Information Systems Frontiers (2020) 22:1095–1111
https://doi.org/10.1007/s10796-020-10035-2

An Overview of Utilizing Knowledge Bases in Neural Networks for Question Answering

Sabin Kafle[1] · **Nisansa de Silva**[1] · **Dejing Dou**[1]

Published online: 29 July 2020

Abstract

Question Answering (QA) requires understanding of queries expressed in natural languages and identification of relevant information content to provide an answer. For closed-world QAs, information access is obtained by means of either context texts, or a Knowledge Base (KB), or both. KBs are human-generated schematic representations of world knowledge. The representational ability of neural networks to generalize world information makes it an important component of current QA research. In this paper, we study the neural networks and QA systems in the context of KBs. Specifically, we focus on surveying methods for KB embedding, how such embeddings are integrated into the neural networks, and the role such embeddings play in improving performance across different question-answering problems. Our study of multiple question answering methods finds that the neural networks are able to produce state-of-art results in different question answering domains, and inclusion of additional information via KB embeddings further improve the performance of such approaches. Further progress in QA can be improved by incorporating more powerful representations of KBs.

Keywords Knowledge base · Question answering · Neural networks

1 Introduction

Neural Question Answering (NQA) has led to significant interest in question answering, especially due to the ability of modeling to incorporate multimodal information sources. To serve as a question-answering system, a typical neural network is capable of: leveraging text information via word or character embeddings (Mikolov et al. 2013); image representation (Wu et al. 2017) via pretrained representations; textual information using unsupervised large-scale language models (Devlin et al. 2019; Radford et al. 2019; Howard and Ruder 2018); and/or KBs using embedding methods similar to word embeddings (Bordes et al. 2013). NQAs systems largely follow a three-stage process, comprised of (a) information retrieval based on the question understanding; (b) answer extraction to generate an answer; and, optionally, (c) a ranking module, to rank the answers (Kratzwald et al. 2019).

Knowledge Graphs (KGs) are the simpler representational form of Knowledge Bases (KBs), expressed in the form of triples of - *entity, relation, entity* -. Unlike KBs which represent a richer hierarchy and structure symbolic to the real-world model, KGs are much less constrained. The simpler representations of KGs have given rise to methods for the representation learning of entities and relations present in a KG. This is in line with advances in embedding methods for multimodal data representation. Most KBs are written in formats (e.g., OWL Antoniou and Van Harmelen 2004), which makes them accessible via query languages such as SPARQL (Seaborne and Prud'hommeaux 2006). This itself is a significant research area and contributes to reasoner systems such as HermiT (Shearer et al. 2008), which can be used to generate an answer from large knowledge graphs based on SPARQL query formulation. While KBs, which are often represented in structured format, are challenging to integrate into the neural network paradigms, KG embeddings are significantly easier to integrate into the existing systems. This leads to a multitude of

✉ Sabin Kafle
skafle@cs.uoregon.edu

Nisansa de Silva
nisansa@cs.uoregon.edu

Dejing Dou
dou@cs.uoregon.edu

[1] Department of Computer and Information Science, University of Oregon, Eugene, Oregon, USA

applications including factoid question answering, visual question answering, reading comprehension, and open-world question answering, all using KGs as an auxiliary data source for improved performance. A KB is also interchangeably called KG as in Ehrlinger and Wöß (2016).

Several KBs (and their triple-based variant KGs) are readily available, with huge amount of information and facts structured within. Some widely used KBs include Freebase (Bollacker et al. 2008), DBPedia (Auer et al. 2007), YAGO (Suchanek et al. 2007), Gene Ontology (Ashburner et al. 2000), Wordnet (Miller 1995), ConceptNet (Liu and Singh 2004), and Google Knowledge Graph (Singhal 2012). Semantic parsing (Berant et al. 2013) approach to the factoid question answering parse a natural language question into a structured query, which is executed into KBs. A major limitation of a KG is its completeness - no KB exists with all the world's information content incorporated into it. NELL (Mitchell et al. 2018; Carlson et al. 2010) is an example system incorporating semi-automatic KBs, which are reliable in effective context understanding and information-extracting frameworks.

In this survey, we study neural question-answering methods applied to a wide range of question-answering problems including factoid question-answering, visual question-answering, and reading comprehension. We primarily explore the usability and contribution of KGs to neural question-answering. While several methods have been proposed to embed KBs, their usage is rather limited due to the current representation limitations of embedding methods along with the lack of incorporation of improvements in embeddings methods into question answering problems due to the differences in research area. Moreover, we explore neural network based methods for question answering, how knowledge bases are incorporated into neural networks, and what is the state of performance of such methods in comparison to other existing methods.

Specifically, we study the benefit of incorporating KBs as a source of knowledge in the context question answering using neural networks. The recent surge of incorporating large scale pretrained language models (Devlin et al. 2019) with billions of parameters have lead to state-of-art performance across multiple question answering datasets, even obtaining human-level performance across some datasets (e.g., Rajpurkar et al. 2016). The usage of pretrained language models as global knowledge into the neural networks can be considered a form of an unstructured KB. We hypothesize a more precise filtering of knowledge obtained from KBs should be a viable (and also more energy-efficient) alternative to the current trend of the incorporating large-scale language models, especially in the context of QAs where the knowledge incorporation is more straight-forward in comparison to other areas requiring more subtle and hard to quantify knowledge (e.g.; text generation). We explore key differences in the KB embeddings methodology in comparison to other pretraining methods, and also understand differences between the neural networks based on non-KB context as some form of "general knowledge" and the neural architectures incorporating KBs.

Some of our findings can be summarized as:

1. KB embedding is the primary method for incorporating KBs into the neural architecture.
2. The performance improvements across multitude of question answering tasks on introduction of large pretrained language models hints towards the inefficient nature of knowledge compression by neural networks and the need to have larger-sized models to obtain better performance.
3. While there is a trend towards incorporating more parameters into the language models, KB methods are more scalable in comparison and are much simpler and efficient. This difference in focus is likely to be a key reason behind KB incorporation being not as powerful as pretrained language models, especially in the context of larger-sized pretrained models obtaining significant improvement over smaller-sized models.
4. We also hypothesize that a KB-embedding representation with greater number of parameters or even the incorporation of more KB-suitable embeddings (e.g.; hierarchical) into the neural networks could result in more improvements.

This journal paper is an extensions of our conference paper on the same subject (Kafle et al. 2019). In addition to covering everything covered by the conference paper, this journal paper:

- Evaluates newer KB embedding methods including quaternion-based methods, which are the new state-of-art, along with multiple other KB embedding methods.
- Adds analysis of KB embeddings in comparison to symbolic representations, the need for improvements and the current state.
- Includes experimental evaluation for KB embeddings, along with data-set introduction and result from the current highly relevant methods.
- Adds question answering using language models (which are the state of art methods for almost all QA except for KB-based QA datasets) since language models have been shown to be a somewhat noisy version of a knowledge base.
- Expands the KB usage methodologies into three distinct frameworks - translation-based, tensor factorization, and complex-hyperplanes for better categorization of methodology.

The rest of the paper is organized as follow. We survey the KG embedding methods after briefly defining the Knowledge Base (Section 2), which are crucial for incorporating KGs into neural question answering systems, followed by a brief survey of question answering specific neural architectures (Section 3). We then dive into factoid question answering problem and the role of KBs in those methods (Section 4), followed by the discussion of attention-based question answering methodology (Section 5) and the conclusion.

2 Knowledge Base

A Knowledge Base (KB) comprises of a structured database with a schema, such as an ontology, describing entities, relations, and attributes, which form the foundation of metadata comprising the structural information (Krótkiewicz et al. 2018). Facts are then added to the KB in accordance with the metadata description, forming the entirety of a KB. Facts are often added as a triples, with either an *isA* relation to describe the instance type of a fact, or with any other relation described in the KB itself. A KB can also be represented as a graph of facts, with entities representing the nodes of the graph, and with the relationships among entities being described by edges. For the rest of the survey, we treat Knowledge Base and Knowledge Graph as the same entity. The reason for such treatment is the simple possibility of potential transformation of KB into graphs by transforming all metadata into the triple form. Also, the existing neural network literature does not draw any distinction between the two terms and often uses them interchangeably.

An ontology is a collection of definitions which model a domain using classes, attributes, and relationships (Gruber 2009). The collection of facts, their attributes, and the relationships containing the discourse of a particular domain, together with its ontology, constitutes a KB. A KB is also a type of Knowledge Graph (KG); and thus, a KB is richly structured, based on its ontology.

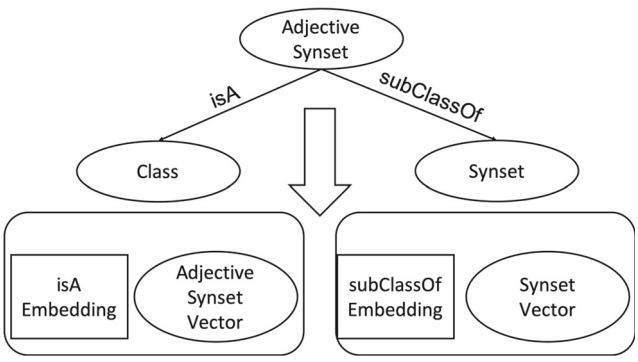

Fig. 1 An example of an ontology embedding model

A KB can be simplistically interpreted as a database system with the schema analogous to ontology, and its *tuples* can be considered *facts*. A KB, though, is capable of incorporating a much richer set of information, such as logical relationships among facts, and can also be inferred using a formal logic reasoner, both for inference and validation purposes. KBs are specifically useful for representing a domain that involves a rich set of relationships among different classes (Chandrasekaran et al. 1999), e.g., Word Net (Miller 1995), UMLS meta thesaurus (Bodenreider 2004).

