The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

# Open-World Knowledge Graph Completion

**Baoxu Shi, Tim Weninger**
University of Notre Dame
{bshi,tweninge}@nd.edu

## Abstract

Knowledge Graphs (KGs) have been applied to many tasks including Web search, link prediction, recommendation, natural language processing, and entity linking. However, most KGs are far from complete and are growing at a rapid pace. To address these problems, Knowledge Graph Completion (KGC) has been proposed to improve KGs by filling in its missing connections. Unlike existing methods which hold a closed-world assumption, *i.e.*, where KGs are fixed and new entities cannot be easily added, in the present work we relax this assumption and propose a new open-world KGC task. As a first attempt to solve this task we introduce an open-world KGC model called ConMask. This model learns embeddings of the entity's name and parts of its text-description to connect unseen entities to the KG. To mitigate the presence of noisy text descriptions, ConMask uses a relationship-dependent content masking to extract relevant snippets and then trains a fully convolutional neural network to fuse the extracted snippets with entities in the KG. Experiments on large data sets, both old and new, show that ConMask performs well in the open-world KGC task and even outperforms existing KGC models on the standard closed-world KGC task.

## Introduction

Knowledge Graphs (KGs) are a special type of information network that represents knowledge using RDF-style triples $\langle h,\ r,\ t\rangle$, where $h$ represents some head entity and $r$ represents some relationship that connects $h$ to some tail entity $t$. In this formalism a statement like "*Springfield is the capital of Illinois*" can be represented as $\langle$Springfield, capitalOf, Illinois$\rangle$. Recently, a variety of KGs, such as DBPedia (Lehmann et al. 2015), and ConceptNet (Speer, Chin, and Havasi 2017), have been curated in the service of fact checking (Shi and Weninger 2016), question answering (Lukovnikov et al. 2017), entity linking (Hachey et al. 2013), and for many other tasks (Nickel et al. 2016). Despite their usefulness and popularity, KGs are often noisy and incomplete. For example, DBPedia, which is generated from Wikipedia's infoboxes, contains $4.6$ million entities, but half of these entities contain less than 5 relationships.

Based on this observation, researchers aim to improve the accuracy and reliability of KGs by predicting the existence (or probability) of relationships. This task is often called Knowledge Graph Completion (KGC). Continuing the example from above, suppose the relationship capitalOf is missing between Indianapolis and Indiana; the KGC task might predict this missing relationship based on the topological similarity between this part of the KG and the part containing Springfield and Illinois.

Progress in vector embeddings originating with word2vec has produced major advancements in the KGC task. Typical embedding-based KGC algorithms like TransE (Bordes et al. 2013) and others learn low-dimensional representations (*i.e.*, embeddings) for entities and relationships using topological features. These models are able to predict the existence of missing relationships thereby "completing" the KG.

Existing KGC models implicitly operate under the *Closed-World Assumption* (Reiter 1978) in which all entities and relationships in the KG cannot be changed – only discovered. We formally define the Closed-word KGC task as follows:

**Definition 1** *Given an incomplete Knowledge Graph* $\mathcal{G} = (\mathbf{E}, \mathbf{R}, \mathbf{T})$*, where* $\mathbf{E}$*,* $\mathbf{R}$*, and* $\mathbf{T}$ *are the entity set, relationship set, and triple set respectively,* ***Closed-World Knowledge Graph Completion*** *completes* $\mathcal{G}$ *by finding a set of missing triples* $\mathbf{T}' = \{\langle h, r, t\rangle | h \in \mathbf{E}, r \in \mathbf{R}, t \in \mathbf{E}, \langle h, r, t\rangle \notin \mathbf{T}\}$ *in the incomplete Knowledge Graph* $\mathcal{G}$*.*

Closed-world KGC models heavily rely on the connectivity of the existing KG and are best able to predict relationships between existing, well-connected entities. Unfortunately, because of their strict reliance on the connectivity of the existing KG, closed-world KGC models are unable to predict the relationships of poorly connected or new entities. Therefore, we assess that closed-world KGC is most suitable for fixed or slowly evolving KGs.

However, most real-world KGs evolve quickly with new entities and relationships being added by the minute. For example, in the 6 months between DBPedia's October 2015 release and its April 2016 release $36,340$ new English entities were added – a rate of $200$ new entities per day. Recall that DBPedia merely tracks changes to Wikipedia infoboxes, so these updates do not include newly added articles without valid infobox data. Because of the accelerated growth of online information, repeatedly re-training closed-world models every day (or hour) has become impractical.

In the present work we borrow the idea of open-world assumption from probabilistic database literature (Ceylan, Darwiche, and Van den Broeck 2016) and relax the closed-world assumption to develop an *Open-World Knowledge Graph Completion* model capable of predicting relationships involving unseen entities or those entities that have only a few connections. Formally we define the open-world KGC task as follows:

**Definition 2** *Given an incomplete Knowledge Graph* $\mathcal{G} = (\mathbf{E}, \mathbf{R}, \mathbf{T})$*, where* $\mathbf{E}$*,* $\mathbf{R}$*, and* $\mathbf{T}$ *are the entity set, relationship set, and triple set respectively,* ***Open-World Knowledge Graph Completion*** *completes* $\mathcal{G}$ *by finding a set of missing triples* $\mathbf{T}' = \{\langle h, r, t\rangle | \langle h, r, t\rangle \notin \mathbf{T}, h \in \mathbf{E}^i, t \in \mathbf{E}^i, r \in \mathbf{R}\}$ *in the incomplete Knowledge Graph* $\mathcal{G}$ *where* $\mathbf{E}^i$ *is an entity superset.*

In Defn. 2 we relax the constraint on the triple set $\mathbf{T}'$ so that triples in $\mathbf{T}'$ can contain entities that are absent from the original entity set $\mathbf{E}$.

