*) that can readily be presented to end users, generated from facts in the knowledge graph.

Apart from type descriptions, certain knowledge graphs, such as Freebase and DBpedia, also provide a paragraph-length textual abstract for every entity. In the latter case, these are sourced from Wikipedia. There has also been research on generating such abstracts automatically (Biran and McKeown, 2017). While abstracts of this sort provide considerably more detail than ontological types, they are not sufficiently concise to be grasped at a single glance, and thus the onus is put on the reader to comprehend and summarize them.

Typically, a short description of an entity will hence need to be synthesized just by drawing on certain most relevant facts about it. While in many circumstances, humans tend to categorize entities at a level of abstraction commonly referred to as basic level categories (Rosch et al., 1976), in an information seeking setting, however, such as in Fig. 1, humans naturally expect more detail from their interlocutor. For example, *occupation* and *nationality* are often the two most relevant properties used in describing a person in Wikidata, while terms such as *person* or *human being* are likely to be perceived as overly unspecific. However, choosing such most relevant and distinctive attributes from the set of available facts about the entity is non-trivial, especially given the diversity of different kinds of entities in broad-coverage knowledge graphs. Moreover, the generated text should be coherent, succinct, and non-redundant.

To address this problem, we propose a dynamic memory-based generative network that can generate short textual descriptions from the available factual information about the entities. To the best of our knowledge, we are the first to present neural methods to tackle this problem. Previous work has suggested generating short descriptions using predefined templates (cf. Section 4). However, this approach severely restricts the expressivity of the model and hence such templates are typically only applied to very narrow classes of entities. In contrast, our goal is to design a broad-coverage open domain description generation architecture.

In our experiments, we induce a new benchmark dataset for this task by relying on Wikidata, which has recently emerged as the most popular crowd-sourced knowledge base, following Google's designation of Wikidata as the successor to Freebase (Tanon et al., 2016). With a broad base of 19,000 casual Web users as contributors, Wikidata is a crucial source of machine-readable knowledge in many applications. Unlike DBpedia and Freebase, Wikidata usually contains a very concise description for many of its entities. However, because Wikidata is based on user contributions, many new entries are created that still lack such descriptions. This can be a problem for downstream tools and applications using Wikidata for background knowledge. Hence, even for Wikidata, there is a need for tools to generate fine-grained type descriptions. Fortunately, we can rely on the entities for which users have already contributed short descriptions to induce a new benchmark dataset for the task of automatically inducing type descriptions from structured data.

2 A Dynamic Memory-based Generative Network Architecture

Our proposed dynamic memory-based generative network consists of three key components: an input module, a dynamic memory module, and an output module. A schematic diagram of these are given in Fig. 2.

2.1 Input Module

The input to the input module is a set of N facts $F = \{f_1, f_2, \ldots, f_N\}$ pertaining to an entity. Each of these input facts are essentially (s, p, o) triples, for subjects s, predicates p, and objects o. Upon being encoded into a distributed vector representation, we refer to them as *fact embeddings*.

Although many different encoding schemes can be adopted to obtain such fact embeddings, we opt for a positional encoding as described by Sukhbaatar et al. (2015), motivated in part by the considerations given by Xiong et al. (2016). For completeness, we describe the positional encoding scheme here.

We encode each fact f_i as a vector $\mathbf{f_i} = \sum_{j=1}^{J} \mathbf{l_j} \circ \mathbf{w_j^i}$, where $\circ$ is an element-wise multiplication, and $\mathbf{l_j}$ is a column vector with the structure $l_{kj} = (1 - \frac{j}{J}) - (k/d)(1 - 2\frac{j}{J})$, with J being the number of words in the factual phrase, $\mathbf{w_j^i}$ as the embedding of the j-th word, and d as the dimensionality of the embedding. Details about how these factual phrases are formed for our data are