Knowledge Graphs are typically stored as directed graphs of multi-relational data, whose nodes correspond to entities, and whose edges correspond to relations among them. KBs are represented as a triplet of form (h, l, t) or *(head, label, tail)*, which indicates that there exists a relationship of name $label$ between the entities $head$ and $tail$. The most widely used Knowledge Base is Freebase (Bollacker et al. 2008). It is a structured KB in which entities are connected by predefined predicates (a.k.a relations). All predicates are directional, connecting from subject to object. A triple *(subject, predicate, object)* denoted by (h, p, t) describes a fact; e.g; *(Nepal, capital, Kathmandu)* refers to the fact that *Kathmandu* is the capital of *Nepal*. The usage of knowledge graphs is limited by two issues - completeness (Socher et al. 2013; West et al. 2014) and compatibility. The issue of completeness arises from the fact that no KB can ever be exhaustively completed without any form of conflict simply due to the vast amount of world knowledge that exist along with the very large magnitude of relationships between the entities that represent that knowledge. This inadequacy can lead to error in a query-based system, which completely relies on KBs. Another challenge in usage of KBs lies in its compatibility. Each KB has their own design decisions, and thus, even for the same concepts and relations, different naming conventions are preferred, which presents a challenge in applying more than one KB to a problem. However, application of more than one KB could potentially decrease the incompleteness of KBs (Bordes et al. 2013). A common solution is preferred to both problems: embedding of knowledge bases (Fig. 1).

The conversion of Knowledge Bases (KBs) entities and relations into a numeric structure, whose geometry can then be considered a partial representation of the structure of ontological constraints defined within the KB itself. Conversion of entities and relations into numeric vectors present an interesting perspective into the KB itself, with multiple KBs being potentially used jointly due to their similarity in the structure as opposed to the naming and terminologies consistencies which varies across different Knowledge Bases.

2.1 Knowledge Base Embedding

The general intuition of KB embedding methods is to learn connections and existing patterns from the KB, which can then either be used to extract further patterns using link predictions (Bordes et al. 2013), to verify the truth of a KB triple using either a ranked (Bordes et al. 2013) or probabilistic (Nickel and Tresp 2013) likelihood metric for a fact triple or used in downstream tasks as an extremely compact representation of the global knowledge of KB. In general, the relations in KB are of the form - **symmetric**, **anti-symmetric**, **inversion** and **composition**. KB embedding methods aim to infer the relations using either implicit or explicit modeling of one or many forms of KB relations (Sun et al. 2019). *Symmetric* relations are valid even with the replacement of head with tail entity, while *anti-symmetric* relations are not. *Inverse* relations are conjugate of one another, while *composition* refers to a relation defined as a path walk over multiple relations. We summarize some of the more popular approaches and their objectives and relation factorization in Table 1.

An embedding is a transformation of an entity into a set of numbers (called vectors), which then provides a geometrical interpretation of the entity in the n-dimensional space, where n is the count of numbers used to represent an entity. The vector space thus formulated has a distance metric which can be used to compute distances between different entities and relations via geometrical interpretation. Closely related entities are often placed closer to one another. For example, embeddings of $king$ and $queen$ are often close to each other.

2.2 General Embedding Framework

For E entities and R relations where G denotes the knowledge graph consisting of a set of triples (h, r, t) such that $h, t \in E$ and $r \in R$. The embedding model defines a score function $f(h, r, t)$ for each triple, which is the score of its implausibility. The objective of embedding models is to choose f such that score of a plausible triple (h, r, t) is smaller than score of an implausible one (h', r', t'). The model parameters are learned by minimizing a margin-based objective function:

$$\mathcal{L} = \sum_{\substack{(h,r,t)\in\mathcal{G} \\ (h',r,t')\in\mathcal{G}'_{(h,r,t)}}} [\gamma - f(h,r,t) + f(h',r,t')]_+$$

where $[x]_+ = max(0, x)$, and γ is the margin hyper-parameter. $\mathcal{G}'$ is the set of incorrect triples generated by corrupting the correct triple $(h, r, t) \in G$. The above formulation is exact for margin or translation based embedding methods, while the tensor factorization methods (Nickel and Tresp 2013) can be trained similarly by either

Table 1 Distance functions $f_r(\mathbf{h}, \mathbf{t})$ of knowledge graph embedding models, where $\langle\cdot\rangle$ is the generalized dot product, $\circ$ is the Hadamard product, $\otimes$ is circular correlation, σ is activation function and $*$ is 2D convolution

Embedding Methods	Distance Function		Properties
SE (Bordes et al. 2011)	$\lVert \boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t} \rVert$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$, $\boldsymbol{W}_{r,\cdot} \in \mathbb{R}^{k\times k}$	Not Applicable
TransE (Bordes et al. 2013)	$\lVert \mathbf{h} + \mathbf{r} - \mathbf{t} \rVert$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$	Asymmetry, Inversion, Composition
DistMult (Yang et al. 2015)	$-\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$	Symmetry
ComplEx (Trouillon et al. 2016)	$-\Re(\langle \mathbf{r}, \mathbf{h}, \bar{\mathbf{t}} \rangle)$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$	Symmetry, Asymmetry, Inversion
HolE (Nickel et al. 2016)	$-\langle \mathbf{r}, \mathbf{h} \otimes \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$	Symmetry, Asymmetry, Inversion
ConvE (Dettmers et al. 2018)	$-\langle \sigma(\mathrm{vec}(\sigma([\bar{\mathbf{r}}, \bar{\mathbf{h}}] * \boldsymbol{\Omega}))\boldsymbol{W}), \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$	Not Applicable
RotatE (Sun et al. 2019), QuatE (Zhang et al. 2019)	$\lVert \mathbf{h} \circ \mathbf{r} - \mathbf{t} \rVert$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$, $\lvert r_i \rvert = 1$	Symmetry, Asymmetry, Inversion, Composition

$\bar{\cdot}$ is conjugacy relation of complex vectors, and vector reshaping for the ConvE model. ComplEx and HolE methods are equivalent. Distance-based methods are a complementary-but-similar view of similarity-based methods. $\mathbb{C}$ represents complex space and $\mathbb{R}$ represents Euclidean space

Bold signifies vector and matrices as opposed to singular values

inversing the score or using non-margin training paradigms (e.g. logistic regression). Due to the size of knowledge base (or the number of triples, along with negative triples), the optimization algorithm is characterized by gradient-based techniques.

2.2.1 Mathematical Model

The pioneer work in translation-based embedding models is TransE (Bordes et al. 2013). It assumes all relations and entities can be represented by vectors of uniform size. One issue with the TransE model lies in its inability to differentiate among different relation mappings, such as *one-to-one*, *many-to-one*, and *one-to-many*, which makes the model unsuitable for representing such relations. TransH (Wang et al. 2014) treats each relation to be on a different plane. Figure 2 shows the geometrical contrast between TransH and TransE. Other translation methods, TransD (Ji et al. 2015) and TransX (Lin et al. 2015b), consider diversity of both entity and relation.

In addition, there are several tensor factorization methods for relational learning that generate embeddings for KBs (Nickel et al. 2011, 2012, 2013, Krompaß et al. 2015; Socher et al. 2013). Bayesian Clustering methods have also been successfully applied to embed a KB (Sutskever et al. 2009). Distance-based embedding methods (Bordes et al. 2013; Wang et al. 2014; Guu et al. 2015; Nguyen et al. 2016a; 2016b) have simpler frameworks, making them preferable for usage in underlying applications.

The neural tensor model (Socher et al. 2013) uses bilinear tensor operator to represent each relation. A bilinear score function without any non-linearity to learn embeddings is used by Yang et al. (2015). Quadratic forms are used to model entities and relations in He et al. (2015), Trouillon et al. (2016), and García-Durán et al. (2016). Such methods are similar due to the three-way interactions between relation, head and tail entities during the score computation. ProjE (Shi and Weninger 2017) uses diagonal matrix and linear interaction to combine entity and relations. A circular correlation operation while learning embedding which can be interpreted as compression of tensor product is used in the approach proposed by Nickel et al. (2016).

Additionally, relation paths between entities in Knowledge Graphs provide richer context information, which enables learning more structured embeddings (Luo et al. 2015; García-Durán et al. 2015; Guu et al. 2015; Liang and Forbus 2015; Lin et al. 2015a; Toutanova et al. 2016; Nguyen et al. 2016a). Path queries, to obtain a relational transformation, which is then integrated into a translation model, such as TransE, are used by Guu et al. (2015). The approach in Lin et al. (2015a) extends the TransE method by the additional objective of learning scoring from a different relation path representation, which is a summation over all relation paths that are termed reliable. Toutanova et al. (2016) proposed a dynamic algorithm to enable efficient incorporation of relation paths of bounded length in compositional path models. The authors of Neelakantan et al. (2015) propose a KB completion method using RNNs, which are able to infer multi-hop relationships. An external text corpus for correlating KBs with text is used by Wang and Li (2016).

We discuss three major mathematical formulation in more detail. Let us consider a Knowledge Graph (KG) consisting of components `(subject, predicate, object)`.