Closed-world KGC models learn entity and relationship embedding vectors by updating an initially random vector based on the KG's topology. Therefore, any triple $\langle h, r, t\rangle \in \mathbf{T}'$ such that $h \notin \mathbf{E}$ or $t \notin \mathbf{E}$ will only ever be represented by its initial random vector because its absence does not permit updates from any inference function. In order to predict the missing connections for unseen entities, it is necessary to develop alternative features to replace the topological features used by closed-world models.

Text content is a natural substitute for the missing topological features of disconnected or newly added entities. Indeed, most KGs such as FreeBase (Bollacker et al. 2008), DBPedia (Lehmann et al. 2015), and SemMedDB (Kilicoglu et al. 2012) were either directly extracted from (Lin et al. 2016; Ji and Grishman 2011), or are built in parallel to some underlying textual descriptions. However, open-world KGC differs from the standard information extraction task because 1) Rather than extracting triples from a large text corpus, the goal of open-world KGC is to discover missing relationships; and 2) Rather than a pipeline of independent subtasks like Entity Linking (Francis-Landau, Durrett, and Klein 2016) and Slotfilling (Liu and Lane 2016), etc., open-world KGC is a holistic task that operates as a single model.

Although it may seem intuitive to simply include an entity's description into an existing KGC model, we find that learning useful vector embeddings from unstructured text is much more challenging than learning topology-embeddings as in the closed-world task. First, in closed-world KGC models, each entity will have a unique embedding, which is learned from its directly connected neighbors; whereas open-world KGC models must fuse entity embeddings with the word embeddings of the entity's description. These word embeddings must be updated by entities sharing the same words regardless of their connectivity status. Second, because of the inclusion of unstructured content, open-world models are likely to include noisy or redundant information.

With respect to these challenges, the present work makes the following contributions:

1. We describe an open-world KGC model called **ConMask** that uses relationship-dependent content masking to reduce noise in the given entity description and uses fully convolutional neural networks (FCN) to fuse related text into a relationship-dependent entity embedding.
2. We release two new Knowledge Graph Completion data sets constructed from DBPedia and Wikipedia for use in closed-world and open-world KGC evaluation.

Before introduce the ConMask model, we first present preliminary material by describing relevant KGC models. Then we describe the methodology, data sets, and a robust case study of closed-world and open-world KGC tasks. Finally, we draw conclusions and offer suggestions for future work.

## Closed-World Knowledge Graph Completion

A variety of models have been developed to solve the closed-world KGC task. The most fundamental and widely used model is a translation-based Representation Learning (RL) model called TransE (Bordes et al. 2013). TransE assumes there exists a simple function that can translate the embedding of the head entity to the embedding of some tail entity via some relationship:

$$\mathbf{h} + \mathbf{r} = \mathbf{t}, \tag{1}$$

where $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ are embeddings of head entity, relationship, and tail entity respectively. Based on this function, many other KGC models improve the expressive power of Eq. 1 by introducing more relationship-dependent parameters. TransR (Lin et al. 2015), for example, augments Eq. 1 to $\mathbf{h}\mathbf{M}_r + \mathbf{r} = \mathbf{t}\mathbf{M}_r$ where $\mathbf{M}_r$ is a relationship-dependent entity embedding transformation.

In order to train the KGC models, TransE defines an energy-based loss function as

$$\mathcal{L}(\mathbf{T}) = \Sigma_{\langle h,r,t\rangle \in \mathbf{T}}[\gamma + E(\langle h, r, t\rangle) - E(\langle h', r', t'\rangle)]_+, \tag{2}$$

where the energy function $E(\langle h, r, t\rangle) = \| \mathbf{h} + \mathbf{r} - \mathbf{t} \|_{L_n}$ measures the closeness of the given triple, $\langle h, r, t\rangle$ is some triple that exists in the triple set $\mathbf{T}$ of an incomplete KG $\mathcal{G}$, and $\langle h', r', t'\rangle$ is a "corrupted" triple derived by randomly replacing one part of $\langle h, r, t\rangle$ so that it does not exist in $\mathbf{T}$.

In other recent work, ProjE (Shi and Weninger 2017) considered closed-world KGC to be a type of ranking task and applied a list-wise ranking loss instead of Eq. 2. Other closed-world models such as PTransE (Lin, Liu, and Sun 2015) and dORC (Zhang 2017) maintain a simple translation function and use complex topological features like extended-length paths and "one-relation-circle" structures to improve predictive performance.

