https://doi.org/10.7494/csci.2020.21.3.3701

Benyamin Ahmadnia
Bonnie J. Dorr
Parisa Kordjamshidi

KNOWLEDGE GRAPHS EFFECTIVENESS IN NEURAL MACHINE TRANSLATION IMPROVEMENT

Abstract *Maintaining semantic relations between words during the translation process yields more accurate target-language output from Neural Machine Translation (NMT). Although difficult to achieve from training data alone, it is possible to leverage Knowledge Graphs (KGs) to retain source-language semantic relations in the corresponding target-language translation. The core idea is to use KG entity relations as embedding constraints to improve the mapping from source to target. This paper describes two embedding constraints, both of which employ Entity Linking (EL)—assigning a unique identity to entities—to associate words in training sentences with those in the KG: (1) a monolingual embedding constraint that supports an enhanced semantic representation of the source words through access to relations between entities in a KG; and (2) a bilingual embedding constraint that forces entity relations in the source-language to be carried over to the corresponding entities in the target-language translation. The method is evaluated for English-Spanish translation exploiting* Freebase *as a source of knowledge. Our experimental results demonstrate that exploiting KG information not only decreases the number of unknown words in the translation but also improves translation quality.*

Keywords natural language processing, neural machine translation, knowledge graph representation

Citation Computer Science 21(3) 2020: 261–280

1. Introduction

Machine Learning (ML) has been the quintessential solution for many Artificial Intelligence (AI) problems. Nowadays, ML is centered around algorithms that are trained on available task-specific labeled and unlabeled training examples. Although learning paradigms like Transfer Learning [36] attempt to incorporate knowledge from one task into another, these techniques are limited in scalability and are specific to the task at hand. On the other hand, humans have the intrinsic ability to elicit required past knowledge from the world on demand and infuse it with newly learned concepts to solve problems [4].

Two major issues have emerged for current Neural Machine Translation (NMT) systems: (1) vocabulary size is limited by training data content, thus yielding many Out-Of-Vocabulary (OOV) cases [29]; and (2) current neural architectures [7, 19, 45, 50] only model parallel sentence relationships without any explicit attempt to leverage word-level relationships for disambiguation. Ideally, semantic relations and distinctions between words (e.g., *king, man* vs. *queen, woman*) are identified and maintained during the translation process, but NMT models do not currently support this functionality.

This paper demonstrates the viability of using entities and relations in existing Knowledge Graphs (KGs) [8] for NMT. KG knowledge is encoded as a triple that includes a *head* entity (e.g., *Barack Obama*), a relation (e.g., *president*), and a tail entity (e.g., *United States*). KGs bring external knowledge to bear so that semantic relationships between entities are gleaned in many Natural Language Processing (NLP) tasks [4] including MT [30, 33].

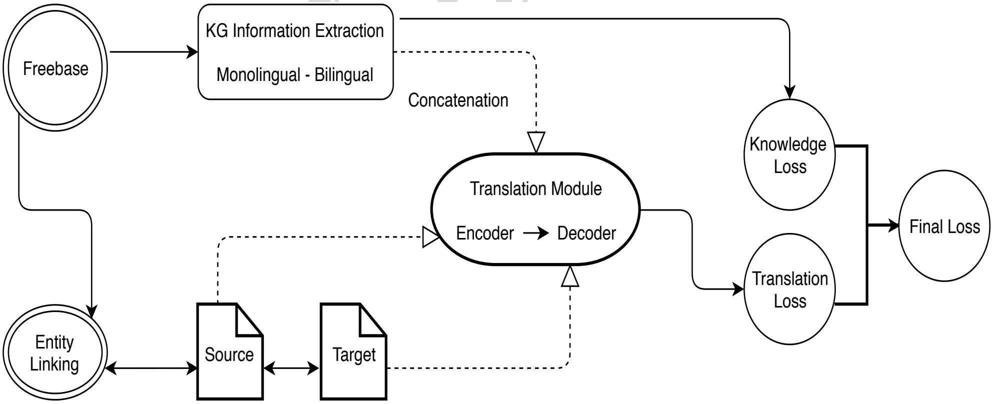

Figure 1. KG-based NMT model with monolingual and bilingual embedding constraints using entity linking system

As an illustration, consider the source sentence *Barack Obama took the presidential oath of office at the White House.* Translation into Spanish via a baseline NMT would be: *Barack Obama tomó juramento presidencial en el* <*UNK*> <*UNK*>, where <*UNK*> is an OOV indicator. Clearly this output is deficient in comparison to the corresponding human reference: *Barack Obama, hizo el juramento presidencial en la Casa Blanca*. We demonstrate that it is possible to use a KG to improve NMT output quality, assuming *Barack Obama* appears both in the source vocabulary and in the KG—even when the word *White House* does not appear in the source vocabulary. Specifically, we leverage knowledge about the entity *Barack Obama*, coupled with the KG's trained representation of its relationship to *White House*, to map to the corresponding Spanish term *Casa Blanca*.