2.2.2 Tensor Factorization

RESCAL (Nickel et al. 2011) represents a KG as using a three-way tensor $\mathcal{X}$ with a tensor entry $\mathcal{X}_{ijk} = 1$ when there exists a relation `(i-th entity, k-th predicate, j-th relation)`. For non-existing and unknown relations, the entry is set to `zero`. This method factorizes each relation slice of the tensor as

$$\mathcal{X}_k \approx A R_k A^T, \text{for } k = 1, .., m \tag{1}$$

where A is $n \times r$ matrix containing latent component representation of entities, and R_k is an asymmetric $r \times r$ matrix that models the interactions of the latent components in the k-th predicate. The factor matrices A and R_k

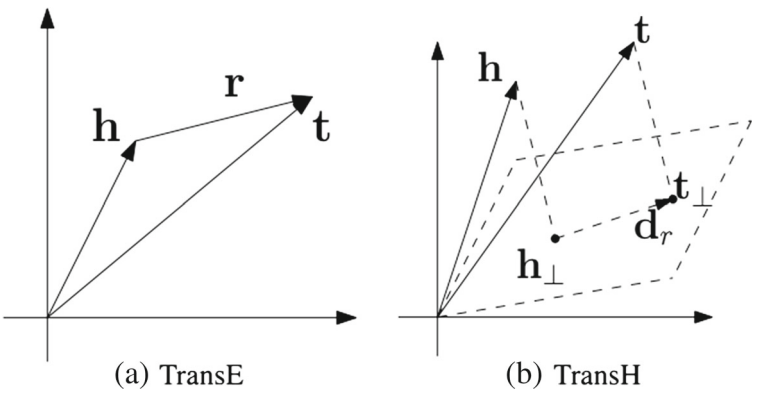

Fig. 2 Geometrical modeling of *TransE* (Bordes et al. 2013) and *TransH* (Wang et al. 2014). TransE translated head entity to tail entity using relation as a vector, while TransH projects the entity embeddings into a relation plane where the actual translation is performed. Such geometric innovations are often the defining factors in improving KB embedding benchmarks

can be computed by solving the regularized minimization formulation. The asymmetry of R_k takes into account whether the latent component occurs as a subject or an object. Nickel et al. (2011) further explore how RESCAL is related to other tensor factorization methods of rank-r DEDICOM and Tucker3 (Balazevic et al. 2019b).

For similarly related bilinear models (e.g., (Yang et al. 2015)), the input for a KG embedding consists of relation triplets of fort (e_1, r, e_2) with e_1 being the `subject` and entity e_2 the `object` that are in a certain `relation` r. Denoting the $\mathbf{x}_{e_i}$ as the input for entity e_i and $\mathbf{W}$ as the first neural network parameter, the scoring function for the triplet (e_1, r, e_2) can be written as

$$S_{(e_1,r,e_2)} = G_r(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) \tag{2}$$

where $\mathbf{y}_{e_1} = f(\mathbf{W}\mathbf{x}_{e_1})$ and $\mathbf{y}_{e_2} = f(\mathbf{W}\mathbf{x}_{e_2})$. The scoring function S available is characterized by either a basic-linear transformation of form

$$g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{A}_r^T \begin{pmatrix} \mathbf{y}_{e_1} \\ \mathbf{y}_{e_2} \end{pmatrix}$$

and a bilinear transformation of the form

$$g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{y}_{e_1}^T \mathbf{B}_r \mathbf{y}_{e_2}$$

DistMult (Yang et al. 2015) proposes a diagonal bilinear formulation of form

$$S_{(e_1,r,e_2)} = \mathbf{y}_{e_1}^T \mathbf{B}_r \mathbf{y}_{e_2} \tag{3}$$

where $\mathbf{B}_r$ is diagonal and thus Eq. 3 can be reformulated as

$$S_{(e_1,r,e_2)} = \mathbf{b}_r^T (\mathbf{y}_{e_1} \odot \mathbf{y}_{e_2}) \tag{4}$$

where $\mathbf{b}_r$ is the vector comprising of diagonal entries of $\mathbf{B}_r$ and $\odot$ is the element-wise product.

Another approach for tensor factorization is possible via circular correlation, which provides a transition of tensor factorization method into the complex space i.e. with anti-symmetry relations. HOLE (Nickel et al. 2016) is a compositional embedding method based on composition operation $\circ$.

$$\mathrm{P}(\phi_r(h,t) = 1|\Theta) = \sigma(\eta_{hrt}) = \sigma(r_r^{\mathrm{T}}(e_h \circ e_t)) \tag{5}$$

where $\phi_r(h,t)$ is the probability of relation between h and h, with η_{hrt}, the full tensor product, represented as composition of head and tail entities $(e_h \circ e_t)$ vector and transformed by the relation matrix r_r^{T}.

The composition operation $\circ$ between two entities can be either a full tensor product, concatenations, or even a circular correlation $[a \otimes b]_{ij} = a_i b_j$. Additionally, it can also be of concatenations $[\psi(W(a \oplus b))]_i = \psi\left(\sum_j w_{ij}^a a_j + \sum_j w_{ij}^b b_j\right)$ It can be intuitively thought of as an *or* gate where the feature is *on* if at least one corresponding feature is *on*. Here ψ means non-linearity.

The holographic embedding method comprises of the following scoring objective based on the circular correlation.

$$[\mathbf{a} \star \mathbf{b}]_k = \sum_{i=0}^{d-1} a_i b_{(k+i) \bmod d} \tag{6}$$

The probability of triples can then be modeled as

$$\Pr(\phi_p(s,o) = 1|\Theta) = \sigma(\mathbf{r}_p^T(\mathbf{e}_s \star \mathbf{e}_t)) \tag{7}$$

2.2.3 Translation-based methods

Most translation based methods are related to TransE in terms of their optimization, with differences being in the representation of the entities and relations. STransE (Nguyen et al. 2016b) is comprised of a triples scoring function as

$$f_r(h,t) = \|\mathbf{W}_{r,1}\mathbf{h} + \mathbf{r} - \mathbf{W}_{r,2}\mathbf{t}\|_{l_{1/2}} \tag{8}$$

where $\mathbf{W}$ is the embedding matrix and $\mathbf{r}$ is the relation vector. TransR (Lin et al. 2015b) is comprised of a triples scoring function of the form

$$f_r(h,t) = \|\mathbf{h}\mathbf{M}_r + \mathbf{r} - \mathbf{t}\mathbf{M}_r\|_2^2 \tag{9}$$

Furthermore, Cluster-based TransR (CTransR) is proposed as well, where for each relation, the entity pairs (h,t) are clustered based on their distance $(\mathbf{h} - \mathbf{t})$ where $\mathbf{h}$ and $\mathbf{t}$ are obtained through TransE.

Afterward, the relation vector and transformation matrix are learned separately for each cluster. The scoring function consists of a constraint to limit the divergence of the cluster specific relation vector from the central relation vector.

2.2.4 Complex-space embeddings

ComplEx (Trouillon et al. 2016) is very closely related to HOLE mathematically, where complex embedding is used to solve the problem through latent factorization. The dot product in complex space involves the conjugate transpose of one of the vectors, thus making it non-symmetric and anti-symmetric. Relations can receive different scores, depending on the ordering of the entities involved. The tensor of KG can be learned in simple manner using Eigenvalue decomposition

$$X = EWE^{-1} \tag{10}$$

But for cases where the relations can be anti-symmetric, using eigenvalue decomposition is not possible as it is symmetric in real space. With complex numbers, the dot product is also called *Hermitian* product or *sesquilinear* form and is defined as

$$< u, v >:= \bar{u}^\top v \tag{11}$$

where $u = \text{Re}(u) + \text{Im}(u)$ and $\bar{u} = \text{Re}(u) - \text{Im}(u)$. Eigen decompositions is computationally expensive task except for the space of *normal matrices* where $X\bar{X}^\top = \bar{X}^\top X$. The spectral theorem for normal matrices state that a matrix X is normal if and only if it is unitarily diagonalizable i.e. $X = EW\bar{E}^\top$ where $W \in \mathcal{C}^{n\times n}$ is the diagonal matrix of eigen values with decreasing modulus and $E \in \mathcal{C}^{n\times n}$ is a unitary matrix of eigen vectors, with $\bar{E}$ representing its complex conjugate. The set of purely real normal matrices includes all symmetric, anti-symmetric sign matrices as well as orthogonal matrices and many other matrices that are useful to represent binary relations. However the score of the product must be real number, so only the real part of the decomposition is kept.

$$X = \text{Re}(EW\bar{E}^\top) \tag{12}$$

The above factorization shows that the head entity is the complex conjugate of its tail entity in vector space.

For multiple relations, the scoring function can be extended into different formulations as

$$\phi(r, s, o; \Theta) = \text{Re}(< w_r, e_s, \bar{e}_o >) \tag{13}$$

$$= \text{Re}\left(\sum_{k=1}^{K} w_{rk} e_{sk} \bar{e}_{ok}\right) \tag{14}$$

$$\begin{aligned} &= < \text{Re}(w_r), \text{Re}(e_s), \text{Re}(e_o) > \\ &+ < \text{Re}(w_r), \text{Im}(e_s), \text{Im}(e_o) > \\ &+ < \text{Im}(w_r), \text{Re}(e_s), \text{Im}(e_o) > \\ &- < \text{Im}(w_r), \text{Im}(e_s), \text{Re}(e_o) > \end{aligned} \tag{15}$$

where $w_r \in \mathcal{C}^K$ is a complex vector. Equation 13 is DistMult (Yang et al. 2015) with real embeddings but handles asymmetry due to complex conjugate.

The Knowledge Graph embedding must be capable of leveraging the **symmetry/asymmetry**, **inversion**, and **composition** relations from the observed data in order to predict missing links.