Unlike topology-based models, which have been studied extensively, there has been little work that utilizes text information for KGC. Neural Tensor Networks (NTN) (Socher et al. 2013) uses the averaged word embedding of an entity to initialize the entity representations. DKRL (Xie et al. 2016) uses the combined distance between topology-embeddings and text-embeddings as its energy function. Jointly (Xu et al. 2016) combines the topology-embeddings and text-embeddings first using a weighted sum and then calculates

the $L_n$ distance between the translated head entity and tail entity. However, gains in predictive performance from these joint-learning models are rather small compared to advances in topology-based models.

Furthermore, the aforementioned models are all closed-world KGC models, which can only learn meaningful representations for entities that are present during training and are well connected within the KG. These models have no mechanism by which new entities can be connected with the existing KG as required in open-world KGC.

In the present work, we present an open-world KGC model called ConMask that uses primarily text features to learn entity and relationship embeddings. Compared to topology-based and joint-learning models, ConMask can generate representations for unseen entities if they share the same vocabulary with entities seen during training. To properly handle one-to-many and many-to-one relationships, we also apply a relationship-dependent content masking layer to generate entity embeddings.

## ConMask: A Content Masking Model for Open-World KGC

In this section we describe the architecture and the modelling decisions of the ConMask model. To illustrate how this model works, we begin by presenting an actual example as well as the top-ranked target entity inferred by the ConMask model:

**Example Task**: Complete triple ⟨Ameen Sayani, residence, ? ⟩, where Ameen Sayani is absent from the KG.

**Snippet of Entity Description**: "... ***Ameen Sayani*** *was introduced to All India Radio,* ***Bombay****, by his brother Hamid Sayani. Ameen participated in English programmes there for ten years ...*".

**Predicted Target Entity**: Mumbai.

In this example, if a human reader were asked to find the residence of Ameen Sayani, a popular radio personality in India, from the entity description, then the human reader is unlikely to read the entire text from beginning to end. Instead, the reader might skim the description looking for contextual clues such as family or work-related information. Here, Ameen's workplace All India Radio is located in Bombay, so the human reader may infer that Ameen is a resident of Bombay. A human reader may further reason that because Bombay has recently changed its name to Mumbai, then Mumbai would be the (correct) target entity.

Here and throughout the present work, we denote the missing entity as the *target* entity, which can be either the head or the tail of a triple.

We decompose the reasoning process described above into three steps: 1) Locating information relevant to the task, 2) Implicit reasoning based on the context and the relevant text, and 3) Resolving the relevant text to the proper target entity. The ConMask model is designed to mimic this process. Thus, ConMask consists of three components:

1. Relationship-dependent content masking, which highlights words that are relevant to the task,
2. Target fusion, which extracts a target entity embedding from the relevant text, and

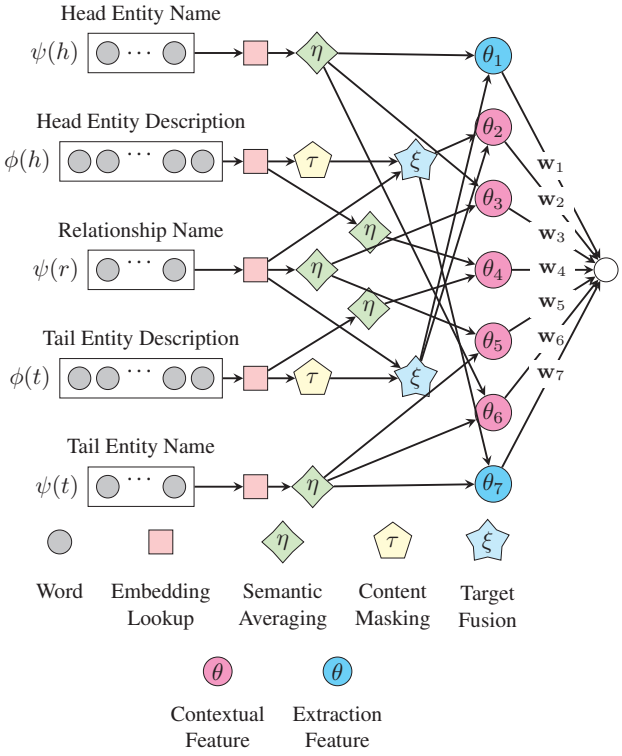


Figure 1: Illustration of the ConMask model for Open-World Knowledge Graph Completion.

3. Target entity resolution, which chooses a target entity by computing a similarity score between target entity candidates in the KG, the extracted entity embeddings, and other textual features.

ConMask selects words that are related to the given relationship to mitigate the inclusion of irrelevant and noisy words. From the relevant text, ConMask then uses fully convolutional network (FCN) to extract word-based embeddings. Finally, it compares the extracted embeddings to existing entities in the KG to resolve a ranked list of target entities. The overall structure of ConMask is illustrated in Fig. 1. Later subsections describe the model in detail.

### Relationship-Dependent Content Masking

In open-world KGC, we cannot rely solely on the topology of the KG to guide our model. Instead, it is natural to consider extracting useful information from text in order to infer new relationships in the KG. The task of extracting relationships among entities from text is often called relation extraction (Mintz et al. 2009). Recent work in this area tends to employ neural networks such as CNN (Xu et al. 2016) or abstract meaning representations (AMRs) (Huang et al. 2017) to learn a unified kernel to remove noise and extract the relationship-agnostic entity representations. For open-world KGC, it may be possible to create a model with relationship-dependent CNN kernels. But this type of model would significantly increase the number of parameters and may overfit on rare relationships.