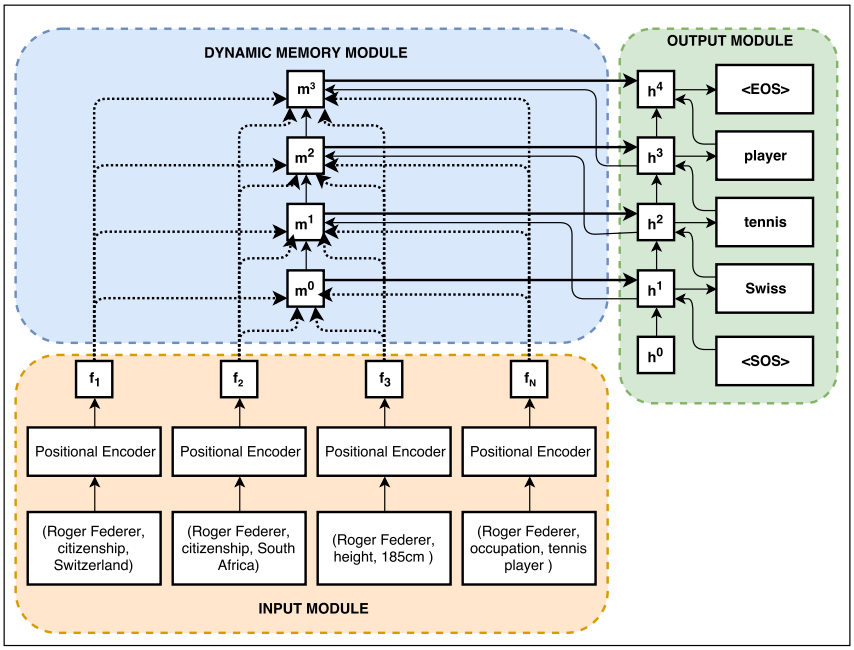

Figure 2: Model architecture.

given in Section 3.3.

Thus, the output of this module is a concatenation of N fact embeddings $\mathbf{F} = [\mathbf{f_1}; \mathbf{f_2}; \ldots; \mathbf{f_N}]$.

2.2 Dynamic Memory Module

The dynamic memory module is responsible for memorizing specific facts about an entity that will be useful for generating the next word in the output description sequence. Intuitively, such a memory should be able to update itself dynamically by accounting not only for the factual embeddings but also for the current context of the generated sequence of words.

To begin with, the memory is initialized as $\mathbf{m^{(0)}} = \max(\mathbf{0}, \mathbf{W_m F} + \mathbf{b_m})$. At each time step t, the memory module attempts to gather pertinent contextual information by attending to and summing over the fact embeddings in a weighted manner. These attention weights are scalar values informed by two factors: (1) how much information from a particular fact is used by the previous memory state $\mathbf{m}^{(t-1)}$, and (2) how much information of a particular fact is invoked in the current context of the output sequence $\mathbf{h}^{(t-1)}$. Formally,

$$\mathbf{x_i}^{(t)} = [|\mathbf{f_i} - \mathbf{h}^{(t-1)}|; |\mathbf{f_i} - \mathbf{m}^{(t-1)}|], \quad (1)$$

$$\mathbf{z_i}^{(t)} = \mathbf{W_2} \tanh(\mathbf{W_1 x_i}^{(t)} + \mathbf{b_1}) + \mathbf{b_2}, \quad (2)$$

$$a_i^{(t)} = \frac{\exp(\mathbf{z_i}^{(t)})}{\sum_{k=1}^{N} \exp(\mathbf{z_k}^{(t)})}, \quad (3)$$

where $|.|$ is the element-wise absolute difference and $[;]$ denotes the concatenation of vectors.

Having obtained the attention weights, we apply a soft attention mechanism to extract the current context vector at time t as

$$\mathbf{c}^{(t)} = \sum_{i=1}^{N} a_i^{(t)} \mathbf{f_i}. \quad (4)$$

This newly obtained context information is then used along with the previous memory state to update the memory state as follows:

$$\mathbf{C}^{(t)} = [\mathbf{m}^{(t-1)}; \mathbf{c}^{(t)}; \mathbf{h}^{(t-1)}] \quad (5)$$

$$\mathbf{m}^{(t)} = \max(\mathbf{0}, \mathbf{W_m C}^{(t)} + \mathbf{b_m}) \quad (6)$$

Such updated memory states serve as the input to the decoder sequence of the output module at each time step.

2.3 Output Module

The output module governs the process of repeatedly decoding the current memory state so as to emit the next word in an ordered sequence of output words. We rely on GRUs for this.