KG information is represented in the form of embedding constraints, that is, during training, word embeddings are trained to support the mapping from source to target language while also satisfying KG requirements. The translation module and KG information extraction module are shown as two independent parts in Figure 1. Shown as two independent parts in Figure 1; however, these two interact through the concatenation of KG embedding vectors and translation module embedding vectors. These two modules yield the Knowledge Loss and Translation Loss that are adopted into monolingual embedding constraints and bilingual embedding constraints, respectively.

In this paper, we use Freebase as a source of information for NMT models and we employ TransR technique to learn entities' embeddings (represent entities and their relations). We also utilize Entity Linking (EL)—assigning a unique identity to entities—to align triples in the KG with the source sentences. Based on a monolingual embedding constraint, KG entity relations are used to influence the source side; this constraint forces the embedding of the source words to hold the semantic relations provided by the KG. Based on a bilingual embedding constraint, the relation equivalence between the source words and their corresponding translations is modeled. Thus, semantic relations between the source words are maintained throughout their translation. Both monolingual and bilingual embedding constraints are modeled during the training process to enable enrichment of the NMT system with KG information. We demonstrate for English-Spanish translation that this method achieves a higher quality translation than baselines and decreases the number of <*UNK*> tokens.

Generally, the Spanish language uses the Latin alphabet, with a few special letters, vowels with an acute accent *(á, ú, é, ó, í)*, *u* with an umlaut (*ü*), and an *n* with a tilde (*ñ*). Due to a number of reforms, the Spanish spelling system is almost perfectly phonemic and, therefore, easier to learn than the majority of languages. Spanish is pronounced phonetically, but includes the trilled *r* which is somewhat complex to reproduce. In the Spanish IPA, the letters *b* and *v* correspond to the same symbol *b* and the distinction only exists in regional dialects. The letter *h* is silent except in conjunction with *c*, *ch*, which changes the sound into *tf*. Spanish language punctuation is very close to English. There are a few significant differences. For example, in Spanish, exclaim and interrogative sentences are preceded by inverted question and exclamation marks. Also, in a Spanish conversation, a change in speakers is indicated by a dash, while in English, each speaker's remark is placed in separate paragraphs. Formal and informal translations address several different characteristics. Inflection, declination and grammatical gender are important features of Spanish language [1,3].

The remaining parts of this paper are organized as follows; Section 2 reviews the previous related work. In Section 3, we describe the methodology of the present work. The experimental details are provided in Section 4. In Section 5, we evaluate our experimental results. Finally, Section 6 presents conclusions and future work.

2. Related Work

To resolve issues of vocabulary size in NMT, several approaches have already been explored by MT researchers. Byte Pair Encoding (BPE) [41] is a form of data compression that iteratively replaces the most frequent pair of bytes in a sequence with a single unused byte. Hybrid solutions have been implemented to combine word and character models in order to achieve an open vocabulary NMT system [29]. In addition, using monolingual data for data reinforcement has gained considerable attention [52] as such an approach does not alter the neural network architectures.

There are several recent proposals for integration of external knowledge in NMT during training. Gülçehre et al., (2015) used monolingual data to train a neural Language Model (LM) that is integrated into the NMT decoder through concatenation of hidden states [21]. In the work of Arthur et al., (2016), the probability of the next target word in the NMT decoder is biased by lexicon probabilities computed from a bilingual lexicon [5]. When external knowledge is available in the form of linguistic information, Sennrich and Haddow, (2015) computed separate embedding vectors for each aspect of linguistic information, and these are then concatenated without altering the decoder [40].

Knowledge embedding has received a lot of attention in recent years, existing knowledge embedding methods aim to represent entities and relations of KG as vectors in continuous vector space, where they define a loss function to learn the representations. Different methods differ in the definition of loss functions with respect to the triple in a KG. The loss function implies some type of transformation on head and tail. With the help of a Knowledge Base (KB), [42] formulate a semantic space to connect the source and target languages, and apply it to the sequence-to-sequence framework to propose a KB Semantic Embedding (KBSE) method. In this method, the source sentence is first mapped into a KB semantic space, and then the target sentence is generated using a Recurrent Neural Network (RNN) with the internal meaning preserved. Yang and Mitchell, (2017) exploited of external KBs in a neural model called as KBLSTM that leverages continuous representations of KBs to improve the learning of RNNs for machine reading [53]. Their model utilizes an attention mechanism with a sentinel to adaptively decide whether to tap into background knowledge to determine which KB information is useful. The architecture of the KBLSTM model draws on the development of attention mechanisms that are employed in MT and Image Captioning tasks.

Du and Way, (2016) proposed an approach to address the issue of OOV words through the application of different methods using BabelNet [35]. They create additional training data and apply a post-editing technique that replaces OOV words while querying BabelNet [17].

Other prior work has considered external knowledge in the design of the mapping function from source sentence. Li et al., (2018) utilized the synonym as well as hypernym relations extracted from WordNet to find appropriate replacements for low-frequency words [26].