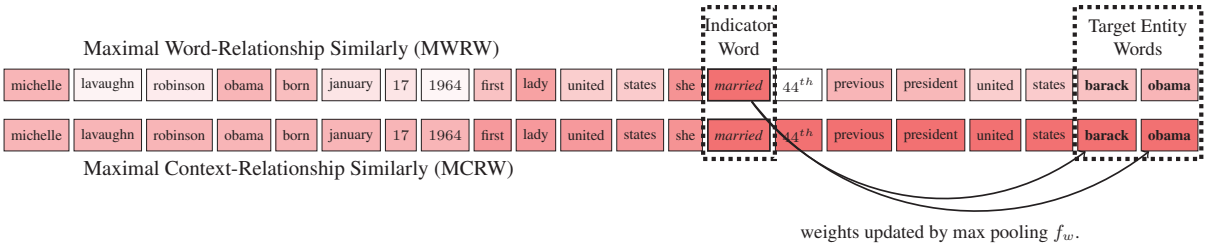


Figure 2: Relationship-dependent Content Masking heat map for the description of Michelle Obama given relationship type spouse. Stop-words are removed. Higher weights show in darker color.

In the proposed ConMask model we developed an alternative approach called *relationship-dependent content masking*. The goal is to pre-process the input text in order to select small relevant snippets based on the given relationship – thereby masking irrelevant text. The idea of content masking is inspired by the attention mechanism used by recurrent neural network (RNN) models (Chorowski et al. 2015), which is widely applied to NLP tasks. In a typical attention-based RNN model, each output stage of a recurrent cell is assigned an attention score.

In ConMask, we use a similar idea to select the most related words given some relationship and mask irrelevant words by assigning a relationship-dependent similarity score to words in the given entity description. We formally define relationship-dependent content masking as:

$$\tau(\phi(e), \psi(r)) = \mathbf{W}_{\phi(e)} \circ f_w(\mathbf{W}_{\phi(e)}, \mathbf{W}_{\psi(r)}), \quad (3)$$

where $e$ is an entity, $r$ is some relationship, $\phi$ and $\psi$ are the description and name mapping functions respectively that return a word vector representing the description or the name of an entity or relationship. $\mathbf{W}_{\phi(e)} \in \mathbb{R}^{|\phi(e)| \times k}$ is the description matrix of $e$ in which each row represents a $k$ dimensional embedding for a word in $\phi(e)$ in order, $\mathbf{W}_{\psi(r)} \in \mathbb{R}^{|\psi(r)| \times k}$ is the name matrix of $r$ in which each row represents a $k$ dimensional embedding for a word in the title of relationship $\psi(r)$, $\circ$ is row-wise product, and $f_w$ calculates the masking weight for each row, *i.e.*, the embedding of each word, in $\mathbf{W}_{\phi(e)}$.

The simplest way to generate these weights is by calculating a similarity score between each word in entity description $\phi(e)$ and the words in relationship name $\psi(r)$. We call this simple function Maximal Word-Relationship Weights (MWRW) and define it as:

$$f_w^{\text{MWRW}}\left(\mathbf{W}_{\phi(e)}, \mathbf{W}_{\psi(r)}\right)_{[i]} = \max_j \left( \frac{\sum_m^k \mathbf{W}_{\phi(e)[i,m]} \mathbf{W}_{\psi(r)[j,m]}}{\sqrt{\sum_m^k \mathbf{W}^2_{\phi(e)[i,m]}} \sqrt{\sum_m^k \mathbf{W}^2_{\psi(r)[j,m]}}} \right), \quad (4)$$

where the weight of the $i^{\text{th}}$ word in $\phi(e)$ is the largest cosine similarity score between the $i^{\text{th}}$ word embedding in $\mathbf{W}_{\phi(e)}$ and the word embedding matrix of $\psi(r)$ in $\mathbf{W}_{\psi(r)}$.

This function assigns a lower weight to words that are not relevant to the given relationship and assigns higher scores to the words that appear in the relationship or are semantically similar to the relationship. For example, when inferring the target of the partial triple ⟨Michelle Obama, AlmaMater, ?⟩, MWRW will assign high weights to words like *Princeton*, *Harvard*, and *University*, which include the words that describe the target of the relationship. However the words that have the highest scores do not always represent the actual target but, instead, often represent words that are similar to the relationship name itself. A counter-example is shown in Fig. 2, where, given the relationship spouse, the word with the highest MWRW score is *married*. Although spouse is semantically similar to *married*, it does not answer the question posed by the partial triple. Instead, we call words with high MWRW weights *indicator words* because the correct target-words are usually located nearby. In the example-case, we can see that the correct target *Barack Obama* appears after the indicator word *married*.