RotatE (Sun et al. 2019) maps the entities and relations to the complex vector space and defines each relation as a rotation from the source entity to the target entity. Given a triplet (h, r, t), we expect $\mathbf{t} = \mathbf{h} \circ \mathbf{r}$ where $\circ$ is element-wise product and $|r_i| = 1$, and $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathcal{C}^k$. The distance function is RotatE, which can be defined as

$$d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| \tag{16}$$

The objective for optimization is based on negative sampling loss

$$L = -\log \sigma(\gamma - d_r(\mathbf{h}, \mathbf{t})) - \sum_{i=1}^{n} \frac{1}{k} \log \sigma(d_r(\mathbf{h}'_i, \mathbf{t}'_i) - \gamma) \tag{17}$$

where γ is the fixed margin, and (h'_i, r, t'_i) are negative samples.

The negative samples are sampled based on the probability

$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(\mathbf{h}'_j, \mathbf{t}'_j)}{\sum_i \exp \alpha f_r(\mathbf{h}'_i, \mathbf{t}'_i)} \tag{18}$$

which is then taken as probability of the negative samples in order to update (17) as

$$\begin{aligned} L = &-\log \sigma(\gamma - d_r(\mathbf{h}, \mathbf{t})) \\ &- \sum_{i=1}^{n} p(h'_j, r, t'_j) \log \sigma(d_r(\mathbf{h}'_i, \mathbf{t}'_i) - \gamma) \end{aligned} \tag{19}$$

In addition to the above mentioned three paradigms of KB embedding using tensor factorization, translation-based methods, and complex spaces, the more recent embedding methods focus on variable geometry of embedding space, such as hyperbolic geometry (Thurston 1982), leading to learning multiple models of embedding in hyperbolic space (Nickel and Kiela 2017, 2018; Ganea et al. 2018; Sala et al. 2018), which shows much promise for both learning compact representation and using smaller dimensions for learning embeddings. Another research direction is along learning ordered embeddings (Vendrov et al. 2016), which are capable of representing hierarchy and order within the geometrical structures (Vilnis et al. 2018). In addition, hyper-complex embeddings (Zhang et al. 2019) extend the generalization of Knowledge Base embedding from complex geometrical space into the hyper-complex space which offer better geometrical interpretations. Quaternion embeddings, which are the embeddings in hyper-complex spaces, are the current state-of-art in the KB embedding space. Additionally, Ebisu and Ichise (2018) learns KB embeddings in Lie Group manifolds, which is a generalization of translation based embedding methods into the tourus shaped manifolds. The relation between manifold learning and its alignment with respect to the real nature of the geometry of the KB is currently unexplored, though (Sala et al. 2018) learn embeddings across different manifolds by means of product of mixture. Such mixtures are capable of offering better performance gains due to greater freedom into structure induction. Hierarchical embeddings are often learned in geometrical spaces which are generalizations of tree including hyperbolic spaces (Nickel and Kiela 2018) where the learned embeddings are highly dependent upon the optimization methods since the space is very restricted and can lead to inefficient local space.

Additionally, Zhang et al. (2018) learns KB embedding by explicit encoding of relations in a form of hierarchy such that closely related relations form a relation cluster, and sub-relation hierarchy. This enables encoding a hierarchy above the defined relations, along with an explicit defined hierarchy of relations which better models the relationship

between relations in the symbolic schema. (Ma et al. 2019) learns embedding via composition from word embeddings. This enables leveraging pretrained information into the KB embeddings, while at the same time also makes the model capable of generalizing to unknown entities.

2.3 Evaluation of Knowledge Base Embedding methods

Knowledge Base (KB) embedding methods are evaluated mostly on the *link prediction problem* i.e., given the head entity and relation, predict the tail entity. The evaluation is performed based on the rank of the correct entity with respect to the other entities which are shown to be relevant. There is some inconsistencies regarding how to rank a correct entity when it has a similar score to an incorrect entity. The idea of using random or lexicographic ordering (Sun et al. 2019) to compensate for simple baseline where all entities are scored equally but correct triples is ranked at the top resulting in lower (and better) rank score is commonly used in more recent experiments (e.g., Zhang et al. 2019).

2.3.1 Datasets

– **WordNet**

WordNet is a structured lexicon designed for the generation of an ordered dictionary and thesaurus. It is popular with applications in Natural Language Processing tasks, such as augmenting data with words replaced by synonyms and antonyms from the WordNet. The WordNet KB consists of two major classes - *synsets* and *wordsense* - with a hierarchy of classes generated from two of them. There are four basic types of properties defined in WordNet - Transitive, Symmetric, DataType and Object Property. All the instances are derived from either $synset$ or $wordsense$ class hierarchy. The wordnet KB dataset defined in Bordes et al. (2013), called WN18, comprises of 18 relations and 40,943 entities, and 151,442 triples predominantly of hyponym and hypernym relations. The WN18 dataset suffers from test leakage, leading to creation of WN18RR (Dettmers et al. 2018) containing 93,003 triples with 40,943 entities and 11 relations. The reduction in number of triples is needed to eliminate test leakages which can lead to noisy performances.

– **Freebase**

Freebase is a large collaborative knowledge graph of general facts (Bollacker et al. 2008) with FB15k (Bordes et al. 2013), which is a relatively dense subgraph of Freebase where all entities are present in Wikilink database. FB15k contains about 14,951 entities with 1,345 different relations. Kadlec et al. (2017) showed issues with the FB15k dataset due to the test leakages into the test set and how it impacts the overall performance of the models by designing a sample baseline bilinear model (Yang et al. 2015) which outperformed other state-of-art models. This resulted in the newer refinement of the dataset to FB15k-237 (Dettmers et al. 2018), a subset of FB15k where inverse relations are removed e.g., (hyponym, hypernym) to obtain 237 relations and 272k triples out of 483k triples.

2.3.2 Evaluation Protocol

For each test triple, the head entity is replaced with every entity of same type in the knowledge base and a similarity score is computed for the corrupted triple. The rank of the original correct triple is obtained after ordering the scores in an ascending order. A similar ranking is also obtained for the triple with the corrupted tails. Aggregating over all the test triples, three metrics are most commonly used for evaluation:

1. **MR** - The mean rank of average rank of the test dataset is calculated by averaging the rank of all the correct triples. Smaller the rank, better the performance. $MR = \frac{1}{N}\sum_{i=1}^{N} rank_i$ where $rank_i$ is rank of the correct triple.
2. **MRR** - The mean reciprocal rank is multiplicative inverse of rank of the correct triple with higher value representing better performance. $MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{rank_i}$ where $rank_i$ is the rank of correct triple.
3. **Hits@n**- It is the count of number of triples whose correct entity is ranked within the first-n entities with higher number representing better performance. $Hits@n = \frac{1}{N}\sum_{i=1}^{N} I_n(rank_i)$ where $I_n(rank_i) = 1$ if $rank_i \leq n$ else 0.

The performance of the most popular KB embedding methods for improved version of Word Net and Freebase data set is given in Tables 2 and 3 respectively.

The study of results presented in Tables 2 and 3 illustrate the current limitations into the KB embedding methods. The Mean Rank (MR) can be used to throw some light into the current challenges in the KB embedding methods since the effective triple prediction ability of KB-embedding methods is still very limited which can be partly attributed to the incomplete nature of the KB with many triples being assigned as false negatives.

2.4 Comparison to symbolic KB

KB embeddings are simpler to use in neural networks and deep learning frameworks due to the breakdown of complex relations as vectors. Embedding methods have been found to be extremely informative for usage as semantic informa-

Table 2 Link prediction results on WN18RR

Model		WN18RR MR	MRR	Hit@10	Hit@3	Hit@1
TransE	Bordes et al. (2013)	3384	0.226	0.501	-	-
DistMult	Yang et al. (2015)	5110	0.43	0.49	0.44	0.39
ComplEx	Trouillon et al. (2016)	5261	0.44	0.51	0.46	0.41
ConvE	Dettmers et al. (2018)	4187	0.43	0.52	0.44	0.40
RotatE	Sun et al. (2019)	*3277*	0.470	0.565	0.488	0.422
a-RotatE	Sun et al. (2019)	3340	0.476	0.571	0.492	0.428
QuatE^1	Zhang et al. (2019)	3472	0.481	0.564	*0.500*	*0.436*
QuatE^2	Zhang et al. (2019)	-	*0.482*	*0.572*	0.499	*0.436*
QuatE^3	Zhang et al. (2019)	**2314**	**0.488**	**0.582**	**0.508**	**0.438**

Results are taken from the respective papersBest results are in bold, and second best are underlined. **a-RotatE** (Sun et al. 2019) describe method with adversarial sampling while **RotatE** refers without adversarial sampling training. (Zhang et al. 2019) have different implementation for Quaternion embeddings, with QuatE^1 being without type constraints, QuatE^2 with regularization and reciprocal learning, and QuatE^3 comprising of type constraints

Bold signify best performance value

tion in both work embeddings and language model methods (Devlin et al. 2019). Hence, it is important to evaluate the geometry, structure, and inference information pertained within the embeddings to compare the embedding methods with the symbolic representations. There exist a few works that evaluate the geometry and structure of embeddings learned by KB embedding methods (Chandrahas and Talukdar 2018; Gutiérrez-Basulto and Schockaert 2018; Kazemi and Elqursh 2017) which give a more detailed exploration of the geometry of the embeddings. The geometry induced by different embedding methods differ due to their explicit geometrical assumption (Chandrahas and Talukdar 2018). For example, translation based embedding methods are different from tensor decomposition, while the embeddings in complex and hyperbolic spaces are often different from the regular Euclidean spaces, since each geometry impacts the distance and norm computation which are crucial in understanding and reasoning about the learned embedding points. Moreover, distribution based embedding methods are often difficult to evaluate due to their geometrical spaces lying in probabilistic domain as opposed to a point in geometry.