Approaches above have succeeded in addressing the OOV problem (to a degree) and in building semantic embedding models reliant upon a specific KB to be used in NMT systems. However, each has shortcomings related to distinguishing among potential target-language options for a source-language word (disambiguation) and incorporation of external knowledge (for translation of relations). Prior approaches have revealed the importance of finding ways to equip translation techniques with external knowledge that supports production of target-language sentences that convey the meaning of source-language counterparts.

It is clear that a NMT framework is desired that provides access to external knowledge during the translation. We adopt KG due to its suitability to applications such as Machine Reading [53], Question Answering [44], and Natural Language Interface (NLI) [4]. We expect that improved translation of sentence meaning relies on the type of knowledge that enhances these related applications, i.e., knowledge about entities and the relations between entities. To the best of our knowledge, the main KGs are Freebase [8], Google Knowledge Vault [14] and DBpedia [6], which are mainly used in English.

Of course, integration of KGs into NMT is not new. For example, Moussallem et al., (2019) describe a range of strategies for incorporating KGs into neural models and examine the influence of DBpedia in English-German translation [33]. The work of Lu et al., (2019) is most similar to ours in that it uses an external KG (WordNet) to support semantic relation modeling between source and target sentences and uses monolingual and bilingual constraints that are similar to those used in our work [30]. However, our NMT approach differs from these in that we produce a framework that is designed to overcome the high false-negative rates that lead to a reduction in overall performance.

For example, Lu et al., (2019) incorporate exact matching for linking words of training sentences in the KG [30]. A disadvantage of this approach is that, due to its low coverage, the bilingual constraint requires filtering test-set sentences for those that contain at least one trained entity. Because some words do not appear on one side of a bilingual constraint, their embedding cannot be affected by KG extraction. Thus, the words remain the same as those in the original NMT model, and the selected sentences explicitly reflect the influence of KG. In contrast, our approach utilizes EL instead of exact matching, thus eliminating the need for filtering of test-set sentences.

An additional difference between our work and that of [30] concerns the construction of positive and negative examples for distinguishing between viable translations and non-viable translations. Lu et al., (2019) adopt an approach where, for each positive KG triple, the head or tail word is replaced to construct a negative example [30]. In our work, each positive triple associated with our monolingual embedding constraint is subject to head-word replacement only. The intuition behind the single replacement choice is that it allows one-to-many relations to be captured, e.g., *each customer can have many sales orders*. Our head-word replacement approach also reduces the parameter set in the KG embedding technique and thus enables learning of a low-dimensional vector for each entity and each relationship. Additionally, probabilities are used to avoid any random replacements that may introduce false negative labels. Of course, the proof is in the pudding: the head-word replacement approach must be demonstrated as a step forward to be considered a contribution. Section 3 demonstrates improvements over prior methods due to this technique.

Other differences to be described below are the following: (1) we conduct experiments with unmodified versions of OpenNMT-py [24] and Transformer [50]; (2) we use the TransR technique [27] for training KG-embedding; (3) and we adopt Freebase [8] as our KG. We implement and test this approach for the task of English-Spanish translation and show that our method achieves a significant decrease in the number of <*UNK*> tokens.

Dasgupta et al., (2018) proposed HyTE, a temporally-aware KG embedding method that explicitly incorporates time in the entity-relation space by associating each timestamp with a corresponding hyperplane [13]. HyTE performs KG inference using temporal guidance, and predicts temporal scopes for relational facts with missing time annotations. HyTE is built on top of TransE, yielding gains over TransE alone, but also modifies TransE by treating the timestamps as hyperplanes (TransH). However, HyTE combines entities and relations into a single semantic space. As we will see below, our approach employs TransR to represent entities and their relations in distinct semantic spaces—with relation-specific bridging—and yields improved performance over HyTE.

3. Methodology

The core idea behind our methodology is to use the learned word embeddings as an encoding of the semantic relations imposed by KG and to demonstrate how this external knowledge influences translation quality. We integrate entity relations—transformed to embedding patterns independently—to boost the semantic relations between source and target words. We use external knowledge expressed in KGs by linking the words in the source sentences to the entity types in a reference KG. We jointly train two modules, a translation module and a KG embedding module, and impose consistency constraints on the embedding spaces. The goal is show that the word embeddings trained by the translation module and the word embeddings of the linked KG concepts (trained by the KG embedding module) consistently represent the same semantic relationships between words.

3.1. Neural Machine Translation Embeddings

The NMT module uses a commonly-used encoder-decoder architecture [7], where a source sentence $x = x_1, x_2, ..., x_J$ is transformed (encoded) into an internal representation $h = h_1, h_2, ..., h_J$, and then h is transformed (decoded) into a target sentence $y = y_1, y_2, ..., y_I$.

For example, to translate an English sentence *the dog likes to eat an apple* into Spanish *al perro le gusta comer la manzana*, each word is transformed into a *1-hot* encoding vector (with a single 1 associated with the index of that word, and all other indexed values 0). Each word in the dataset has a distinct 1-hot encoding vector that serves as a numerical representation that serves as input to the model.

The first step toward creating these vectors is to assign an index to each unique word in English (as the input language). This process is then repeated for Spanish (as the output language). The assignment of an index to each unique word creates a vocabulary for each language [2].