In order to assign the correct weights to the target words, we improve the content masking by using Maximal Context-Relationship Weights (MCRW) to adjust the weights of each word based on its context:

$$f_w\left(\mathbf{W}_{\phi(e)}, \mathbf{W}_{\psi(r)}\right)_{[i]} = \max\left(f_w^{\text{MWRW}}\left(\mathbf{W}_{\phi(e)}, \mathbf{W}_{\psi(r)}\right)_{[i-k_m:i]}\right), \quad (5)$$

in which the weight of the $i^{th}$ word in $\phi(e)$ equals the maximum MWRW score of the $i^{th}$ word itself and previous $k_m$ words. From a neural network perspective, the re-weighting function $f_w$ can also be viewed as applying a row-wise max reduction followed by a 1-D max-pooling with a window size of $k_m$ on the matrix product of $\mathbf{W}_{\phi(e)}$ and $\mathbf{W}^T_{\psi(r)}$.

To recap, the relationship-dependent content masking process described here assigns importance weights to words in an entity's description based on the similarity between each word's context and the given relationship. After non-relevant content is masked, the model needs to learn a single embedding vector from the masked content matrix to compare with the embeddings of candidate target entities.

## Target Fusion

Here we describe how ConMask extracts word-based entity embeddings. We call this process the *target fusion* function $\xi$, which distills an embedding using the output of Eq. 3.

Initially, we looked for solutions to this problem in recurrent neural networks (RNNs) of various forms. Despite their popularity in NLP-related tasks, recent research has found that RNNs are not good at performing "extractive"

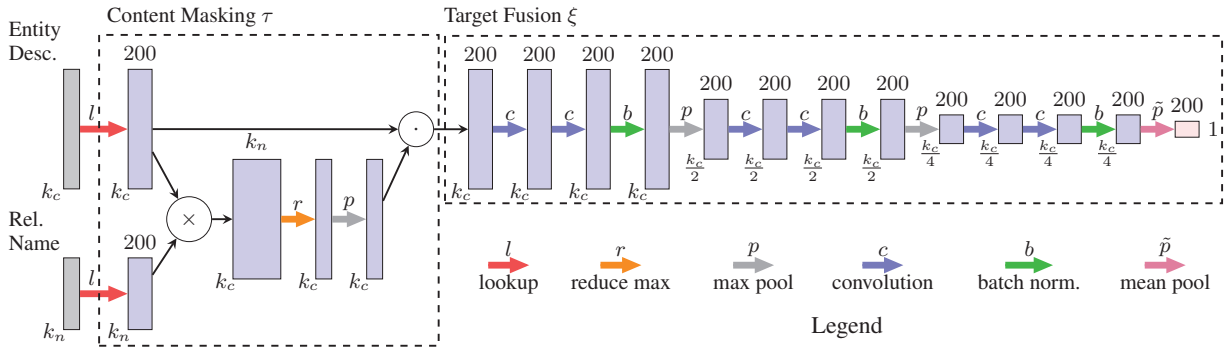


Figure 3: Architecture of the target fusion and relationship-dependent content masking process in ConMask. $k_c$ is the length of the entity description and $k_n$ is the length of the relationship name. This figure is best viewed in color.

tasks (See, Liu, and Manning 2017). RNNs do not work well in our specific setting because the input of the Target Fusion is a masked content matrix, which means most of the stage inputs would be zero and hence hard to train.

In this work we decide to use fully convolutional neural network (FCN) as the target fusion structure. A CNN-based structure is well known for its ability to capture peak values using convolution and pooling. Therefore FCN is well suited to extract useful information from the weighted content matrix. Our adaptation of FCNs yields the target fusion function $\xi$, which generates a $k$-dimensional embedding using the output of content masking $\tau(\phi(e), \psi(r))$ where $e$ is either a head or tail entity from a partial triple.

Figure 3 shows the overall architecture of the target fusion process and its dependent content masking process. The target fusion process has three FCN layers. In each layer, we first use two 1-D convolution operators to perform affine transformation, then we apply $sigmoid$ as the activation function to the convoluted output followed by batch normalization (Ioffe and Szegedy 2015) and max-pooling. The last FCN layer uses mean-pooling instead of max-pooling to ensure the output of the target fusion layer always return a single $k$-dimensional embedding.

Note that the FCN used here is different from the one that typically used in computer vision tasks (Chen et al. 2016). Rather than reconstructing the input, as is typical in CV, the goal of target fusion is to extract the embedding w.r.t given relationship, therefore we do not have the de-convolution operations. Another difference is that we reduce the number of embeddings by half after each FCN layer but do not increase the number of channels, *i.e.*, the embedding size. This is because the input weighted matrix is a sparse matrix with a large portion of zero values, so we are essentially fusing peak values from the input matrix into a single embedding representing the target entity.

## Semantic Averaging

Although it is possible to use target fusion to generate all entity embeddings used in ConMask, such a process would result in a large number of parameters. Furthermore, because the target fusion function is an extraction function it would be odd to apply it to entity names where no extraction is necessary. So, we also employ a simple *semantic averaging* function $\eta(\mathbf{W}) = \frac{1}{k_l}\Sigma_i^{k_l}\mathbf{W}_{[i,:]}$ that combines word embeddings to represent entity names and for generating background representations of other textual features, where $\mathbf{W} \in \mathcal{R}^{k_l \times k}$ is the input embedding matrix from the entity description $\phi(\cdot)$ or the entity or relationship name $\psi(\cdot)$.

To recap: at this point in the model we have generated entity embeddings through the content masking and target fusion operations. The next step is to define a loss function that finds one or more entities in the KG that most closely match the generated embedding.

