A2N: Attending to Neighbors for Knowledge Graph Inference

Trapit Bansal[†*] and **Da-Cheng Juan**[‡] and **Sujith Ravi**[‡] and **Andrew McCallum**[†]
† University of Massachusetts, Amherst
‡ Google Research
{tbansal, mccallum}@cs.umass.edu
{dacheng, sravi}@google.com

Abstract

State-of-the-art models for knowledge graph completion aim at learning a fixed embedding representation of entities in a multi-relational graph which can generalize to infer unseen entity relationships at test time. This can be sub-optimal as it requires memorizing and generalizing to all possible entity relationships using these fixed representations. We thus propose a novel attention-based method to learn query-dependent representation of entities which adaptively combines the relevant graph neighborhood of an entity leading to more accurate KG completion. The proposed method is evaluated on two benchmark datasets for knowledge graph completion, and experimental results show that the proposed model performs competitively or better than existing state-of-the-art, including recent methods for explicit multi-hop reasoning. Qualitative probing offers insight into how the model can reason about facts involving multiple hops in the knowledge graph, through the use of neighborhood attention.

1 Introduction

Knowledge graphs, such as Freebase (Bollacker et al., 2008), contain a wealth of structured knowledge in the form of relationships between entities and are useful for numerous end applications. However, knowledge graphs (KG)—whether automatically constructed or human curated—are incomplete (Banko et al., 2007) and thus automatic methods for KG completion have been an important area of research (Nickel et al., 2016). The task of KG completion requires inferring missing entity relationships from the observed graph and is often formulated as predicting a target entity for a given query of source entity e and relation r, that is, to complete the tuple $(e, r, ?)$.

Most state-of-the-art methods for KG completion learn vector embeddings of entities and relations (Bordes et al., 2013; Toutanova et al., 2015; Dettmers et al., 2017; Trouillon et al., 2016) which are used in conjunction with a (potentially parameterized) scoring function that scores every tuple in the graph. These embeddings are optimized such that the score for observed graph tuples is higher than a random tuple. While these models achieve good performance, they learn a fixed-dimensional embedding for every entity, which necessitates that this embedding must memorize and then be able to generalize to infer all possible relationships for the entity, which may require multiple-hops of reasoning in the KG (Neelakantan et al., 2015; Das et al., 2017).

In contrast, it can be beneficial to compose embeddings from a query-relevant subset of the graph neighborhood of the entity. As a motivating example, consider answering the query *(e, nationality, ?)* for some entity e. Observing the KG neighbor *(e, lived_in, Maui)*, can allow us to project e into the *Maui* region of the embedding space which can lead to a high score for predicting the target *USA* (through an appropriate scoring function), as *Maui* and *USA* were close in embedding space due to other relations between them in KG. Note that here e can have a type that is very different than the type of $Maui$, for example e can be *Oprah Winfrey* in which case it's type would be Actor but using the neighborhood we can still project it to be close to *USA* for the query.

Thus, we propose A2N, an effective model (Section 2) which, conditioned on the query, uses a bi-linear attention on the graph neighborhood of an entity to generate an embedding representation of the entity. This query-specific and neighborhood-informed representation is then used to score target entities for the query. Intuitively, for the example described above, the model

[*] Work done as an intern at Google Research

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4387–4392
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

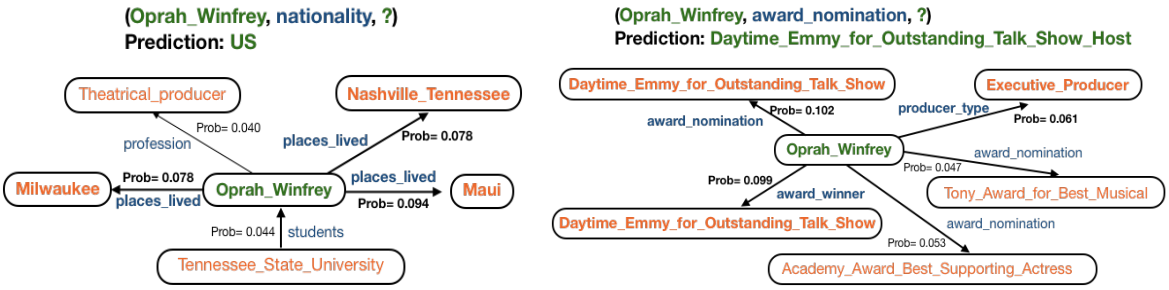

Figure 1: An actual example of how the A2N model generates the answer for two different queries for the same entity *Oprah Winfrey* on the FB15k-237 graph. We show a subset of the top neighbors. Each neighbor is assigned a probability based on the query and the neighbor representations are pooled based on these probabilities to obtain the entity embedding for the source entity. Top 3 neighbors are in bold face.