The encoder portion of the NMT model takes a sentence in English and creates a representational vector (an *embedding*) from this sentence. This vector represents the meaning of the sentence and is subsequently passed to a decoder which outputs the translation of the sentence in Spanish.

NMT models the conditional probability of the target sentence as:

$$P(y|x) = \prod_{i=1}^{I} P(y_i|y_{<i}, x) \tag{1}$$

where y_i is the target word emitted by the decoder at step i and $y_{<_i} = (y_1, y_2, ..., y_{i-1})$. The conditional output probability of a target word y_i defined as follows:

$$P(y_i|y_{<i}, x) = softmax\ (f(d_i, y_{i-1}, c_i)) \tag{2}$$

where f is a non-linear function and $d_i = g(d_{i-1}, y_{i-1}, c_i)$, g is a non-linear function. c_i is a context vector computed as the weighted sum of the hidden vectors h_j,

$$c_i = \sum_{j=1}^{J} \alpha_{t,j} h_j, \tag{3}$$

where h_j is the annotation of source word x_j, $\alpha_{t,j}$ is computed by what is known as the *attention model*, which focuses on sub-parts of the sentence during translation.

The attention mechanism supports memorization of long source sentences in NMT. Rather than building a single context vector out of the encoder's last hidden state, an attention model creates shortcuts between the context vector and the entire source input. The weights of these shortcut connections are customizable for each output element.

The context vector has access to the entire input sequence—for retention of the full context of the sentence—and controls the alignment between the source and target. Stated simply: the attention mechanism converts two sentences into a matrix where the words of one sentence form the columns, and the words of another sentence form the rows. From this, matches are obtained, thus identifying the relevant and yielding a positive impact on MT. Apart from improving the performance on MT, attention-based networks allow models to learn alignments between different modalities (different data types) for e.g., between speech frames and text or between visual features of a picture and its text description.

We adopt the OpenNMT-py translation architecture [24] based on LSTM [22] and use it for our NMT-embedding. Given a training data set with N bilingual sentences, an attention-based NMT training loss function [31] is defined as the conditional log-likelihood:

$$Loss = \sum_{n=1}^{N}\sum_{i=1}^{I} -logP(y_i^n|y_{<i}^n, x^n) \tag{4}$$

In addition to the RNN-based attentional model described above, we conduct experiments employing the Transformer model [50]. In contrast to the RNN-based mechanism, the Transformer model is a purely-attention architecture. It abandons the idea of successive encoding and iteratively applies a self-attention mechanism over inputs to obtain contextual information. The decoder also performs self-attention itself and applies a multi-head attention on the output of the encoder to produce the target translation. The encoder is a stack of six identical layers, each of which includes two sub-layers: (1) a multi-head self-attention layer; and (2) a simple position-wise fully connected feed-forward network. A residual connection around each sub-layer is used and followed by a normalization layer. The decoder is also composed of six identical layers that have the same sub-layers as those in the encoder. In addition, a multi-head attention is used over the encoder outputs to help produce the target translations [50]. Based on Equation 2, for a training dataset $\{x^n, y^n\}_{n=1}^{N}$, the NMT training loss function is defined the same as Equation 4.

3.2. Knowledge Graph Embeddings

KG embedding aims at representing entities and relations in a large-scale KG as elements in a continuous vector space. KG entities are encoded into a numerical representation for processing. Based on Annervaz et al., (2018), KG embedding techniques are classified as follows [4]:

- Structure-based embeddings, which translates subject entity to object entity using low-dimensional relation vector.
- Semantically-enriched embeddings, which learns to represent entities of the KG along with their semantic information.

We exploit the structure-based embeddings technique and use *Freebase* as a source of knowledge.

The KG yields a set of triples T consisting of a head h, a relation r, and a tail t, denoted as (h, r, t). For example, the triple <*Spain, capital, Madrid*> is extracted for *Madrid is the capital of Spain.* We view the learning of entity embeddings as central aspect of EL and employ TransR[1] [27] to learn entities' embeddings. TransR represents entities and relations in distinct semantic spaces bridged by relation-specific matrices. For each relation r in TransR, we set a projection matrix M_r that projects entities from entity space to relation space. With the mapping matrix, we define the projected vectors of entities as follows:

$$h_r = hM_r \tag{5}$$

$$t_r = tM_r \tag{6}$$

The score function is correspondingly defined as:

$$f_r(h,t) = \|h_r + r - t_r\|_2^2 \tag{7}$$

We thus enforce constraints on the norms of the embeddings h, r, t, and the mapping matrices.[2]

We align triples with the source sentences using EL. Specifically, for a document C and a set of KG entities T, we generate an assignment Q of labels $l = (l_1, l_2, .., l_n)$ to entities $Q(l) \in (T)^n$. The result is a set of named entities in the source and target sentences, linked to the KG via EL.

Next, we incorporate the Uniform Resource Identifiers (URIs) of entities along with their named entity tags. After this, we embed our KG employing the TransR technique and then concatenate the embedding vectors to the internal vectors of the NMT embeddings [33].