## Loss Function

To speed up the training and take to advantage of the performance boost associated with a list-wise ranking loss function (Shi and Weninger 2017), we designed a partial list-wise ranking loss function that has both positive and negative target sampling:

$$\mathcal{L}(h,r,t) = \begin{cases} \sum\limits_{h_+ \in E^+} -\frac{\log(S(h_+,r,t,E^+ \cup E^-))}{|E^+|}, p_c > 0.5 \\ \sum\limits_{t_+ \in E^+} -\frac{\log(S(h,r,t_+,E^+ \cup E^-))}{|E^+|}, p_c \leq 0.5 \end{cases}, \tag{6}$$

where $p_c$ is the corruption probability drawn from an uniform distribution $U[0,1]$ such that when $p_c > 0.5$ we keep the input tail entity $t$ but do positive and negative sampling on the head entity and when $p_c \leq 0.5$ we keep the input head entity $h$ intact and do sampling on the tail entity. $E^+$ and $E^-$ are the sampled positive and negative entity sets drawn from the positive and negative target distribution $P_+$ and $P_-$ respectively. Although a type-constraint or frequency-based distribution may yield better results, here we follow the convention and simply apply a simple uniform distribution for both $P_+$ and $P_-$. When $p_c \leq 0.5$, $P_+$ is a uniform distribution of entities in $\{t_+ | \langle h, r, t_+ \rangle \in \mathbf{T}\}$ and $P_-$ is an uniform distribution of entities in $\{t_- | \langle h, r, t_- \rangle \notin \mathbf{T}\}$. On the other hand when $p_c > 0.5$, $P_+$ is an uniform distribution of entities in $\{h_+ | \langle h_+, r, t \rangle \in \mathbf{T}\}$ and $P_-$ is an uniform distribution of entities in $\{h_- | \langle h_-, r, t \rangle \notin \mathbf{T}\}$. The function $S$ in Eq. 6 is the softmax normalized output of ConMask:

Table 1: Open-world Entity prediction results on DBPedia50k and DBPedia500k. For Mean Rank (MR) lower is better. For HITS@10 and Mean Reciprocal Rank (MRR) higher is better.

| | DBPedia50k | | | | | | DBPedia500k | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Head | | | Tail | | | Head | | | Tail | | |
| Model | MR | HITS@10 | MRR | MR | HITS@10 | MRR | MR | HITS@10 | MRR | MR | HITS@10 | MRR |
| Target Filtering Baseline | 605 | 0.07 | 0.07 | 104 | 0.23 | 0.11 | 20667 | 0.01 | 0.01 | 3480 | 0.02 | 0.01 |
| Semantic Averaging | 513 | 0.11 | 0.12 | 93 | 0.29 | 0.15 | 8144 | 0.04 | 0.09 | 1002 | 0.16 | 0.13 |
| DKRL (2-layer CNN) | 490 | 0.09 | 0.08 | 70 | 0.40 | 0.23 | 19776 | 0.01 | 0.01 | 2275 | 0.04 | 0.04 |
| **ConMask** | **95** | **0.39** | **0.35** | **16** | **0.81** | **0.61** | **2877** | **0.17** | **0.33** | **165** | **0.52** | **0.47** |

$$S(h,r,t,E^{\pm}) = \begin{cases} \frac{\exp(\text{ConMask}(h,r,t))}{\sum_{e\in E^{\pm}} \exp(\text{ConMask}(e,r,t))}, p_c > 0.5 \\ \frac{\exp(\text{ConMask}(h,r,t))}{\sum_{e\in E^{\pm}} \exp(\text{ConMask}(h,r,e))}, p_c \leq 0.5 \end{cases} . \tag{7}$$

Note that Eq. 6 is actually a generalized form of the sampling process used by most existing KGC models. When $|E_+| = 1$ and $|E_-| = 1$, the sampling method described in Eq. 6 is the same as the triple corruption used by TransE (Bordes et al. 2013), TransR (Lin et al. 2015), TransH (Wang et al. 2014), and many other closed-world KGC models. When $|E_+| = |\{t|\langle h,r,t\rangle \in \mathbf{T}\}|$, which is the number of all true triples given a partial triple $\langle h, r, ?\rangle$, Eq. 6 is the same as ProjE_listwise (Shi and Weninger 2017).

## Experiments

The previous section described the design decisions and modelling assumptions of ConMask. In this section we present the results of experiments performed on old and new data sets in both open-world and closed-world KGC tasks.

### Settings

Training parameters were set empirically but without fine-tuning. We set the word embedding size $k = 200$, maximum entity content and name length $k_c = k_n = 512$. The word embeddings are from the publicly available pre-trained 200-dimensional GloVe embeddings (Pennington, Socher, and Manning 2014). The content masking window size $k_m = 6$, number of FCN layers $k_{fcn} = 3$ where each layer has 2 convolutional layers and a BN layer with a moving average decay of 0.9 followed by a dropout with a keep probability $p = 0.5$. Max-pooling in each FCN layer has a pool size and stride size of 2. The mini-batch size used by ConMask is $k_b = 200$. We use Adam as the optimizer with a learning rate of $10^{-2}$. The target sampling set sizes for $|E_+|$ and $|E_-|$ are 1 and 4 respectively. All open-world KGC models were run for at most 200 epochs. All compared models used their default parameters.