Chandrahas and Talukdar (2018) study the embedding space of translation and tensor factorization based embedding methods in terms of first and second order moments (mean and variance) and find that, it is difficult to obtain an insight related to the spread of the embeddings and their performance in evaluation tasks. While different methods of embeddings learn vector space with distinct properties, their

Table 3 Link prediction results on FB15K-237

Model		FB15K-237 MR	MRR	Hit@10	Hit@3	Hit@1
TransE	Bordes et al. (2013)	357	0.294	0.465	-	-
DistMult	Yang et al. (2015)	254	0.241	0.419	0.263	0.155
ComplEx	Trouillon et al. (2016)	339	0.247	0.428	0.275	0.158
ConvE	Dettmers et al. (2018)	244	0.325	0.501	0.356	0.237
RotatE	Sun et al. (2019)	185	0.297	0.480	0.328	0.205
a-RotatE	Sun et al. (2019)	177	0.338	0.533	0.375	0.241
QuatE^1	Zhang et al. (2019)	*176*	0.311	0.495	0.342	0.221
QuatE^2	Zhang et al. (2019)	-	**0.366**	**0.556**	**0.401**	**0.271**
QuatE^3	Zhang et al. (2019)	**87**	*0.348*	*0.550*	*0.382*	*0.248*

Results are taken from the respective papers. Best results are in bold, and second best are underlined. **a-RotatE** (Sun et al. 2019) describe method with adversarial sampling while **RotatE** refers without adversarial sampling training. Zhang et al. (2019) have different implementation for Quaternion embeddings, with QuatE^1 being without type constraints, QuatE^2 with regularization and reciprocal learning, and QuatE^3 comprising of type constraints

Bold signify best performance value

performance in general is independent of the organization of vector within the geometry.

A fundamental difference between the symbolic representation of ontology and Knowledge Base (KBs) lies in the restriction placed on the data points. The hard-coded restrictions presented as rules in the symbolic interpretation enables a more powerful inference mechanism to generate and validate newer relations, while the embedding representation often do not offer a faithful obedience to the rules that are universally true, due to the noisy nature of point representation as a result of generalization. Gutiérrez-Basulto and Schockaert (2018) explore the representation of rules learned by KB embeddings. Their evaluation is based on the rules described by the ontology of KB and using those rules to access the alignment of KB embeddings to those rules in a subset of geometric space manner. For example; If one relation is a subset of another, then it is likely that the space describing such relation has some geometric regularities which the embedding space exploits. Unfortunately, a lot of KBs are incomplete and they play a significant role in creating a noisy sub-spaces making it difficult to infer the spatial relation in parallel to the symbolic rule-based coding. Kazemi and Poole (2018) and Kazemi and Elqursh (2017) also show the limitation of expressiveness of translation based methods in a manner similar to dimensionality reductions where the expressiveness is limited but regularity is stronger with smaller number of dimensions. From this view, it is plausible to see more further improvements are necessary to be able to use KB embeddings as a knowledge encoding representation. Such insights are also able to explain the performance of tensor factorization methods in KB embeddings such as Tucker factorization (Balazevic et al. 2019b; 2019a).

3 Question-Answering Architectures

We briefly review some of the neural networks widely used for question answering. Neural networks (LeCun et al. 2015) enable learning of representation of data with multiple levels of abstraction. These levels of abstractions enable deep learning methods to generalize information, while also being able to narrow down to a specific aspect of information. Different architectures of neural networks, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and more recently, transformer networks (Vaswani et al. 2017), are widely used for challenging learning tasks (Devlin et al. 2019), including question answering (Iyyer et al. 2014; Antol et al. 2015). Additionally, smaller models are widely used for unsupervised pretraining (Mikolov et al. 2013; Pennington et al. 2014). Attention mechanisms (Bahdanau et al. 2015) have proven useful for filtering useful content for retrieval tasks in NQAs.

Memory Networks (Weston et al. 2015) and related architectures including Neural Turing Machines (Graves et al. 2014; Sukhbaatar et al. 2015) are neural networks with external memory. They represent an extremely useful paradigm for solving factoid question answering (Bordes et al. 2015) and question answering involving reasoning (Weston et al. 2015). Their novelty lies in their ability to manipulate external memory locations, such as a Knowledge Base (KB) or a Universal Schema (Riedel et al. 2013). Another advantage lies in their different level of guidance applied (i.e, additional information incorporation is easier than in standard neural architectures (Weston et al. 2015)). We show a simple architecture used for general question-answering systems in Fig. 3.

3.1 Discussion

The general architecture of neural network based methods with knowledge bases for question answering shows some of the challenges along with the advantages offered by the current architecture. The biggest bottleneck in both performance and model's scalability lies in the relation to the KG since KGs are represented as independent representation of entities and relations. The incorporation of millions of entities is often infeasible and therefore a filtering mechanism is often used. A pairwise similarity metric (e.g., cosine similarity) is a simple choice for such filtering but this leads to the problem of a large-number of

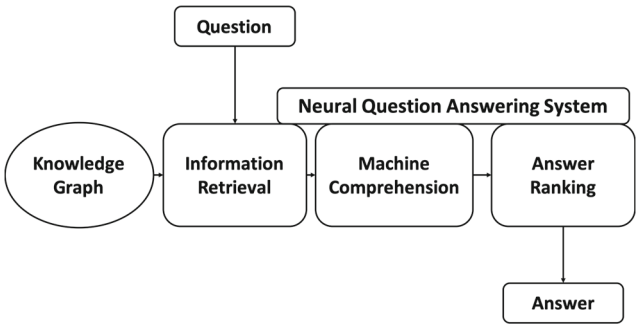

Fig. 3 A general architecture for neural question answering, comprised of three components: *Information Retrieval*, which often interacts with Knowledge Graph in embedded form, for generating answer candidates; *Machine Comprehension* and *Answer Ranking*, which are mostly model-dependent. The *Machine Comprehension* component is comprised of attending over multiple layers of information to generate answer candidates. The *Answer Ranking* is based on relevance to the question, while the *machine comprehension* is focused on validating the answer by attending over information sources. Memory networks enable *Machine Comprehension* to interact directly with KB by performing multi-stage retrieval in an iterative manner

relevant entities being excluded from the problem context and also the subset of KG being selected is already quite similar to the current context and as such may not have as much information value as other components of KG. An ideal scenario would be a more compact representation of the entire KG itself as provided by graph neural network based methods (Wu et al. 2020) though a more conditional filtering approach is preferable. Overall, graph neural networks should be helpful in overcoming some of the limitations of current KB-incorporated neural network approaches.

4 Factoid Question Answering

Factoid Question Answering (FQA) refers to questions which can be answered effectively by a phrase or an entity of a KG. There are mainly two approaches to FQAs - answering questions over a KG or obtaining answer from natural text using open information extraction mechanisms. Few approaches exist which attempt to combine both resources or use multiple KBs (Fader et al. 2014). In this section, we explore the factoid question answering methods in the context of neural networks. Diefenbach et al. (2018) is a more specific survey of factoid question answering methods including non-neural network based method.

A Knowledge Graph-based factoid question answering involves mapping the question in natural language into triples of Knowledge Graphs. The distinction is made between FQA systems mapping to just one triples and mapping to multiple triples. The system which maps to a single triple is called Simple Question Answering (SimpleQA). Simple QA is a relatively easy problem compared to other factoid and non-factoid QAs. They are also the most frequent type of questions asked (Fader et al. 2013). A SimpleQA task involves answering a question such as *"What is the hometown of Obama?"* which asks for a direct topic of an entity *"Obama"* which is *"hometown"*. The challenges to SimpleQA systems lie in how to formulate a question in multiple ways, making the mapping process hard to generalize. Another highly successful paradigm to factoid question is semantic parsing (Berant et al. 2013; Yih et al. 2014; Yao and Durme 2014). The semantic parser transforms natural language into logical form. It is capable of solving tricky questions involving multiple relations and questions involving ordering.

4.1 Simple Question Answering (SimpleQA)

A common approach to solving a SimpleQA problem is to extract a set of candidate answers from Knowledge Base using relation extraction (Yao and Durme 2014, 2015; Yih et al. 2014; Bast and Haussmann 2015) or distributed representation (Bordes et al. 2014; Dong et al. 2015; Xu et al. 2016).**WikiAnswers** (Fader et al. 2013) is introduced as a paraphrasing dataset which helps generalize for unseen words and question patterns. Another dataset, **SimpleQuestions**, is introduced by (Bordes et al. 2015). SimpleQA involves embedding of a knowledge base to find the entity of the knowledge base which is closest to the question's representation as the answer. The general framework for factoid question answering is: Given an input question sentence $S = \{w_1, w_2, .., w_Q\}$ and a sentence representation $s \in R^k$, we find the entity e in KB E such that $f(s, e) > f(s, e'), e' \cup e = E$.

A CNN-based approach can be applied to factoid QAs (Yin et al. 2016b) with a two-step pipeline: entity linking, and fact selection. Memory networks are applied in Bordes et al. (2015) to simple question answering. The memory network consists of a memory, and of a neural network which is trained to query that memory, given some inputs. It consists of four components: Input map (I); Output map (O); Generalization (G); and Response (R). The workflow is to store *Freebase* into memory and then train the model to answer questions. A KB triplet is represented by a bag-of-words model, with subject and relationship having value 1 and object entries set to $1/k$, where k is the number of objects. The answer ranking is based on cosine similarity. Lukovnikov et al. (2017) encode questions using GRUs, and a word is represented as a concatenation of Glove vectors (Pennington et al. 2014) with character level encoding. He and Golub (2016) propose a character-level approach based on the attention-enhanced encoder-decoder architecture (Bahdanau et al. 2015). The model of He and Golub (2016) consists of a character-level RNN-based question encoder and an attention-enhanced RNN decoder, coupled with two separate character-level CNN-based entity label and predicate URI encoders.