can score neighbors connected via the *lived_in* relations higher so that the resulting embedding of the entity would be in the *US* region of the embedding space which when scored in conjunction with the query relation *nationality* can yield a high score for the target entity *US*. Fig. 1 shows an actual example how the model scores the graph neighborhood for two different queries on the same node, attending to a different relevant subset of the neighborhood for each query.

On two standard benchmark datasets for KG completion (Dettmers et al., 2017) – FB15k-237 (Toutanova et al., 2015) and WN18RR – we show (Section 3.2) that the model performs competitively or better than existing state-of-the-art models. Qualitative analysis (Fig. 1, Section 3.2) shows that the model indeed assigns higher scores to relevant neighbors based on the query and provides insight into how the model answers queries requiring multiple hops.

2 Methodology

Problem Formulation and Notation: Let $[X]$ represent the integer set $\{1, \ldots, X\}$. We are given a KG, $\mathcal{G} := \{(s, r, t)\}$ where each tuple consists of a *source entity* $s \in [V_e]$, a *relation* $r \in [V_r]$ and a *target entity* $t \in [V_e]$, with V_r being the number of relations and V_e the number of entities in the graph. The objective is to predict the target entity for a given query of source entity and relation, $q := (s, r, ?)$ – such that the predicted tuple doesn't already exist in $\mathcal{G}$.

Entities, e, and relations, r, are represented as k-dimensional embeddings, $\tilde{e}$ and $\tilde{r}$. Most embedding-based methods for KG completion work by defining a scoring function f for every possible tuple in the KG. For example, DistMult (Yang et al., 2014) uses the following score:

$$f(s, r, t) = \tilde{s}^T Diag(\tilde{r}) \tilde{t} \quad (1)$$

where $Diag(\tilde{r})$ is a $k \times k$ diagonal matrix with $\tilde{r}$ in its diagonal. There are other functions proposed in the literature (Bordes et al., 2013; Dettmers et al., 2017; Trouillon et al., 2016). We used the DistMult scoring function in our experiments for its simplicity and good performance when tuned properly (Kadlec et al., 2017), though the model can be combined with any other scoring function.

2.1 A2N Model

We now describe our graph-attention model. Consider the neighborhood of an entity s to be $N_s = \{(r_i, e_i) | (s, r_i, e_i) \in \mathcal{G}\}$. We associate each graph entity e with an initial embedding $\tilde{e}^0$, and each relation r with an embedding $\tilde{r}$. We first encode every neighbor into an embedding. The embedding of a neighbor $(r_i, e_i) \in N_s$ of entity s, is obtained by concatenating the initial embeddings and projecting using a linear transform. The model then attends to each element of N_s, assigning it a probability for its relevance in answering the query and generates the query-dependent embedding of the entity s by aggregating the neighbor embeddings weighted by their relevance. Concretely, given a query $(s, r, ?)$, we assign each neighbor $n_i \in N_s$ a scalar attention score a_i which is then normalized over all neighbors to obtain the probabilities p_i. The neighbor embeddings are then then aggregated with p_i as weights to generate new source embedding $\hat{s}$. This is concatenated with the initial source embedding and projected to K dimension to obtain the final source embedding $\tilde{s}$:

$$\tilde{n}_i = W_n[\tilde{r}_i; \tilde{e}_i^0]$$
$$a_i = f(s, r, n_i) = (\tilde{s}^0)^T Diag(\tilde{r}) \tilde{n}_i \quad (2)$$

$$p_i = \frac{\exp(a_i)}{\sum_{j \leq |N_s|} \exp(a_j)}$$
$$\hat{s} = \sum_{i \leq |N_s|} p_i \tilde{n}_i$$
$$\tilde{s} = W_s[\hat{s}; \tilde{s}^0] \quad (3)$$

where $W_n \in \mathbb{R}^{K \times 2K}$, $W_s \in \mathbb{R}^{K \times 2K}$ are projection matrices. We use this attention-based embedding for the source entity $\tilde{s}$ along with the query relation embedding $\tilde{r}$ and the base embedding for a potential target entity $\tilde{t}^0$ in the DistMult scoring function Eq(1) to generate a score for the tuple (s, r, t). This is done for all possible entities $t \in [V_e]$ to obtain a ranked list of potential target entities for the query.