ConMask is implemented in TensorFlow. The source code is available at https://github.com/bxshi/ConMask.

### Data Sets

The Freebase 15K (FB15k) data set is widely used in KGC. But FB15k is fraught with reversed- or synonym-triples (Toutanova and Chen 2015) and does not provide sufficient textual information for content-based KGC methods to use. Due to the limited text content and the redundancy found in the FB15K data set, we introduce two new data sets DBPedia50k and DBPedia500k for both open-world and closed-world KGC tasks. Statistics of all data sets are shown in Tab. 2.

Table 2: Data set statistics.

| | | | Triples | | |
|---|---|---|---|---|---|
| Data set | Entities | Rel. | Train | Validation | Test |
| FB15k | 14,951 | 1,345 | 483,142 | 50,000 | 59,071 |
| FB20k | 19,923 | 1,345 | 472,860 | 48,991 | 90,149 |
| DBPedia50k | 49,900 | 654 | 32,388 | 399 | 10,969 |
| DBPedia500k | 517,475 | 654 | 3,102,677 | 10,000 | 1,155,937 |

The methodology used to evaluate the open-world and closed-world KGC tasks is similar to the related work. Specifically, we randomly selected 90% of the entities in the KG and induced a KG subgraph using the selected entities, and from this reduced KG, we further removed 10% of the relationships, *i.e.*, graph-edges, to create $\text{KG}_{\text{train}}$. All other triples not included in $\text{KG}_{\text{train}}$ are held out for the test set.

### Open-World Entity Prediction

For the open-world KGC task, we generated a test set from the 10% of entities that were held out of $\text{KG}_{\text{train}}$. This held out set has relationships that connect the test entities to the entities in $\text{KG}_{\text{train}}$. So, given a held out entity-relationship partial triple (that was not seen during training), our goal is to predict the correct target entity within $\text{KG}_{\text{train}}$.

To mitigate the excessive cost involved in computing scores for all entities in the KG, we applied a target filtering method to all KGC models. Namely, for a given partial triple $\langle h, r, ?\rangle$ or $\langle ?, r, t\rangle$, if a target entity candidate has not been connected via relationship $r$ before in the training set, then it is skipped, otherwise we use the KGC model to calculate the actual ranking score. Simply put, this removes relationship-entity combinations that have never before been seen and are likely to represent nonsensical statements. The experiment results are shown in Tab. 1.

As a naive baseline we include the target filtering baseline method in Tab. 1, which assigns random scores to all the entities that pass the target filtering. Semantic Averaging is a simplified model which uses contextual features only. DKRL is a two-layer CNN model that generates entity embeddings with entity description (Xie et al. 2016). We implemented DKRL ourselves and removed the structural-related features so it can work under open-world KGC settings.

We find that the extraction features in ConMask do boost

Table 3: Closed-world KGC on head and tail prediction. For HITS@10 higher is better. For Mean Rank (MR) lower is better.

| | FB15k | | | | DBPedia50k | | | | DBPedia500k | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Head | | Tail | | Head | | Tail | | Head | | Tail | |
| Model | MR | HITS@10 | MR | HITS@10 | MR | HITS@10 | MR | HITS@10 | MR | HITS@10 | MR | HITS@10 |
| TransE | 189 | 0.68 | 92 | 0.75 | 2854 | 0.37 | 734 | 0.68 | 10034 | 0.15 | 2472 | **0.45** |
| TransR | 186 | **0.71** | 87 | **0.77** | 2689 | 0.39 | 718 | 0.67 | - | - | - | - |
| **ConMask** | **116** | 0.62 | **80** | 0.62 | **1063** | **0.41** | **141** | **0.72** | **1512** | **0.21** | **1568** | 0.20 |

Table 4: Entity prediction results on DBPedia50k data set. Top-3 predicted tails are shown with the correct answer in bold.

| Head | Relationship | Predicted Tails |
|---|---|---|
| Chakma language | languageFamily | **Indo-Aryan language**, Hajong language, Language |
| Gabrielle Stanton | notableWork | Star Trek: Deep Space Nine, A Clear and Present Danger, Pilot (Body of Proof) |
| MiniD | influencedBy | **Lua**, Delphi, MorphOS |
| The Time Machine (1960 film) | writer | Writer, **David Duncan**, Jeff Martin |

mean rank performance by at least 60% on both data sets compared to the extraction-free Semantic Averaging. Interestingly, the performance boost on the larger DBPedia500k data set is more significant than the smaller DBPedia50k, which indicates that the extraction features are able to find useful textual information from the entity descriptions.

### Closed-World Entity Prediction

Because the open-world assumption is less restrictive than the closed-world assumption, it is possible for ConMask to perform closed-world tasks, even though it was not designed to do so. So in Tab. 3 we also compare the ConMask model with other closed-world methods on the standard FB15k data set as well as the two new data sets. Results from TransR are missing from the DBPedia500k data set because the model did not complete training after 5 days.