A word-level RNN-based approach with emphasis on possible paraphrases of questions is proposed by Dai et al. (2016). The task of predicting subject and relation is factorized into two sub-tasks: prediction of relation first, followed by entity given the relation and question. Both (Dai et al. 2016) and Yin et al. (2016b) improve the performance of their approaches using a BiLSTM-CRF tagging model which is separately trained to label parts of the question as entity mention or context (relation pattern).

4.2 Multi-Relation Question Answering

The formulation of multi-relation question answering is driven by the necessity to map questions in natural text to more than one triple in a knowledge base. For challenging questions, such as "*What mountain is the highest in North America?*," which requires learning a representation for mathematical function "highest", Xu

et al. (2016) use textual data to filter out wrong answers. A dependency parser-based query node expansion is devised in Yao and Durme (2014) where ClueWeb text is used to learn correlation between KB relations and words using co-occurrence statistics with the alignment model. Dong et al. (2015) uses multi-column CNNs to understand questions from three different aspects: answer path, answer context, and answer type. Then it learns their distributed representations. Yang et al. (2014) maps natural language to knowledge base by semi-automatically generating mappings between knowledge base triples and natural text, using information extraction methods.

Yin et al. (2016a) propose an encoder-decoder framework model for factoid question answering, with ability to query a KB. Jain (2016) pre-process Freebase to remove dummy entities and to obtain more direct triples. An L-hop factual memory network is constructed for computational layers, where each layer accesses candidate facts and question embedding.

A major constraint on factoid question answering models is the data limitation. While there are multiple ways to phrase a single question, the dataset size suffers from sparseness and is unable to work with methods that require a larger training datasize. SimpleQA have made substantial progress recently, due to the introduction of the SimpleQuestions (Bordes et al. 2015) dataset, making larger neural network models trainable until convergence without overfitting. While the focus on the SimpleQA task is to generalize mapping of questions to facts, non-simple QA tasks and multi-resource open domain QA tasks require learning the mathematical and functional dependencies required to answer the question. This makes the problem considerably more complex, while at the same time, limited training data constrains the model to use lesser parameters. There are also very few methods which attempt to leverage multiple knowledge sources.

4.3 Discussion

Factoid question answering relies on KBs for problem formulation and are therefore tightly coupled with the role of KBs in neural question answering. The incompleteness of KBs is aggravating to factoid question answering systems since the training data itself is dependent on KBs, and incompleteness often leads to a larger uncertainty within the models predictions framework. A viable alternative is to consider usage of multiple KBs or pretrained language model as a more general form of representation to such problems. Such systems should be capable of handling ambiguities often encountered in factoid question answering.

5 Attention-based Question Answering

Attention-based QA are extremely popular approaches for multi-modal data problems such as Visual Question Answering (VQA) and problems requiring deeper understanding of input data, such as Reading Comprehension (RC) (also called Machine Comprehension). A common approach to VQA concatenates visual and textual representations obtained from CNN and RNN respectively, to perform joint inference. This approach can be improved upon by introduction of attention maps for input image, each with embedding for a certain section of image, which are then attended over using attention mechanism for learning a joint embedding which then performs the final classification or sequence generation task. Multimodal bilinear compact pooling (Fukui et al. 2016) proposes an efficient but highly optimized bilinear pooling over two data sources, enabling a robust embedding for visual question answering.

R-Net (Wang et al. 2017) obtain significant performance gains on RC dataset, SQuAD (Rajpurkar et al. 2016). The difference between VQA and RC lies in decoding stage of inference, where VQA decoding is done based upon preset vocabulary. RC datasets require sampling of input text to generate answer phrases or sequences. This requires probabilistic decoding, using a combination of language decoding and pointer networks (Vinyals et al. 2015) to obtain answer effectively. R-Net uses GRUs to learn embeddings for the input question and sentence, which are then passed to gated attention-based recurrent networks to determine importance of information in the passage regarding a question. Each passage representation incorporates aggregating matching information from the whole question. Another gate is added to determine the importance of passage parts relevant to the question. Another attention to match over itself is used to incorporate context into question-aware embeddings. A Pointer Network is used to predict the start and end position of the answer. The success of R-Net has given rise to Reasonet (Shen et al. 2017), Fusionnet (Huang et al. 2018), QA-Net (Yu et al. 2018), Macnet (Pan et al. 2018), and S2-Net (Park et al. 2019).

While there are many different variants of visual question answering and reading comprehension methods in literature (see Antol et al. (Antol et al. 2015) for more details), the underlying mechanism entails learning the fixed vector representation for both question and input data (either image or text), then using the attention or bilinear pooling to learn joint embeddings. The learned vectors are used for making predictions. We do not attempt to cover the entire attention-based question-answering methods, due to space and time constraints. Recently, it was found that

using transfer-learning approaches (Devlin et al. 2019) often significantly improves the performance of the model in their introduction of BERT. This was utilized in multiple novel works.

CoQA (Reddy et al. 2019) was adapted to BERT by Zhu et al. (2018) to show superior results. BERT was proven to be efficient for reading comprehension in multi-hop (Min et al. 2019) and visual question answering (Li et al. 2019). Usage of BERT in the domain of Artificial Social Intelligence (ASI) was shown by Zadeh et al. (2019).

The incomparable score issue of BERT which is caused by the fact that original version of the algorithm considers passages corresponding to the same question as independent has been solved in the extension by Wang et al. (2019). There are two specialized BERT models in the biology domain question answering which start with the original BERT model: BioBERT (Lee et al. 2019) is pre-trained on biomedical articles from PMC full text articles and PubMed abstracts, ClinicalBERT (Huang et al. 2019) is pre-trained on clinical notes from the MIMIC-III dataset. Alsentzer et al. (2019) extended the BioBERT model by pre-training it on the full set of MIMIC-III notes and a subset of discharge summaries. A merger of BERT and Anserini (Yang et al. 2018) was created by Yang et al. (2019) under the name BERTserini to facilitate the ability to identify answers in an end-to-end fashion from a large corpus of Wikipedia articles. A variation of BERT on OpenBookQA (Mihaylov et al. 2018) was attempted by Banerjee et al. (2019) and was shown to have an 11.6% improvement over the contemporary state of the art. On the matter of optimization, RoBERTa (Liu et al. 2019) and ALBERT (Lan et al. 2019) attempt to optimize BERT in different ways. RoBERTa focuses on the pre-training of BERT while ALBERT incorporates two parameter-reduction techniques for the purpose of lower memory consumption and faster training speed. We summarize the attention-based question answering methods in Table 4.

5.1 KB incorporation in attention-based question answering

The attention-based question answering approach is based upon the neural-networks and is designed mostly in an end-to-end fashion where using a Knowledge Base (KB) is challenging primarily due to the overall size of the KB. There exists some alternatives for using KBs in attention based question answering, namely graph networks (Schlichtkrull et al. 2018), and Memory Networks (Weston et al. 2015). Another challenge still exists for using KBs with attention-based question answering in the form of filtering. Since neural networks rely on non-linear transformation of input features, the size of input features is limited. KBs are represented in terms of entities and relations embeddings, which is intractable to be incorporated as features into the neural networks. This requires the QA model to limit the number of entities and

Table 4 Comparative analysis of attention-based question answering

Study	Method	Problem Domain
Fukui et al. (2016)	Bilinear pooling	Visual Question Answering
R-Net (Wang et al. 2017)	GRUs	Reading Comprehension
Reasonet (Shen et al. 2017)	Multi-turn inference	Reading Comprehension
Fusionnet (Huang et al. 2018)	Fusion of levels of abstraction	Reading Comprehension
QANet (Yu et al. 2018)	Self-attention	Reading Comprehension
Macnet (Pan et al. 2018)	Transfer Learning	Abstractive Summarization
S2-Net (Park et al. 2019)	Self-matching networks	Reading Comprehension
SDNet (Zhu et al. 2018)	Attention Mechanism	Conversational Question Answering
Min et al. (2019)	Question Decomposition	Reading Comprehension
Li et al. (2019)	Image to text transformation	Visual Question Answering
Zadeh et al. (2019)	BERT	Visual Question Answering
BioBERT (Lee et al. 2019)	BERT	Biomedical Text
Clinical BERT (Huang et al. 2019)	BERT	Clinical Text
Alsentzer et al. (2019)	BioBERT	Clinical Text
BERTserini (Yang et al. 2019)	BERT and IR	Reading Comprehension
Banerjee et al. (2019)	BERT and IR	Open Question Answering
RoBERTa (Liu et al. 2019)	BERT pretraining	NLP (multi-domain)
ALBERT (Lan et al. 2019)	Parameters reduction on BERT	NLP (multi-domain)

relations which can be considered as features, often done via simple linear similarity measure (e.g., cosine similarity). It is one of the primary factor for currently limited success of KBs incorporation into attention-based QA models, where using pre-trained language models (e.g., BERT) provides a tractable model which contains some aspect of knowledge in the form of its internal representations (Devlin et al. 2019).

6 Conclusion

In this paper, we surveyed multiple areas of neural question answering, including Knowledge Base embeddings, neural networks architecture, and various advances in factoid and attention-based question answering. While Knowledge Base (KB) embeddings methods are advanced enough to be relied upon as information resources, we observe that multitudes of works on question answering still rely on older approaches. This leads to suboptimal performance from KBs, making a proper evaluation difficult. We believe this paper serves as an important milestone in syncing up the progress across different fields, in order to leverage strong, connected components for building richer sets of question answering models. The advancements in research in KB embeddings toward different geometrical spaces, including hyperbolic spaces, suggests that neural networks with representational capacity in such spaces with curvature may be the next application for building question-answering models.