At each time step, the decoder GRU is presented as input a glimpse of the current memory state $\mathbf{m}^{(t)}$ as well as the previous context of the output sequence, i.e., the previous hidden state of the decoder $\mathbf{h}^{(t-1)}$. At each step, the resulting output of the GRU is concatenated with the context vector $\mathbf{c_i}^{(t)}$ and is passed through a fully connected

layer and finally through a softmax layer. During training, we deploy *teacher forcing* at each step by providing the vector embedding of the previous correct word in the sequence as an additional input. During testing, when such a signal is not available, we use the embedding of the predicted word in the previous step as an additional input to the current step. Formally,

$$\mathbf{h}^{(t)} = \text{GRU}([\mathbf{m}^{(t)}; \mathbf{w}^{(t-1)}], \mathbf{h}^{(t-1)}), \quad (7)$$

$$\tilde{\mathbf{h}}^{(t)} = \tanh(\mathbf{W_d}[\mathbf{h}^{(t)}; \mathbf{c}^{(t)}] + \mathbf{b_d}), \quad (8)$$

$$\hat{\mathbf{y}}^{(t)} = \text{Softmax}(\mathbf{W_o}\tilde{\mathbf{h}}^{(t)} + \mathbf{b_o}), \quad (9)$$

where $[;]$ is the concatenation operator, $\mathbf{w}^{(t-1)}$ is vector embedding of the previous word in the sequence, and $\hat{\mathbf{y}}^{(t)}$ is the probability distribution for the predicted word over the vocabulary at the current step.

2.4 Loss Function and Training

Training this model amounts to picking suitable values for the model parameters θ, which include the matrices $\mathbf{W_1}$, $\mathbf{W_2}$, $\mathbf{W_m}$, $\mathbf{W_d}$, $\mathbf{W_o}$ and the corresponding bias terms $\mathbf{b_1}$, $\mathbf{b_2}$, $\mathbf{b_m}$, $\mathbf{b_d}$, and $\mathbf{b_o}$ as well as the various transition and output matrices of the GRU.

To this end, if each of the training instances has a description with a maximum of M words, we can rely on the categorical cross-entropy over the entire output sequence as the loss function:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{M}\sum_{j=1}^{|\mathcal{V}|} y_j^{(t)} \log(\hat{y}_j^{(t)}). \quad (10)$$

where $y_j^{(t)} \in \{0, 1\}$ and $|\mathcal{V}|$ is the vocabulary size.

We train our model end-to-end using Adam as the optimization technique.

3 Evaluation

In this section, we describe the process of creating our benchmark dataset as well as the baseline methods and the experimental results.

3.1 Benchmark Dataset Creation

For the evaluation of our method, we introduce a novel benchmark dataset that we have extracted from Wikidata and transformed to a suitable format. We rely on the official RDF exports of Wikidata, which are generated regularly (Erxleben et al., 2014), specifically, the RDF dump dated 2016-08-01, which consists of 19,768,780 entities with 2,570 distinct properties. A pair of a property and its corresponding value represents a fact about an entity. In Wikidata parlance, such facts are called *statements*. We sample a dataset of 10K entities from Wikidata, and henceforth refer to the resulting dataset as WikiFacts10K. Our sampling method ensures that each entity in WikiFacts10K has an English description and at least 5 associated statements. We then transform each extracted statement into a phrasal form by concatenating the words of the property name and its value. For example, the (subject, predicate, object) triple (*Roger Federer*, *occupation*, *tennis player*) is transformed to *'occupation tennis player'*. We refer to these phrases as the *factual phrases*, which are embedded as described earlier. We randomly divide this dataset into training, validation, and test sets with a 8:1:1 ratio. We have made our code and data available[1] for reproducibility and to facilitate further research in this area.

3.2 Baselines

We compare our model against an array of baselines of varying complexity. We experiment with some variants of our model as well as several other state-of-the-art models that, although not specifically designed for this setting, can straightforwardly be applied to the task of generating descriptions from factual data.

1. **Facts-to-sequence Encoder-Decoder Model.** This model is a variant of the standard sequence-to-sequence encoder-decoder architecture described by Sutskever et al. (2014). However, instead of an input sequence, it here operates on a set of fact embeddings $\{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_N}\}$, which are emitted by the positional encoder described in Section 2.1. We initialize the hidden state of the decoder with a linear transformation of the fact embeddings as $\mathbf{h}^{(0)} = \mathbf{WF} + \mathbf{b}$, where $\mathbf{F} = [\mathbf{f_1}; \mathbf{f_2}; \ldots; \mathbf{f_N}]$ is the concatenation of N fact embeddings.

   As an alternative, we also experimented with a sequence encoder that takes a separate fact embedding as input at each step and initializes the decoder hidden state with the final hidden state of the encoder. However, this approach did not yield us better results.

[1] https://github.com/kingsaint/Open-vocabulary-entity-type-description

Table 1: Automatic evaluation results of different models. For a detailed explanation of the baseline models, please refer to Section 3.2. The best performing model for each column is highlighted in boldface.