Having described the NMT-embedding and KG-embedding modules (and the correspondence between them), we now turn to the monolingual and bilingual constraints imposed on these two embeddings.

3.2.1. Monolingual Embedding Constraint

Monolingual constraints are imposed via KG entity relations that influence the training of semantic embeddings of the source words. Triples whose h word appears in the source sentence are extracted, yielding a set of positive examples, denoted as:

$$S = \{(h, r, t) | h \in x\} \tag{8}$$

[1] https://github.com/thunlp/KB2E/tree/master/TransR

[2] $\forall r, t$ we have $\|h\|_2 \leq 1$, $\|t\|_2 \leq 1$, $\|hM_r\|_2 \leq 1$ and $\|tM_r\|_2 \leq 1$.

For each triple in S, the h word is replaced to make a set of negative examples S' which includes (h', r, t):[3]

$$S' = \{(h', r, t)|h' \in T\} \tag{9}$$

The loss function for monolingual constraints ($Loss_{mono}$) is defined as follows:

$$Loss_{monolingual} = \sum_{(h,r,t)\in S} \sum_{(h',r,t)\in S'} max(0, f_r(h,t) + \lambda - f_r(h',t)) \tag{10}$$

where $\lambda > 0$ is a margin hyper parameter.

Training embeddings under this constraint and using the negative and positive triples guarantees the distance between linked words in the KG is smaller than the distance between irrelevant words. This facilitates entity disambiguation during translation. For example, consider the source sentence *The bill has been added to law by the US President*. We assume the source word *president* has the relation r with the head *Barack Obama* in the KG. We replace the source word with the head in the KG to construct a negative example.

3.2.2. Bilingual Embedding Constraint

The bilingual embedding constraint maintains the relation between source entities and their corresponding translations. All triples for which both h and t appear in the source sentence are extracted:

$$S_{src} = \{(h_{src}, r, t_{src})|(h_{src}, t_{src}) \in x\} \tag{11}$$

Then h_{src} and t_{src} are aligned with their corresponding translations:

$$S_{trg} = \{(h_{trg}, r, t_{trg})|(h_{trg}, t_{trg}) \in y\} \tag{12}$$

Without this constraint there would be a gap between the source triples and their aligned target triples. Following Lu et al., (2019), the loss function $Loss_{bi}$ is applied to minimize the potential for a gap [30]:

$$Loss_{bilingual} = -\sum_{(h_{src},r,t_{src})\in S_{src}} \sum_{(h_{trg},r,t_{trg})\in S_{trg}} |f_r(h_{src}, t_{src}) - f_r(h_{trg}, t_{trg})| \tag{13}$$

For example, given a source sentence *The bill has been added to law by the US president*, the relation between the Spanish words *projecto* (bill) and *presidente* (president) in the target language is the same as that between *bill* and *president* in the source language. The bilingual constraint is modeled during the training process and makes the NMT system more knowledgeable.

[3] To avoid random replacement which may introduce false negative labels, we employ probabilities.

3.2.3. Joint Training

The monolingual and bilingual embedding constraints are employed to augment semantic word embeddings during the NMT training process. To implement this idea, the overall loss function is defined such that it includes the conventional translation loss as well as the entity relation loss described above for the monolingual and bilingual constraints (we call those KG-losses). The translation loss and the KG-losses will be optimized iteratively. Thus, the final loss function are written as follows:

$$Loss_{final} = \sum_{n=1}^{N}\sum_{i=1}^{I} -logP(y_i^n|y_{<i}^n, x^n) + \alpha\frac{1}{N}\sum_{n=1}^{N} Loss(x^n, y^n) \tag{14}$$

where α is a hyper parameter and N denotes the number of training examples. The $Loss(x^n, y^n)$ function for the monolingual and bilingual embedding constraints is denoted as $Loss_{monolingual}$ and $Loss_{bilingual}$, respectively.

4. Experimental Framework

There are massive resources available to build an English-Spanish NMT system in the framework of the WMT'18 translation task.[4] Our bilingual dataset includes about 2.1M sentences collected from *Europarl* as well as *News-Commentary* for training and development sets. We also use the *newstest2012* and *newstest2013* as our test sets.

For the KG, we extract triples from the human-created Freebase[5] (FB15k) which was launched by Metaweb as an open and collaborative KB [8]. Freebase includes general knowledge and partially covers common-sense knowledge and domain knowledge [43].

Our training data consist of 2M parallel sentences. We use 2K parallel sentences for validation and 3K parallel sentences for test. For knowledge extraction, we use FB15k[6] which includes 14,951 entities and 1,345 relationships.

For the RNN-based experiments, we employ OpenNMT-py[7] model [24] on top of PyTorch which is based on a bi-directional 2-layer LSTM encoder-decoder with attention [7]. Training uses a batch size of 32 and the Stochastic Gradient Descent (SGD) [38] with an initial learning rate of 0.01. We set the size of word embeddings as well as hidden layers to 500. We also set dropout to 0.1. We use a maximum sentence length of 50 words and shuffle mini-batches as we proceed.