We use DistMult function for the attention scoring in Eq(2) as it allows the model to learn to project the neighbors in the same space as the target entities, so as to give high scores to correct targets when the resulting embedding is scored again using the DistMult score Eq(1).

Training: The model is randomly initialized and all embeddings and projection parameters are trained by taking a tuple from the graph $(s, r, t) \in G$, hiding the target entity t and randomly sampling negative entities, $t^- = \{e|(s, r, e) \notin G\}$. The scores for the positive and each negative tuple are passed through a softmax to compute the likelihood of predicting the correct target. The same process is repeated for predicting the source entity given (r, t). We also augment the graph by adding an inverse relation for every graph relation which improves training by increasing the possible neighborhood elements for the model to attend.

3 Experiments

3.1 Experimental Setup

Datasets: Following Dettmers et al. (2017), we evaluate the model on two standard benchmark datasets for KG completion, FB15k-237 (Toutanova et al., 2015) and WN18RR (Dettmers et al., 2017).

Evaluation Protocol: We followed the evaluation protocol of Dettmers et al. (2017). Each test tuple (s, r, t) is converted into two queries: target query $(s, r, ?)$ and source query $(?, r, t)$. For every query, the correct entity is ranked among all KG entities excluding the set of other true entities for the query observed in either train/dev/test set for the same query. See Kadlec et al. (2017) for more details. We then report the Mean Reciprocal Rank (MRR) of the correct entity, that is the average of the reciprocal rank of the correct entity, and the Hits@N, that is the accuracy in the top N predictions. Experimental details, including all hyper-parameters, are in Appendix A.

3.2 Results

Results are summarized in Tables 1 and 2. We compare with many state-of-the-art methods for KG completion. Note that we did not fine-tune separate models for target-only and source-and-target prediction, but instead trained a model for source-and-target prediction and used it for both evaluations. In Table 1, we evaluate performance on target-only prediction. The baseline results on target-only prediction are taken from Das et al. (2017), who finetuned all the models for this task. We find that the proposed A2N model performs significantly better than all baseline models for target-only prediction. Interestingly, the model also performs significantly better than MINERVA, a model which uses RL to search for explicit paths for multi-hop reasoning (Das et al., 2017).

In Table 2, we evaluate on both source and target prediction[1]. The remaining baseline results are reproduced from the respective papers. We find that on WN18RR the model performs better than all baselines, except on Hits@10 metric where it is competitive with ConvE. On FB15k-237, the model performs competitively to ConvE and better than all the other models. Among existing baselines, we found ConvE to be the best competitor to our model. Note that in general all models perform better on target-only (Table 1) as compared to both source and target prediction. This is due to more ambiguous and one-to-many queries when predicting source entity (Das et al., 2017), for example *(?, nationality, US)*. For such generic source-prediction queries we expect attention to be of limited use.

Qualitative Results: Fig. 1 shows how the model attends to different subsets of neighbors for the same graph entity for different queries. This example also demonstrates how the model can reason about multiple-hops of facts. Using neighbors such as *places_lived*, the entity is first projected into a relevant subspace of the embedding space and then when scored with the target entity *US* leads to a high DistMult score for the relation *na-*

[1]We tuned our own DistMult implementation and obtained better results

	FB15k-237				WN18RR			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
DistMult	0.370	0.568	0.417	0.275	0.43	0.48	0.44	0.41
ComplEx	0.394	0.572	0.434	0.303	0.42	0.48	0.43	0.38
ConvE	0.410	0.600	0.457	0.313	0.44	0.52	0.45	0.40
MINERVA	0.293	0.456	0.329	0.217	0.45	0.51	0.46	0.41
A2N	**0.422**	**0.608**	**0.464**	**0.328**	**0.49**	**0.55**	**0.50**	**0.45**

Table 1: Results for target-only prediction of various models. A2N performs significantly better.