Model	B-1	B-2	B-3	B-4	ROUGE-L	METEOR	CIDEr
Facts-to-seq	0.404	0.324	0.274	0.242	0.433	0.214	1.627
Facts-to-seq w. Attention	0.491	0.414	0.366	0.335	0.512	0.257	2.207
Static Memory	0.374	0.298	0.255	0.223	0.383	0.185	1.328
DMN+	0.281	0.234	0.236	0.234	0.275	0.139	0.912
Our Model	**0.611**	**0.535**	**0.485**	**0.461**	**0.641**	**0.353**	**3.295**

2. **Facts-to-sequence Model with Attention Decoder.** The encoder of this model is identical to the one described above. The difference is in the decoder module that uses an attention mechanism.

   At each time step t, the decoder GRU receives a context vector $\mathbf{c}^{(t)}$ as input, which is an attention weighted sum of the fact embeddings. The attention weights and the context vectors are computed as follows:

   $$\mathbf{x}^{(t)} = [\mathbf{w}^{(t-1)}; \mathbf{h}^{(t-1)}] \quad (11)$$

   $$\mathbf{z}^{(t)} = \mathbf{W}\mathbf{x}^{(t)} + \mathbf{b} \quad (12)$$

   $$\mathbf{a}^{(t)} = \text{softmax}(\mathbf{z}^{(t)}) \quad (13)$$

   $$\mathbf{c}^{(t)} = \max(\mathbf{0}, \sum_{i=1}^{N} a_i^{(t)} \mathbf{f_i}) \quad (14)$$

   After obtaining the context vector, it is fed to the GRU as input:

   $$\mathbf{h}^{(t)} = \text{GRU}([\mathbf{w}^{(t-1)}; \mathbf{c}^{(t)}], \mathbf{h}^{(t-1)}) \quad (15)$$

3. **Static Memory Model.** This is a variant of our model in which we do not upgrade the memory dynamically at each time step. Rather, we use the initial memory state as the input to all of the decoder GRU steps.

4. **Dynamic Memory Network (DMN+).** We consider the approach proposed by Xiong et al. (2016), which supersedes Kumar et al. (2016). However, some minor modifications are needed to adapt it to our task. Unlike the bAbI dataset, our task does not involve any question. The presence of a question is imperative in DMN+, as it helps to determine the initial state of the episodic memory module. Thus, we prepend an interrogative phrase such as *"Who is"* or *"What is"* to every entity name. The question module of the DMN+ is hence presented with a question such as *"Who is Roger Federer?"* or *"What is Star Wars?"*. Another difference is in the output module. In DMN+, the final memory state is passed through a softmax layer to generate the answer. Since most answers in the bAbI dataset are unigrams, such an approach suffices. However, as our task is to generate a sequence of words as descriptions, we use a GRU-based decoder sequence model, which at each time step receives the final memory state $\mathbf{m}^{(T)}$ as input to the GRU. We restrict the number of memory update episodes to 3, which is also the preferred number of episodes in the original paper.

3.3 Experimental Setup

For each entity in the WikiFacts10K dataset, there is a corresponding set of facts expressed as factual phrases as defined earlier. Each factual phrase in turn is encoded as a vector by means of the positional encoding scheme described in Section 2.1. Although other variants could be considered, such as LSTMs and GRUs, we apply this standard fact encoding mechanism for our model as well as all our baselines for the sake of uniformity and fair comparison. Another factor that makes the use of a sequence encoder such as LSTMs or GRUs less suitable is that the set of input facts is essentially unordered without any temporal correlation between facts.

We fixed the dimensionality of the fact embeddings and all hidden states to be 100. The vocabulary size is 29K. Our models and all other baselines are trained for a maximum of 25 epochs with an early stopping criterion and a fixed learning rate of 0.001.

To evaluate the quality of the generated descriptions, we rely on the standard BLEU (B-1, B-2, B-3, B-4), ROUGE-L, METEOR and CIDEr metrics, as implemented by Sharma et al. (2017). Of course, we would be remiss not to point out that these metrics are imperfect. In general, they tend

to be conservative in that they only reward generated descriptions that overlap substantially with the ground truth descriptions given in Wikidata. In reality, it may of course be the case that alternative descriptions are equally appropriate. In fact, inspecting the generated descriptions, we found that our method often indeed generates correct alternative descriptions. For instance, Darius Kaiser is described as a *cyclist*, but one could also describe him as a *German bicycle racer*. Despite their shortcomings, the aforementioned metrics have generally been found suitable for comparing supervised systems, in that systems with significantly higher scores tend to fare better at learning to reproduce ground truth captions.

3.4 Results

The results of the experiments are reported in Table 1. Across all metrics, we observe that our model obtains significantly better scores than the alternatives.