Following Jean et al., (2015), we limit our vocabulary to be the top 50 most frequent words for both languages [23]. Words that are not in these shortlisted vocabularies are converted into a universal token <*UNK*>. We also set a beam size of 5 and λ to 1. During the training, we set α to 0.001, 0.01, 0.1 for both monolingual and

[4] http://www.statmt.org/wmt18/translation-task.html
[5] https://developers.google.com/freebase
[6] https://everest.hds.utc.fr/doku.php?id=en:transe
[7] https://github.com/OpenNMT/OpenNMT-py

bilingual constraints. The model continues for 20 epochs (both training and testing) on a single GPU. In all experiments, we used an EL system introduced by Moussallem et al., (2017) [34].

Recent comparisons between neural network architectures and RNNs have yielded different conclusions for different Natural Language Processing (NLP) tasks [28, 54]. Tran et al., (2018) concluded that RNNs perform better than Transformers on a subject-verb agreement task [49], but Tang et al., (2018a) also found that Transformer models surpass RNN models only under high-resource conditions [46]. Transformers were compared favorably to RNNs for a Word Sense Disambiguation (WSD) task [47] (determined by scoring contrastive translation pairs) with the conclusion that Transformers are better at extracting semantic features. As such, we employ both architectures for evaluating KGs within NMT. For the PyTorch implementation of the Transformer[8] [50], we use a 6-layer encoder-decoder and a batch size of 2048. We set hidden layers as well as word embeddings of size 512. We set the rest of the values to be the same as in the OpenNMT-py setting. Our evaluation metric is BLEU [37].[9]

5. Results Analysis and Evaluation

Table 1 shows the results of the monolingual embedding constraint employing OpenNMT-py (RNN-based) and Transformer-based architectures on the *newstest* datasets *2012* and *2013*. The KG systems (RNN-KG and Transformer-KG) containing our monolingual embedding constraint lead to BLEU improvements over their corresponding baselines (RNN, Transformer). For *newstest2012*, RNN-KG outperforms RNN by around +0.51 and Transformer-KG outperforms Transformer by around +0.47. Similar increases were found for *newstest2013*, +0.55 and +0.49, respectively.

Table 1
BLEU scores for the English-Spanish translation task using monolingual embedding constraint using RNN and Transformer.

Models	newstest12	newstest13
RNN	17.62	18.06
Transformer	19.46	19.94
RNN-KG	18.13	18.61
Transformer-KG	19.93	20.43

Table 2 shows the results of the bilingual embedding constraint using OpenNMT-py (RNN-based) and Transformer-based architectures on the *newstest* datasets *2012* and *2013*. KG systems (RNN-KG and Transformer-KG) containing our bilingual embedding constraint *also* lead to BLEU improvements over their corresponding base-

[8] https://github.com/SamLynnEvans/Transformer
[9] BLEU scores are computed with *multi-bleu.perl*

lines (RNN, Transformer). For *newstest2012*, RNN-KG outperforms RNN by around +0.74 and Transformer-KG outperforms Transformer by around +0.67. Similar increases were found for *newstest2013*, +0.89 and +0.73, respectively.

Table 2
BLEU scores for the English-Spanish translation task using bilingual embedding constraint using RNN and Transformer.

Models	newstest12	newstest13
RNN	19.55	20.18
Transformer	20.29	21.05
RNN-KG	20.29	21.07
Transformer-KG	20.96	21.78

Furthermore, we observed that our approach achieves a significant decrease in the number of <*UNK*> tokens in both monolingual and bilingual embedding constraints. The reason behind this improvement is that modeling the relations provides sufficient training for low-frequency entities that were difficult to handle before and without KG. Tables 3 and 4 show the statistics for this improvement with respect to <UNK> words in the proposed embedding constraints:

Table 3
Statistics for the improvement in handling <UNK> tokens employing monolingual embedding constraint.

Models	newstest12	newstest13
RNN-KG	27.48%	33.51%
Transformer-KG	29.72%	36.52%

Table 4
Statistics for the improvement in handling <UNK> tokens employing bilingual embedding constraint.

Models	newstest12	newstest13
RNN-KG	18.12%	20.66%
Transformer-KG	23.47%	26.08%

Compared to the results of prior similar work [30, 33], our approach employing TransR improves upon the performance of methods described in the section of Related Work. For example, the HyTE approach combines entities and relations into a single semantic space, whereas ours uses distinct semantic spaces for entities and relations—with bridging between them.

Employing Transformer-based architecture introduces more controlling knobs than RNN-based architecture, which controls the flow and mixing of inputs as per trained weights. So, the Transformer model gives us the most control-ability and thus, better results than the RNN model.

A detailed study of our results shows that the number of OOV words decrease considerably with the KG embedding augmentation. Many OOV words are in fact entities contained in the KG. We consider two cases from *newstest2013* here: (1) In the sentence *The US president is represented in the European Parliament*, the term "US" is not translated by the RNN baseline. However, it is correctly translated into Spanish as "Estados Unidos" by both KG embeddings models. Additionally, the Transformer baseline is capable of translating "US" to "Estados Unidos." (2) In the excerpt *Bill to increase prime minister's powers added to EU law*, the Transformer-KG model translates the word "prime minister" correctly to "primer ministro" with the correct gender.