	FB15k-237				WN18RR			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
DistMult	0.278	0.444	0.304	0.196	0.43	0.49	0.44	0.39
ComplEx	0.247	0.428	0.275	0.158	0.44	0.51	0.46	0.41
R-GCN	0.249	0.417	0.264	0.151	–	–	–	–
ConvE	**0.325**	**0.501**	**0.356**	**0.237**	0.43	**0.52**	0.44	0.40
A2N	0.317	0.486	0.348	0.232	**0.45**	0.51	**0.46**	**0.42**

Table 2: Results for both source and target prediction of various models. A2N performs better or competitively to most state-of-the-art models, specially on top prediction (Hits@1).

tionality. Here the model implicitly reasoned over a two-hop fact, first about *places_lived* and the second about the *country* of those places. More examples of attention are provided in Fig. 2, refer to the Appendix B for more qualitative analysis.

4 Related Work

KG completion is an important research area, with several embedding-based models proposed, such as TransE which scores translations of entities in embedding space (Bordes et al., 2013), DistMult (Toutanova et al., 2015), ComplEx which is an extension to complex space (Trouillon et al., 2016), ConvE which uses 2D convolution layers (Dettmers et al., 2017) as well as recent tensor decomposition methods (Lacroix et al., 2018). Refer to Nickel et al. (2016) for a more comprehensive review. Recently, Das et al. (2017); Xiong et al. (2017) proposed reinforcement learning methods which find paths in KG. We compared with MINERVA (Das et al., 2017), a recent method, and found A2N to perform favorably. Graph Convolution Networks (Kipf and Welling, 2016; Schlichtkrull et al., 2017) and Graph attention networks (Veličković et al., 2017) also learn neighborhood based representations of nodes. However, they do not learn a query-dependent composition of the neighborhood which is sub-optimal as also seen our in experiments and noted previously (Dettmers et al., 2017). They are also computationally expensive. Nguyen et al. (2016) proposed a neighborhood mixture model which is closely related. However, their proposed model learns a fixed mixture over neighbors as opposed to learning an adaptive mixture based on the query, and requires storing an embedding parameter for every entity-relation pair which can be prohibitively large, potentially $O(V_e \times V_r)$ whereas our model only requires $O(V_e + V_r)$. Moreover, their model cannot generalize to unseen entity-relation pairs and new neighbors of an entity even when the entity and relation for the pair was observed with other relations or entities. Our work is also related to Memory Network models, often used for question-answering (Kumar et al., 2016; Miller et al., 2016; Bansal et al., 2017). To the best of our knowledge, this is the first work utilizing attention to learn query-dependent entity embeddings from the entity neighborhoods.

5 Conclusion

We proposed A2N, an attention-based model for learning query-dependent entity embeddings based on graph neighborhood. The model performs favorably when compared with state-of-the-art models for KG completion. The model has attractive properties as it is interpretable and its number of parameters do not depend on the size of entity neighborhoods. Future research will look into applying such methods to reason jointly about text and KG, by attending to textual mentions of entities in addition to graph (Verga et al., 2016).

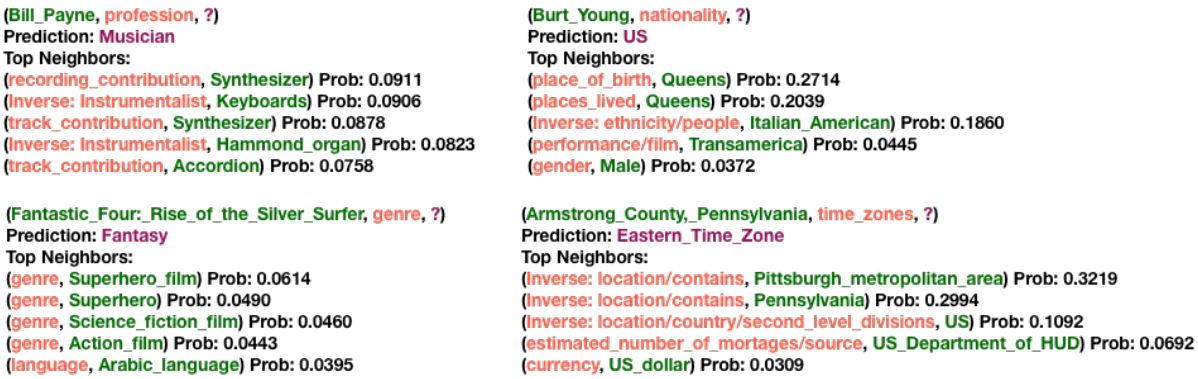

Figure 2: Example of queries, their top prediction and the set of top 5 attention neighbors as well as their attention probabilities for the A2N model.