A facts-to-seq model exploiting our positional fact encoding performs adequately. With an additional attention mechanism (Facts-to-seq w. Attention), the results are even better. This is on account of the attention mechanism's ability to reconsider the attention distribution at each time step using the current context of the output sequence. The results suggest that this enables the model to more flexibly focus on the most pertinent parts of the input. In this regard, such a model thus resembles our approach. However, there are important differences between this baseline and our model. Our model not only uses the current context of the output sequence, but also memorizes how information of a particular fact has been used thus far, via the dynamic memory module. We conjecture that the dynamic memory module thereby facilitates generating longer description sequences more accurately by better tracking which parts have been attended to, as is empirically corroborated by the comparably higher BLEU scores for longer n-grams.

The analysis of the Static Memory approach amounts to an ablation study, as it only differs from our full model in lacking memory updates. The divergence of scores between the two variants suggests that the dynamic memory indeed is vital for more dynamically attending to the facts by taking into account the current context of the output sequence at each step. Our model needs to dynamically achieve different objectives at different time points. For instance, it may start off looking at several properties to infer a type of the appropriate granularity for the entity (e.g., *village*), while in the following steps it considers a salient property and emits the corresponding named entity for it as well as a suitable preposition (e.g., *in China*).

Finally, the poor results of the DMN+ approach show that a naïve application of a state-of-the-art dynamic memory architecture does not suffice to obtain strong results on this task. Indeed, the DMN+ is even outperformed by our Facts-to-seq baseline. This appears to stem from the inability of the model to properly memorize all pertinent facts in its encoder.

Analysis. In Figure 3, we visualize the attention distribution over facts. We observe how the model shifts its focus to different sorts of properties while generating successive words.

Table 2 provides a representative sample of the generated descriptions and their ground truth counterparts. A manual inspection reveals five distinct patterns. The first case is that of exact matches with the reference descriptions. The second involves examples on which there is a high overlap of words between the ground truth and generated descriptions, but the latter as a whole is incorrect because of semantic drift or other challenges. In some cases, the model may have never seen a word or named entity during training (e.g., *Hypocrisy*), or their frequency is very limited in the training set. While it has been shown that GRUs with an attention mechanism are capable of learning to copy random strings from the input (Gu et al., 2016), we conjecture that a dedicated copy mechanism may help to mitigate this problem, which we will explore in future research. In other cases, the model conflates semantically related concepts, as is evident from examples such as a *film* being described as a *filmmaker* and a *polo player* as a *water polo player*. Next, the third group involves generated descriptions that are more specific than the ground truth, but correct, while, in the fourth group, the generated outputs generalize the descriptions to a certain extent. For example, *American musician and pianist* is generalized as *American musician*, since *musician* is a hypernym of *pianist*. Finally, the last group consists of cases in which our model generated descriptions that are factually accurate and may be deemed appropriate despite diverging from the

Figure 3: An example of attention distribution over the facts while emitting words. The *country of citizenship* property gets the most attention while generating the first word *French* of the left description. For generating the next three words, the fact *occupation* attracts the most attention. Similarly, *instance of* attracts the most attention when generating the sequence *Italian comune*.

Table 2: A representative sample of the generated descriptions and its comparison with the ground truth descriptions.

	Item	Ground Truth Description	Generated Description
Matches	Q20538915	painting by Claude Monet	painting by Claude Monet
	Q10592904	genus of fungi	genus of fungi
	Q669081	municipality in Austria	municipality in Austria
	Q23588047	microbial protein found in Mycobacterium abscessus	microbial protein found in Mycobacterium abscessus
Semantic drift	Q1777131	album by Hypocrisy	album by Mandy Moore
	Q16164685	polo player	water polo player
	Q849834	class of 46 electric locomotives	class of 20 british 0-6-0t locomotives
	Q1434610	1928 film	filmmaker
More specific	Q1865706	footballer	Finnish footballer
	Q19261036	number	natural number
	Q7807066	cricketer	English cricketer
	Q10311160	Brazilian lawyer	Brazilian lawyer and politician
More general	Q149658	main-belt asteroid	asteroid
	Q448330	American musician and pianist	American musician
	Q4801958	2011 Hindi film	Indian film
	Q7815530	South Carolina politician	American politician
Alternative	Q7364988	Dean of York	British academic
	Q1165984	cyclist	German bicycle racer
	Q6179770	recipient of the knight's cross	German general
	Q17660616	singer-songwriter	Canadian musician

reference descriptions to an extent that almost no overlapping words are shared with them. Note that such outputs are heavily penalized by the metrics considered in our evaluation.