As another example, the entity "air force" in the sentence *The Army Air Force is equipped with new air defense equipment* (extracted from *newstest2012*), is not translated correctly by baseline models, whereas both KG embeddings models are able to translate it correctly as "fuerza aerea." This evaluation suggests that the RNN-KG models, as well as the Transformer-KG models, are able to correctly learn the translation of entities through the relations found in KG embeddings.

Several options are available for capturing KG embeddings. Our primary motivation for selecting TransR is its adaptability to the MT problem in light of the range of challenges we have identified in this paper (e.g., OOVs and disambiguation).

Another option is TransE [9], which requires $h + r \approx t$ when (h, r, t) holds. This requirement indicates that t is ideally the neighbor of $(h+r)$. Hence, TransE assumes the score function below:

$$f_r(h,t) = \|h + r - t\|_2^2 \tag{15}$$

which is low if (h, r, t) holds, and high otherwise. TransE applies well to 1-to-1 relations but is not designed for N-to-1, 1-to-N and N-to-N relations.

TransH [51] addresses this relational issue by allowing an entity to have distinct distributed representations when involved in different relations. For a relation r, TransH models the relation as a vector r on a hyperplane with w_r as the normal vector. For a triple, the entity embeddings h and t are first projected to the hyperplane of w_r, denoted as $h_\perp$ and $t_\perp$. Then the score function is defined as follows:

$$f_r(h,t) = \|h_\perp + r - t_\perp\|_2^2 \tag{16}$$

If we restrict $\|w_r\|_2 = 1$, we will have:

$$h_\perp = h - w_r^\top h w_r \tag{17}$$

$$t_\perp = t - w_r^\top t w_r \tag{18}$$

By projecting entity embeddings into relation hyperplanes, TransH allows entities to play different roles in different relations. However, the relational complexity imposed by this hyperplane approach is not amenable to the design we have chosen because an entity may have multiple aspects, and various relations focus on different aspects of entities. Hence, it is intuitive that some entities are similar and thus close to each other in the entity space, but are comparably different in some specific aspects and thus far away from each other in the corresponding relation spaces.

Both TransE and TransH assume embeddings of entities and their relations within the same space despite that these are completely different objects. TransH affords more flexibility by employing relation hyper-planes, but does not perfectly lift this representational restriction. Given these representational issues, we have elected to apply the TransR technique, which provides the best of both worlds, i.e., modeling of entities and relations in distinct spaces (relation-specific entity spaces alongside multiple relation spaces) and performing translation in the corresponding relation space.

6. Conclusions and Future Work

In this paper, we used Freebase as a source of KG information for NMT models and we employed TransR technique to represent entities and their relations. We also utilized EL to align triples in the KG with the source sentences. During NMT model training through both OpenNMT-py and Transformer, we applied monolingual embedding constraints to ensure that the embeddings of the source words hold the semantic relations provided by the KG. We also used bilingual embedding constraints to force the semantic relationship between the source words to be exactly maintained by their corresponding translations. Our results show improvements over the NMT baselines.

Our next step is to investigate the influence of a range of different KGs on MT such as Google Knowledge Vault and others with multilingual support (DBpedia and YAGO).

Acknowledgements

The authors would like to express their sincere gratitude to Dr. Kimberly Foster, Dr. Michael W. Mislove, Dr. Carola Wenk, and Dr. Anastasia Kurdia (Tulane University of Louisiana, USA), Dr. K. Brent Venable, and Taher Rahgooy (University of West Florida, USA) for all their unconditional support.

References

[1] Ahmadnia B., Dorr B.J.: Augmenting Neural Machine Translation through Round-Trip Training Approach. In: *Open Computer Science*, vol. 9(1), pp. 268–278, 2019.

[2] Ahmadnia B., Dorr B.J.: Enhancing Phrase-Based Statistical Machine Translation by Learning Phrase Representations Using Long Short-Term Memory Network. In: *Proceedings of Recent Advances on Natural Language Processing*, pp. 25–32. 2019.

[3] Ahmadnia B., Serrano J., Haffari G.: Persian-Spanish low-resource statistical machine translation through English as pivot language. In: *Proceedings of the 9th International Conference of Recent Advances in Natural Language Processing*, pp. 24–30. 2017.

[4] Annervaz K.M., Chowdhury S.B.R., Dukkipati A.: Learning beyond datasets: Knowledge Graph Augmented Neural Networks for Natural language Processing. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 313–322. 2018.

[5] Arthur P., Neubig G., Nakamura S.: Incorporating Discrete Translation Lexicons into Neural Machine Translation. In: *Proceedings of the Second Conference on Machine Translation*, pp. 157–168. 2016.

[6] Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z.: DBpedia: A Nucleus for a Web of Open Data. In: *Proceedings of the 6th International Semantic Web Conference*, pp. 722–735. 2008.