We find that ConMask sometimes outperforms closed-world methods on the closed-world task. ConMask especially shows improvement on the DBPedia50k data set; this is probably because the random sampling procedure used to create DBPedia50k generates a sparse graph. closed-world KGC models, which rely exclusively on structural features, have a more difficult time with sub-sampled KGs.

### Discussion

In this section we elaborate on some actual prediction results and show examples that highlight the strengths and limitations of the ConMask model.

Table 4 shows 4 KGC examples. In each case, ConMask was provided the head and the relationship and asked to predict the tail entity. In most cases ConMask successfully ranks the correct entities within the top-3 results. Gabrielle Stanton's notableWork is an exception. Although Stanton did work on Star Trek, DBPedia indicates that her most notable work is actually The Vampire Diaries, which ranked 4th. The reason for this error is because the indicator word for The Vampire Diaries was "consulting producer", which was not highly correlated to the relationship name "notable work" from the model's perspective.

Another interesting result was the prediction given from the partial triple ⟨The Time Machine, writer, ?⟩. The ConMask model ranked the correct screenwriter David Duncan as the 2nd candidate, but the name "David Duncan" does not actually appear in the film's description. Nevertheless, the ConMask model was able to capture the correct relationship because the words "The Time Machine" appeared in the description of David Duncan as one of his major works.

Although ConMask outperforms other KGC models on metrics such as Mean Rank and MRR, it still has some limitations and room for improvement. First, due to the nature of the relationship-dependent content masking, some entities with names that are similar to the given relationships, such as the Language-entity in the results of the languageFamily-relationship and the Writer-entity in the results of the writer-relationship, are ranked with a very high score. In most cases the correct target entity will be ranked above relationship-related entities. Yet, these entities still hurt the overall performance. It may be easy to apply a filter to modify the list of predicted target entities so that entities that are same as the relationship will be rearranged. We leave this task as a matter for future work.

## Conclusion and Future Work

In the present work we introduced a new open-world Knowledge Graph Completion model ConMask that uses relationship-dependent content masking, fully convolutional neural networks, and semantic averaging to extract relationship-dependent embeddings from the textual features of entities and relationships in KGs. Experiments on both open-world and closed-world KGC tasks show that the ConMask model has good performance in both tasks. Because of problems found in the standard KGC data sets, we also released two new DBPedia data sets for KGC research and development.

The ConMask model is an extraction model which currently can only predict relationships if the requisite information is expressed in the entity's description. The goal for future work is to extend ConMask with the ability to find new or implicit relationships.

## References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD/PODS*, 1247–1250. ACM.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*, 2787–2795.

Ceylan, I. I.; Darwiche, A.; and Van den Broeck, G. 2016. Open-world probabilistic databases. In *Description Logics*.

Chen, H.; Qi, X.; Cheng, J.-Z.; and Heng, P.-A. 2016. Deep contextual networks for neuronal structure segmentation. In *AAAI*, 1167–1173.

Chorowski, J. K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-based models for speech recognition. In *NIPS*, 577–585.

Francis-Landau, M.; Durrett, G.; and Klein, D. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. *arXiv preprint arXiv:1604.00734*.

Hachey, B.; Radford, W.; Nothman, J.; Honnibal, M.; and Curran, J. R. 2013. Evaluating entity linking with wikipedia. *AI* 194:130–150.

Huang, L.; May, J.; Pan, X.; Ji, H.; Ren, X.; Han, J.; Zhao, L.; and Hendler, J. A. 2017. Liberal entity extraction: Rapid construction of fine-grained entity typing systems. *Big Data* 5(1):19–31.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Ji, H., and Grishman, R. 2011. Knowledge base population: Successful approaches and challenges. In *ACL*, 1148–1158.

Kilicoglu, H.; Shin, D.; Fiszman, M.; Rosemblat, G.; and Rindflesch, T. C. 2012. SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics* 28(23):3158–3160.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web* 6(2):167–195.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*, 2181–2187.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *ACL*, 2124–2133.

Lin, Y.; Liu, Z.; and Sun, M. 2015. Modeling Relation Paths for Representation Learning of Knowledge Bases. *EMNLP* 705–714.

Liu, B., and Lane, I. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

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

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *ACL*, 1003–1011.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1):11–33.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, 1532–1543.

Reiter, R. 1978. On closed world data bases. In *Logic and data bases*. Springer. 55–76.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *ACL*.

Shi, B., and Weninger, T. 2016. Fact checking in heterogeneous information networks. In *WWW*, 101–102.

Shi, B., and Weninger, T. 2017. ProjE: Embedding projection for knowledge graph completion. In *AAAI*.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *NIPS*, 926–934.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5 - An Open Multilingual Graph of General Knowledge. *AAAI*.

Toutanova, K., and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *3rd Workshop on Continuous Vector Space Models and Their Compositionality*. ACL.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. *AAAI* 1112–1119.

Xie, R.; Liu, Z.; Jia, J.; Luan, H.; and Sun, M. 2016. Representation learning of knowledge graphs with entity descriptions. In *AAAI*, 2659–2665.

Xu, J.; Chen, K.; Qiu, X.; and Huang, X. 2016. Knowledge graph representation with jointly structural and textual encoding. *arXiv preprint arXiv:1611.08661*.

Zhang, W. 2017. Knowledge graph embedding with diversity of structures. In *WWW*, 747–753.