4 Related Work

Type Prediction. There has been extensive work on predicting the ontological types of entities in large knowledge graphs (Neelakantan and Chang, 2015; Miao et al., 2016; Kejriwal and Szekely, 2017; Shimaoka et al., 2017), in semi-structured resources such as Wikipedia (Ponzetto and Strube, 2007; de Melo and Weikum, 2010), as well as in text (Del Corro et al., 2015; Yaghoobzadeh and Schütze, 2015; Ren et al., 2016). However, the major shortcoming of these sorts of methods, including those aiming at more fine-grained typing, is that they assume that the set of candidate types is given as input, and the main remaining challenge is to pick the correct one(s). In contrast, our work yields descriptions that often indicate the type of entity, but typically are more natural-sounding and descriptive (e.g. *French Impressionist artist*) than the oftentimes abstract ontological types (such as *human* or *artifact*) chosen by type prediction methods.

A separate, long-running series of work has obtained open vocabulary type predictions for named entities and concepts mentioned in text (Hearst, 1992; Snow et al., 2006), possibly also induc-

ing taxonomies from them (Poon and Domingos, 2010; Velardi et al., 2013; Bansal et al., 2014). However, these methods typically just need to select existing spans of text from the input as the output description.

Text Generation from Structured Data. Research on methods to generate descriptions for entities has remained scant. Lebret et al. (2016) take Wikipedia infobox data as input and train a custom form of neural language model that, conditioned on occurrences of words in the input table, generates biographical sentences as output. However, their system is limited to a single kind of description (biographical sentences) that tend to share a common structure. Wang et al. (2016) focus on the problem of temporal ordering of extracted facts. Biran and McKeown (2017) introduced a template-based description generation framework for creating hybrid concept-to-text and text-to-text generation systems that produce descriptions of RDF entities. Their framework can be tuned for new domains, but does not yield a broad-coverage multi-domain model. Voskarides et al. (2017) first create sentence templates for specific entity relationships, and then, given a new relationship instance, generate a description by selecting the best template and filling the template slots with the appropriate entities from the knowledge graph. Kutlak et al. (2013) generates referring expressions by converting property-value pairs to text using a hand-crafted mapping scheme. Wiseman et al. (2017) considered the related task of mapping tables with numeric basketball statistics to natural language. They investigated an extensive array of current state-of-the-art neural pointer methods but found that template-based models outperform all neural models on this task by a significant margin. However, their method requires specific templates for each domain (for example, basketball games in their case). Applying template-based methods to cross-domain knowledge bases is highly challenging, as this would require too many different templates for different types of entities. Our dataset contains items of from a large number of diverse domains such as humans, books, films, paintings, music albums, genes, proteins, cities, scientific articles, etc., to name but a few.

Chen and Mooney (2008) studied the task of taking representations of observations from a sports simulation (Robocup simulator) as input, e.g. *pass(arg1=purple6, arg2=purple3)*, and generating game commentary. Liang et al. (2009) learned alignments between formal descriptions such as *rainChance(time=26-30,mode=Def)* and natural language weather reports. Mei et al. (2016) used LSTMs for these sorts of generation tasks, via a custom coarse-to-fine architecture that first determines which input parts to focus on.

Much of the aforementioned work essentially involves aligning small snippets in the input to the relevant parts in the training output and then learning to expand such input snippets into full sentences. In contrast, in our task, alignments between parts of the input and the output do not suffice. Instead, describing an entity often also involves considering all available evidence about that entity to infer information about it that is often not immediately given. Rather than verbalizing facts, our method needs a complex attention mechanism to predict an object's general type and consider the information that is most likely to appear salient to humans from across the entire input.

The WebNLG Challenge (Gardent et al., 2017) is another task for generating text from structured data. However, this task requires a textual verbalization of every triple. On the contrary, the task we consider in this work is quite complementary in that a verbalization of all facts one-by-one is not the sought result. Rather, our task requires synthesizing a short description by carefully selecting the most relevant and distinctive facts from the set of all available facts about the entity. Due to these differences, the WebNLG dataset was not suitable for the research question considered by our paper.

Neural Text Summarization. Generating entity descriptions is related to the task of text summarization. Most traditional work in this area was extractive in nature, i.e. it selects the most salient sentences from a given input text and concatenates them to form a shorter summary or presents them differently to the user (Yang et al., 2017). Abstractive summarization goes beyond this in generating new text not necessarily encountered in the input, as is typically necessary in our setting. The surge of sequence-to-sequence modeling of text via LSTMs naturally extends to the task of abstractive summarization by training a model to accept a longer sequence as input and learning to generate a shorter compressed sequence as a summary.