References

Alsentzer, E., Murphy, J.R., Boag, W., Weng, W.H., Jin, D., Naumann, T., McDermott, M. (2019). *Publicly available clinical bert embeddings*. CoRR, arXiv:1904.03323.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D. (2015). VQA: visual question answering. In *ICCV* (pp. 2425–2433).

Antoniou, G., & Van Harmelen, F. (2004). Web ontology language: Owl. In *Handbook on ontologies* (pp. 67–92): Springer.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, *25*(1), 25–29.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G. (2007). Dbpedia: A nucleus for a web of open data. In *ISWC* (pp. 722–735): Springer.

Bahdanau, D., Cho, K., Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*.

Balazevic, I., Allen, C., Hospedales, T.M. (2019a). Hypernetwork knowledge graph embeddings. In *ICANN* (pp. 553–565).

Balazevic, I., Allen, C., Hospedales, T.M. (2019b). Tucker: Tensor factorization for knowledge graph completion. In *EMNLP-IJCNLP* pp. 5184–5193.

Banerjee, P., Pal, K.K., Mitra, A., Baral, C. (2019). Careful selection of knowledge to solve open book question answering. In *ACL* (pp. 6120–6129).

Bast, H., & Haussmann, E. (2015). More accurate question answering on freebase. In *CIKM* (pp. 1431–1440).

Berant, J., Chou, A., Frostig, R., Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *EMNLP* (pp. 1533–1544).

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, *32*(Database-Issue), 267–270.

Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD* (pp. 1247–1250).

Bordes, A., Weston, J., Collobert, R., Bengio, Y. (2011). Learning structured embeddings of knowledge bases. In *AAAI*.

Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NeurIPS* (pp. 2787–2795).

Bordes, A., Chopra, S., Weston, J. (2014). Question answering with subgraph embeddings. In *EMNLP* (pp. 615–620).

Bordes, A., Usunier, N., Chopra, S., Weston, J. (2015). Large-scale simple question answering with memory networks. CoRR, arXiv:1506.02075.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr, E.R.H., Mitchell, T.M. (2010). Toward an architecture for never-ending language learning. In *AAAI*.

Chandrahas, S.A., & Talukdar, P.P. (2018). Towards understanding the geometry of knowledge graph embeddings. In *ACL* (pp. 122–131).

Chandrasekaran, B., Josephson, J.R., Benjamins, V.R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems and Their Applications*, *14*(1), 20–26.

Dai, Z., Li, L., Xu, W. (2016). CFO: Conditional focused neural question answering with large-scale knowledge bases. In *ACL*.

Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *AAAI* (pp. 1811–1818).

Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT* (pp. 4171–4186).

Diefenbach, D., Lopez, V., Singh, K., Maret, P. (2018). Core techniques of question answering systems over knowledge bases: a survey. *Knowledge and Information Systems*, *55*(3), 529–569.

Dong, L., Wei, F., Zhou, M., Xu, K. (2015). Question answering over freebase with multi-column convolutional neural networks. In *ACL* (pp. 260–269).

Ebisu, T., & Ichise, R. (2018). Toruse: Knowledge graph embedding on a lie group. In *AAAI* (pp. 1819–1826).

Ehrlinger, L., & Wöß, W. (2016). Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, suCCESS). *Metallurgy - Proceedings*, 48.

Fader, A., Zettlemoyer, L.S., Etzioni, O. (2013). Paraphrase-driven learning for open question answering. In *ACL* (pp 1608–1618).

Fader, A., Zettlemoyer, L., Etzioni, O. (2014). Open question answering over curated and extracted knowledge bases. In *KDD* (pp 1156–1165).

Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M. (2016). Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP* (pp. 457–468).

Ganea, O., Bécigneul, G., Hofmann, T. (2018). Hyperbolic entailment cones for learning hierarchical embeddings. In *ICML* (pp. 1632–1641).

García-Durán, A., Bordes, A., Usunier, N. (2015). Composing relationships with translations. In *EMNLP* (pp. 286–290).

García-Durán, A., Bordes, A., Usunier, N., Grandvalet, Y. (2016). Combining two and three-way embedding models for link prediction in knowledge bases. *Journal of Artificial Intelligence Research*, *55*, 715–742.

Graves, A., Wayne, G., Danihelka, I. (2014). Neural turing machines. CoRR, arXiv:1410.5401.

Gruber, T. (2009). Ontology. *Encyclopedia of database systems*, 1963–1965.

Gutiérrez-Basulto, V., & Schockaert, S. (2018). From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. In *Principles of Knowledge Representation and Reasoning* (pp. 379–388).

Guu, K., Miller, J., Liang, P. (2015). Traversing knowledge graphs in vector space. In *EMNLP* (pp. 318–327).

He, S., Liu, K., Ji, G., Zhao, J. (2015). Learning to represent knowledge graphs with gaussian embedding. In *CIKM* (pp. 623–632): ACM.

He, X., & Golub, D. (2016). Character-level question answering with attention. In *EMNLP* (pp. 1598–1607).

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *ACL* (pp. 328–339).

Huang, H., Zhu, C., Shen, Y., Chen, W. (2018). Fusionnet: Fusing via fully-aware attention with application to machine comprehension. In *ICLR*.

Huang, K., Altosaar, J., Ranganath, R. (2019). Clinicalbert: Modeling clinical notes and predicting hospital readmission. CoRR, arXiv:1904.05342.

Iyyer, M., Boyd-Graber, J.L., Claudino, L.M.B., Socher, III. R.H.D. (2014). A neural network for factoid question answering over paragraphs. In *EMNLP* (pp. 633–644).

Jain, S. (2016). Question answering over knowledge base using factual memory networks. In *Student Research Workshop, SRW@HLT-NAACL* (pp. 109–115).

Ji, G., He, S., Xu, L., Liu, K., Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. In *ACL* (pp. 687–696).

Kadlec, R., Bajgar, O., Kleindienst, J. (2017). Knowledge base completion: Baselines strike back. In *Workshop on Representation Learning for NLP, Rep4NLP@ACL* (pp. 69–74).

Kafle, S., de Silva, N., Dou, D. (2019). An overview of utilizing knowledge bases in neural networks for question answering. In *IRI, IEEE* (pp. 326–333).

Kazemi, V., & Elqursh, A. (2017). Show, ask, attend, and answer: A strong baseline for visual question answering. CoRR, arXiv:1704.03162.

Kazemi, S.M., & Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. In *NeurIPS* (pp. 4289–4300.

Kratzwald, B., Eigenmann, A., Feuerriegel, S. (2019). Rankqa: Neural question answering with answer re-ranking. In *ACL* (pp. 6076–6085).

Krompaß, D., Baier, S., Tresp, V. (2015). Type-constrained representation learning in knowledge graphs. In *ISWC* (pp. 640–655): Springer.

Krótkiewicz, M., Wojtkiewicz, K., Jodłowiec, M. (2018). Towards semantic knowledge base definition. In *International scientific conference BCI 2018 Opole* (pp. 218–239): Springer.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019). Albert: a lite bert for self-supervised learning of language representations. CoRR, arXiv:1909.11942.

LeCun, Y., Bengio, Y., Hinton, G.E. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J. (2019). Biobert: pre-trained biomedical language representation model for biomedical text mining. CoRR, arXiv:1901.08746.

Li, H., Wang, P., Shen, C., van den Hengel, A. (2019). Visual question answering as reading comprehension. In *CVPR* (pp. 6319–6328).

Liang, C., & Forbus, K.D. (2015). Learning plausible inferences from semantic web knowledge by combining analogical generalization with structured logistic regression. In *AAAI* (pp. 551–557).

Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S. (2015a). Modeling relation paths for representation learning of knowledge bases. In *EMNLP* (pp. 705–714).

Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X. (2015b). Learning entity and relation embeddings for knowledge graph completion. In *AAAI* (pp. 2181–2187).

Liu, H., & Singh, P. (2004). Conceptnet-a practical commonsense reasoning tool-kit. *BT Technology journal*, *22*(4), 211–226.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). Roberta: a robustly optimized bert pretraining approach. CoRR, arXiv:1907.11692.

Lukovnikov, D., Fischer, A., Lehmann, J., Auer, S. (2017). Neural network-based question answering over knowledge graphs on word and character level. In *WWW* (pp. 1211–1220).

Luo, Y., Wang, Q., Wang, B., Guo, L. (2015). Context-dependent knowledge graph embedding. In *EMNLP* (pp. 1656–1661).

Ma, L., Sun, P., Lin, Z., Wang, H. (2019). Composing knowledge graph embeddings via word embeddings. CoRR, arXiv:1909.03794.

Mihaylov, T., Clark, P., Khot, T., Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP* (pp. 2381–2391).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NeurIPS* (pp. 3111–3119).

Miller, G.A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Min, S., Zhong, V., Zettlemoyer, L., Hajishirzi, H. (2019). Multi-hop reading comprehension through question decomposition and rescoring. In *ACL* (pp. 6097–6109).

Mitchell, T.M., Cohen, W.W., Jr, E.R.H., Talukdar, P.P., Yang, B., Betteridge, J., Carlson, A., Mishra, B.D., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E.A., Ritter, A., Samadi, M., Settles, B., Wang, R.C., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., Welling, J. (2018). Never-ending learning. *Communications of the ACM*, *61*(5), 103–115.

Neelakantan, A., Roth, B., McCallum, A. (2015). Compositional vector space models for knowledge base completion. In *ACL* (pp. 156–166).

Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M. (2016a). Neighborhood mixture model for knowledge base completion. In *CoNLL* (pp. 40–50).

Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M. (2016b). Stranse: a novel embedding model of entities and relationships in knowledge bases. In *NAACL-HLT* (pp. 460–466).

Nickel, M., Tresp, V., Kriegel, H. (2011). A three-way model for collective learning on multi-relational data. In *ICML* (pp. 809–816).

Nickel, M., Tresp, V., Kriegel, H. (2012). Factorizing YAGO: scalable machine learning for linked data. In *WWW* (pp. 271–280).

Nickel, M., & Tresp, V. (2013). Logistic tensor factorization for multi-relational data. CoRR.

Nickel, M., Rosasco, L., Poggio, T.A. (2016). Holographic embeddings of knowledge graphs. In *AAAI* (pp. 1955–1961).

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *NeurIPS* (pp. 6338–6347).

Nickel, M., & Kiela, D. (2018). Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *ICML* (pp. 3776–3785).

Pan, B., Yang, Y., Li, H., Zhao, Z., Zhuang, Y., Cai, D., He, X. (2018). Macnet: Transferring knowledge from machine comprehension to sequence-to-sequence models. In *NeurIPS* (pp. 6095–6105).

Park, C., Lee, C., Hong, L., Hwang, Y., Yoo, T., Jang, J., Hong, Y., Bae, K.H., Kim, H.K. (2019). S2-net: Machine reading comprehension with sru-based self-matching networks. *ETRI Journal*.

Pennington, J., Socher, R., Manning, C.D. (2014). Glove: Global vectors for word representation. In *EMNLP* (pp. 1532–1543).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog, 1*(8).

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP* (pp. 2383–2392).

Reddy, S., Chen, D., Manning, C.D. (2019). Coqa: A, conversational question answering challenge. *TACL*, *7*, 249–266.

Riedel, S., Yao, L., McCallum, A., Marlin, B.M. (2013). Relation extraction with matrix factorization and universal schemas. In *NAACL-HLT* (pp. 74–84).

Sala, F., Sa, C.D., Gu, A., Ré, C. (2018). Representation tradeoffs for hyperbolic embeddings. In *ICML* (pp. 4457–4466).

Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M. (2018). Modeling relational data with graph convolutional networks. In *ESWC* (pp. 593–607).

Seaborne, A., & Prud'hommeaux, E. (2006). Sparql query language for rdf. *W3C recommendation*.

Shearer, R., Motik, B., Horrocks, I. (2008). Hermit: A highly-efficient OWL reasoner. In *Fifth OWLED Workshop on OWL: Experiences and Directions@ISWC*.

Shen, Y., Huang, P., Gao, J., Chen, W. (2017). Reasonet: Learning to stop reading in machine comprehension. In *KDD* (pp. 1047–1055).

Shi, B., & Weninger, T. (2017). Proje: Embedding projection for knowledge graph completion. In *AAAI* (pp. 1236–1242).

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official google blog*.

Socher, R., Chen, D., Manning, C.D., Ng, A.Y. (2013). Reasoning with neural tensor networks for knowledge base completion. In *NeurIPS* (pp. 926–934).

Suchanek, F.M., Kasneci, G., Weikum, G. (2007). Yago: a core of semantic knowledge. In *WWW, ACM* (pp. 697–706).

Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R. (2015). End-to-end memory networks. In *NeurIPS* (pp. 2440–2448).

Sun, Z., Deng, Z., Nie, J., Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. In *ICLR*.

Sutskever, I., Salakhutdinov, R., Tenenbaum, J.B. (2009). Modelling relational data using bayesian clustered tensor factorization. In *NeurIPS* (pp. 1821–1828).

Thurston, W.P. (1982). Three dimensional manifolds, kleinian groups and hyperbolic geometry. *Bulletin of the American Mathematical Society*, *6*(3), 357–381.

Toutanova, K., Lin, V., Yih, W., Poon, H., Quirk, C. (2016). Compositional learning of embeddings for relation paths in knowledge base and text. In *ACL*.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G. (2016). Complex embeddings for simple link prediction. In *ICML* (pp. 2071–2080).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. In *NeurIPS* (pp. 5998–6008).

Vendrov, I., Kiros, R., Fidler, S., Urtasun, R. (2016). Order-embeddings of images and language. In *ACL*.

Vilnis, L., Li, X., Murty, S., McCallum, A. (2018). Probabilistic embedding of knowledge graphs with box lattice measures. In *ACL* (pp. 263–272).

Vinyals, O., Fortunato, M., Jaitly, N. (2015). Pointer networks. In *NeurIPS* (pp. 2692–2700).

Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. In *ACL* (pp. 189–198).

Wang, Z., Zhang, J., Feng, J., Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *AAAI, Citeseer* (pp. 1112–1119).

Wang, Z., & Li, J. (2016). Text-enhanced representation learning for knowledge graph. In *IJCAI, AAAI Press* (pp. 1293–1299).

Wang, Z., Ng, P., Ma, X., Nallapati, R., Xiang, B. (2019). Multi-passage bert: A globally normalized bert model for open-domain question answering. In *EMNLP-IJCNLP* (pp. 5881–5885).

West, R., Gabrilovich, E., Murphy, K., Sun, S., Gupta, R., Lin, D. (2014). Knowledge base completion via search-based question answering. In *WWW* (pp. 515–526).

Weston, J., Chopra, S., Bordes, A. (2015). Memory networks. In *ICLR*.

Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A.R., van den Hengel, A. (2017). Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, *163*, 21–40.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y. (2020). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems.

Xu, K., Reddy, S., Feng, Y., Huang, S., Zhao, D. (2016). Question answering on freebase via relation extraction and textual evidence. In *ACL*.

Yang, M., Duan, N., Zhou, M., Rim, H. (2014). Joint relational embeddings for knowledge-based question answering. In *EMNLP* (pp. 645–650).

Yang, B., Yih, W., He, X., Gao, J., Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

Yang, P., Fang, H., Lin, J. (2018). Anserini: Reproducible ranking baselines using lucene. *Journal of Data and Information Quality (JDIQ)*, *10*(4), 16.

Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., Lin, J. (2019). End-to-end open-domain question answering with bertserini. *NAACL-HLT*, 72.

Yao, X., & Durme, B.V. (2014). Information extraction over structured data: Question answering with freebase. In *ACL* (pp. 956–966).

Yao, X. (2015). Lean question answering over freebase from scratch. In *NAACL-HLT* (pp. 66–70).

Yih, W., He, X., Meek, C. (2014). Semantic parsing for single-relation question answering. In *ACL* (pp. 643–648).

Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X. (2016a). Neural generative question answering. In *IJCAI* (pp. 2972–2978).

Yin, W., Yu, M., Xiang, B., Zhou, B., Schütze, H. (2016b). Simple question answering by attentive convolutional neural network. In *COLING* (pp. 1746–1756).

Yu, A.W., Dohan, D., Luong, M., Zhao, R., Chen, K., Norouzi, M., Le, Q.V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Zadeh, A., Chan, M., Liang, P.P., Tong, E., Morency, L. (2019). Social-iq: A question answering benchmark for artificial social intelligence. In *CVPR* (pp. 8807–8817).

Zhang, Z., Zhuang, F., Qu, M., Lin, F., He, Q. (2018). Knowledge graph embedding with hierarchical relation structure. In *EMNLP* (pp. 3198–3207).

Zhang, S., Tay, Y., Yao, L., Liu, Q. (2019). Quaternion knowledge graph embeddings. In *NeurIPS* (pp. 2731–2741).

Zhu, C., Zeng, M., Huang, X. (2018). Sdnet: Contextualized attention-based deep network for conversational question answering. CoRR arXiv:1812.03593.

Sabin Kafle is a PhD student in Computer Science at University of Oregon, USA under advice of Professor Dejing Dou. He received his B.Sc in Computer Engineering from Tribhuvan University, Institute of Engineering, Nepal and his M.Sc in Computer and Information Science from University of Oregon, USA. His research interest includes Natural Language Processing, Neural Networks, and Knowledge Base. Currently, he is working on applying Knowledge Bases to neural networks.

Nisansa de Silva joined University of Moratuwa in 2011, as a lecturer. He received his B.Sc (Hons.) in Engineering from University of Moratuwa, Sri Lanka and his M.Sc in Computer and Information Science from University of Oregon, USA. He is reading for his PhD in Computer and Information Science at University of Oregon, USA under the advisement of Professor Dejing Dou. Mr. de Silva has over 35 peer reviewed publications in the field of computer science mostly under the subfields of Natural Language Processing and Artificial Intelligence, earning more than 200 citations. At UO, he is working on semantic oppositeness measurement.

Dejing Dou is a Professor in the Computer and Information Science Department at the University of Oregon and leads the Advanced Integration and Mining (AIM) Lab. He is also the Director of the NSF IUCRC Center for Big Learning (CBL). He received his bachelor degree from Tsinghua University, China in 1996 and his Ph.D. degree from Yale University in 2004. His research areas include artificial intelligence, data mining, data integration, information extraction, and health informatics. Dejing Dou has published more than 100 research papers, some of which appear in prestigious conferences and journals like AAAI, IJCAI, ICML, ICLR, KDD, ICDM, ACL, EMNLP, CIKM, ISWC, JIIS and JoDS. His DEXA'15 paper received the best paper award. His KDD'07 paper was nominated for the best research paper award. He is on the Editorial Boards of Journal on Data Semantics, Journal of Intelligent Information Systems, and PLOS ONE. He has been serving as program committee members for various international conferences and as program co-chairs for five of them.