Acknowledgements

We would like to thank the Expander team from Google Research for helpful feedback.

References

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.

Trapit Bansal, Arvind Neelakantan, and Andrew McCallum. 2017. RelNet: End-to-end modeling of entities & relations. *arXiv preprint arXiv:1706.07179*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2017. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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 *International Conference on Machine Learning*, pages 1378–1387.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. *arXiv preprint arXiv:1806.07297*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base inference. In *2015 aaai spring symposium series*.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Neighborhood mixture model for knowledge base completion. *arXiv preprint arXiv:1606.06461*.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Patrick Verga, Arvind Neelakantan, and Andrew McCallum. 2016. Generalizing to unseen entities and entity pairs with row-less universal schema. *arXiv preprint arXiv:1606.05804*.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

A Experimental Details

We found hyperparameters by selecting the set which performed best on the validation sets according to Hits@10. We evaluated $k \in \{128, 256, 512\}$, number of negative samples $n^- \in \{500, 1000, 2000\}$, batch-size $b \in \{256, 512, 1024, 2048\}$ and chose $k = 512$, $n^- = 2000$ and $b = 1024$. We used Adam (Kingma and Ba, 2014) with a fixed learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. Gradients were clipped to a maximum norm of 10. We capped the maximum number of neighbors of an entity to 500, randomly sub-sampling the set of neighbors when required. We used dropout on all embeddings, and after the projection matrices W_n and W_s. We used the same fixed value of dropout probability d everywhere which we tuned in $\{0.4, 0.3, 0.2, 0.1\}$ and chose $d = 0.3$ as the value on both datasets.

B Qualitative Results

Fig. 2 shows how the model attends to relevant neighbor subsets for different queries. Consider, for example, the query about the profession of *Bill Payne*. Here the model attends to his *recording contribution* as a synthesizer and that he is an *instrumentalist* for Keyboards to infer that he is a musician. On the other hand, for queries like *nationality*, the model attends to neighbors like *place of birth* (see query for *Burt Young*) and *places lived*. For a query about *time zone*, the model attends to the state and metropolitan area containing the location to infer the time-zone. Note that all of these queries requires reasoning over multiple facts and model achieves this by (1) explicitly selecting a subset of neighbors of the entity to project to an appropriate neighborhood in the embedding space, and then (2) selecting the entity with the largest score given by DistMult for the query relation.

Entity	Top 5 Neighbors
Two and a Half Men	Murphy Brown, How I Met Your Mother, The Big Bang Theory, The Larry Sanders Show, Glee
Flute	Saxophone, Fiddle, Violin, Electric Piano, Percussion Instrument
Space Rock	Progressive Metal, Noise Pop, Progressive Rock, Garage Rock, Free Jazz
Madagascar Escape 2 Africa	Shrek Forever After, The Sponge Bob Squarepants, Madagascar (2005), The Prince of Egypt, The Adventures of Tintin
University of Oxford	University of Cambridge, University of Glasgow, University College Oxford, University of Sussex, University of California Berkley
Edinburgh	Aberdeen, Glasgow, Dumfries and Galloway, Dundee, Fife

Table 3: Top 5 neighbors of entities based on cosine similarity.

We found nearest neighbors of entities based on the initial embeddings of entities before attention. These entities should ideally cluster into regions which participate in similar relations as that would benefit attention by allowing entities to be projected into the appropriate region in the embedding space. Table 3 shows the nearest neighbors for some entities. For sitcom TV shows like *Two and a Half Men*, we find other sitcoms like *How I Met Your Mother* in neighbors, for *University of Oxford* we find other universities like *University of Cambridge* as top neighbors, for cities like *Edinburgh* we find other cities in the same country like *Aberdeen* and *Glasgow* as top neighbors. Overall, we found the nearest neighbors to be functionally related which would benefit attention.