Rush et al. (2015) employed this idea to generate a short headline from the first sentence of a text. Subsequent work investigated the use of

architectures such as pointer-generator networks to better cope with long input texts (See et al., 2017). Recently, Liu et al. (2018) presented a model that generates an entire Wikipedia article via a neural decoder component that performs abstractive summarization of multiple source documents. Our work differs from such previous work in that we do not consider a text sequence as input. Rather, our input are a series of entity relationships or properties, as reflected by our facts-to-sequence baselines in the experiments. Note that our task is in certain respects also more difficult than text summarization. While regular neural summarizers are often able to identify salient spans of text that can be copied to the output, our input is of a substantially different form than the desired output.

Additionally, our goal is to make our method applicable to any entity with factual information that may not have a corresponding Wikipedia-like article available. Indeed, Wikidata currently has 46 million items, whereas the English Wikipedia has only 5.6 million articles. Hence, for the vast majority of items in Wikidata, no corresponding Wikipedia article is available. In such cases, a summarization baseline will not be effective.

Episodic Memory Architectures. A number of neural models have been put forth that possess the ability to interact with a memory component. Recent advances in neural architectures that combine memory components with an attention mechanism exhibit the ability to extract and reason over factual information. A well-known example is the End-To-End Memory Network model by Sukhbaatar et al. (2015), which may make multiple passes over the memory input to facilitate multi-hop reasoning. These have been particularly successful on the bAbI test suite of artificial comprehension tests (Weston et al., 2015), due to their ability to extract and reason over the input.

At the core of the Dynamic Memory Networks (DMN) architecture (Kumar et al., 2016) is an episodic memory module, which is updated at each episode with new information that is required to answer a predefined question. Our approach shares several commonalities with DMNs, as it is also endowed with a dynamic memory of this sort. However, there are also a number of significant differences. First of all, DMN and its improved version DMN+ (Xiong et al., 2016) assume sequential correlations between the sentences and rely on them for reasoning purposes. To this end, DMN+ needs an additional layer of GRUs, which is used to capture sequential correlations among sentences. Our model does not need any such layer, as facts in a knowledge graph do not necessarily possess any sequential interconnections. Additionally, DMNs assume a predefined number of memory episodes, with the final memory state being passed to the answer module. Unlike DMNs, our model uses the dynamic context of the output sequence to update the memory state. The number of memory updates in our model flexibly depends on the length of the generated sequence. DMNs also have an additional question module as input, which guides the memory updates and also the output, while our model does not leverage any such guiding factor. Finally, in DMNs, the output is typically a unigram, whereas our model emits a sequence of words.

5 Conclusion

Short textual descriptions of entities facilitate instantaneous grasping of key information about entities and their types. Generating them from facts in a knowledge graph requires not only mapping the structured fact information to natural language, but also identifying the type of entity and then discerning the most crucial pieces of information for that particular type from the long list of input facts and compressing them down to a highly succinct form. This is very challenging in light of the very heterogeneous kinds of entities in our data.

To this end, we have introduced a novel dynamic memory-based neural architecture that updates its memory at each step to continually reassess the relevance of potential input signals. We have shown that our approach outperforms several competitive baselines. In future work, we hope to explore the potential of this architecture on further kinds of data, including multimodal data (Long et al., 2018), from which one can extract structured signals. Our code and data is freely available.[2]

Acknowledgments

This research is funded in part by ARO grant no. W911NF-17-C-0098 as part of the DARPA SocialSim program.

[2] https://github.com/kingsaint/Open-vocabulary-entity-type-description

References

Mohit Bansal, David Burkett, Gerard de Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1041–1051. http://www.aclweb.org/anthology/P14-1098.

Or Biran and Kathleen McKeown. 2017. Domain-adaptable hybrid generation of RDF entity descriptions. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*. pages 306–315. https://aclanthology.info/papers/I17-1031/i17-1031.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '08, pages 128–135. https://doi.org/10.1145/1390156.1390173.

Gerard de Melo and Gerhard Weikum. 2010. MENTA: Inducing multilingual taxonomies from Wikipedia. In Jimmy Huang, Nick Koudas, Gareth Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM 2010)*. ACM, New York, NY, USA, pages 1099–1108.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. FINET: Context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 868–878. http://aclweb.org/anthology/D15-1103.

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. 2014. Introducing Wikidata to the Linked Data Web. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig A. Knoblock, Denny Vrandečić, Paul T. Groth, Natasha F. Noy, Krzysztof Janowicz, and Carole A. Goble, editors, *Proceedings of the 13th International Semantic Web Conference (ISWC'14)*. Springer, volume 8796 of *LNCS*, pages 50–65.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*. Association for Computational Linguistics, Santiago de Compostela, Spain, pages 124–133. http://www.aclweb.org/anthology/W17-3518.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1631–1640. http://www.aclweb.org/anthology/P16-1154.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*.

Mayank Kejriwal and Pedro Szekely. 2017. Supervised typing of big graphs using semantic embeddings. *CoRR* abs/1703.07805. http://arxiv.org/abs/1703.07805.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, New York, New York, USA, volume 48 of *Proceedings of Machine Learning Research*, pages 1378–1387. http://proceedings.mlr.press/v48/kumar16.html.

Roman Kutlak, Kees van Deemter, and Christopher Stuart Mellish. 2013. Generation of referring expressions in large domains.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Generating text from structured data with application to the biography domain. *CoRR* abs/1603.07771. http://arxiv.org/abs/1603.07771.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '09, pages 91–99. http://dl.acm.org/citation.cfm?id=1687878.1687893.

Peter Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by summarizing long sequences. *CoRR* abs/1801.10198. http://arxiv.org/abs/1801.10198.

Xiang Long, Chuang Gan, and Gerard de Melo. 2018. Video captioning with multi-faceted attention. *Transactions of the Association for Computational Linguistics (TACL)* 6:173–184. https://transacl.org/ojs/index.php/tacl/article/view/1289.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of NAACL*.

Qingliang Miao, Ruiyu Fang, Shuangyong Song, Zhongguang Zheng, Lu Fang, Yao Meng, and Jun Sun. 2016. Automatic identifying entity type in Linked Data. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation, PACLIC 30, Seoul, Korea, October 28 - October 30, 2016*. http://aclweb.org/anthology/Y/Y16/Y16-3009.pdf.

Arvind Neelakantan and Ming-Wei Chang. 2015. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 515–525. http://www.aclweb.org/anthology/N15-1054.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*. AAAI Press, AAAI'07, pages 1440–1445. http://dl.acm.org/citation.cfm?id=1619797.1619876.

Hoifung Poon and Pedro Domingos. 2010. Unsupervised ontology induction from text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '10, pages 296–305. http://dl.acm.org/citation.cfm?id=1858681.1858712.

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *EMNLP*.

Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Braem. 1976. Basic objects in natural categories. *Cognitive Psychology* .

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* .

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 1073–1083. https://doi.org/10.18653/v1/P17-1099.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR* abs/1706.09799. http://arxiv.org/abs/1706.09799.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 1271–1280. http://www.aclweb.org/anthology/E17-1119.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL-44, pages 801–808. https://doi.org/10.3115/1220175.1220276.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 2440–2448. http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Niket Tandon, Gerard de Melo, Abir De, and Gerhard Weikum. 2015. Knowlywood: Mining activity knowledge from Hollywood narratives. In *Proceedings of CIKM 2015*.

Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From Freebase to Wikidata: The great migration. In *World Wide Web Conference*.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics* 39(3):665–707. https://doi.org/10.1162/COLI_a_00146.

Nikos Voskarides, Edgar Meij, and Maarten de Rijke. 2017. Generating descriptions of entity relationships. In *ECIR 2017: 39th European Conference on Information Retrieval*. Springer, LNCS.

Yafang Wang, Zhaochun Ren, Martin Theobald, Maximilian Dylla, and Gerard de Melo. 2016. Summary generation for temporal extractions. In *Proceedings of 27th International Conference on Database and Expert Systems Applications (DEXA 2016)*.

Chris Welty, J. William Murdock, Aditya Kalyanpur, and James Fan. 2012. A comparison of hard filters and soft evidence for answer typing in Watson. In Philippe Cudré-Mauroux, Jeff Heflin,

Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2012*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 243–256.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR* abs/1502.05698. http://arxiv.org/abs/1502.05698.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2253–2263. https://www.aclweb.org/anthology/D17-1239.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. JMLR.org, ICML'16, pages 2397–2406. http://dl.acm.org/citation.cfm?id=3045390.3045643.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 715–725. http://aclweb.org/anthology/D15-1083.

Qian Yang, Yong Cheng, Sen Wang, and Gerard de Melo. 2017. HiText: Text reading with dynamic salience marking. In *Proceedings of WWW 2017 (Digital Learning Track)*. ACM.