[7] Bahdanau D., Cho K., Bengio Y.: Neural machine translation by jointly learning to align and translate. In: *Proceedings of the International Conference on Learning Representations*. 2015.

[8] Bollacker K., Evans C., Paritosh P., Sturge T., Taylor J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250. 2008.

[9] Bordes A., Usunier N., Garcia-Duran A., Weston J., Yakhnenko O.: Translating embeddings for modeling multi-relational data. In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 2787–2795. 2013.

[10] Chah N.: OK Google, What Is Your Ontology? Or: Exploring Freebase Classification to Understand Google's Knowledge Graph. In: *CoRR*, vol. abs/1805.03885, 2018.

[11] Chatterjee R., Negri M., Turchi M., Federico M., Specia L., Blain F.: Guiding Neural Machine Translation Decoding with External Knowledge. In: *Proceedings of the Second Conference on Machine Translation*, pp. 157–168. 2017.

[12] Chousa K., Sudoh K., Nakamura S.: Training Neural Machine Translation using Word Embedding-based Loss. In: *CoRR*, vol. abs/1807.11219, 2018.

[13] Dasgupta S.S., Ray S.N., Talukdar P.: HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2001–2011. 2018.

[14] Dong X., Gabrilovich E., Heitz G., Horn W., Lao N., Murphy K., Strohmann T., Sun S., Zhang W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 601–610. 2014.

[15] Dorr B.J.: Machine Translation Divergences: A Formal Description and Proposed Solution. In: *Computational Linguistics*, vol. 20(4), pp. 597–633, 1994.

[16] Dorr B.J., Pearl L., Hwa R., Habash N.: DUSTer: A Method for Unraveling Cross-Language Divergences for Statistical Word-Level Alignment. In: *Proceedings of the 5th conference of the Association for Machine Translation in the Americas*. 2002.

[17] Du J., Way A.: Using babelnet to improve OOV coverage in SMT. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pp. 9–15. 2016.

[18] Färber M., Ell B., Menne C., Rettinger A.: A Comparative Survey of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. In: *Semantic Web Journal*, pp. 1–26, 2015.

[19] Gehring J., Auli M., Grangier D., Yarats D., Dauphin Y.N.: Convolutional Sequence to Sequence Learning. In: *CoRR*, 2017.

[20] Gu J., Lu Z., Li H., Li V.O.: Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1631–1640. 2016.

[21] Gülçehre Ç., Firat O., Xu K., Cho K., Barrault L., Lin H., Bougares F., Schwenk H., Bengio Y.: On Using Monolingual Corpora in Neural Machine Translation. In: *CoRR*, vol. abs/1503.03535, 2015.

[22] Hochreiter S., Schmidhuber J.: Long short-term memory. In: *Neural computation*, vol. 9(8), pp. 1735–1780, 1997.

[23] Jean S., Cho K., Memisevic R., Bengio Y.: On using very large target vocabulary for neural machine translation. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pp. 1–10. 2015.

[24] Kelin G., Kim Y., Deng Y., Senellart J., Rush A.M.: OpenNMT: Open-Source Toolkit for Neural Machine Translation. In: *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics*, pp. 67–72. 2017.

[25] Koehn P., Och F.J., Marcu D.: Statistical phrase-based translation. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 48–54. 2003.

[26] Li S., Xu J., Miao G., Zhang Y., Chen Y.: A Semantic Concept Based Unknown Words Processing Method in Neural Machine Translation. In: *Natural Language Processing and Chinese Computing*, pp. 233–242. 2018. ISBN 978-3-319-73618-1.

[27] Lin Y., Liu Z., Sun M., Liu Y., Zhu X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 2181–2187. 2015.

[28] Linzen T., Dupoux E., Goldberg Y.: Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. In: *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 521–535, 2016.

[29] Loung M.T., Manning C.D.: Achieving open vocabulary neural machine translation with hybrid word-character models. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1054–1063. 2016.

[30] Lu Y., Zhang J., Zong C.: Exploiting Knowledge Graph in Neural Machine Translation. In: *Proceedings of Machine Translation: 14th China Workshop, CWMT*, pp. 27–38. 2019.

[31] Luong T., Pham H., Manning C.D.: Effective Approaches to Attention-based Neural Machine Translation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421. 2015.

[32] Luong T., Sutskever I., Le Q., Vinyals O., Zaremba W.: Addressing the Rare Word Problem in Neural Machine Translation. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 11–19. 2015.

[33] Moussallem D., Arcan M., Ngonga Ngomo A.C., Buitelaar P.: Augmenting Neural Machine Translation with Knowledge Graphs. In: *CoRR*, vol. abs/1902.08816, 2019.

[34] Moussallem D., Usbeck R., Röder M., Ngonga Ngomo A.C.: MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach. In: *Proceedings of Knowledge Capture Conference*. 2017.

[35] Navigli R., Ponzetto S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. In: *Artif. Intell.*, vol. 193, pp. 217–250, 2012.

[36] Pan S.J., Yang Q.: A Survey on Transfer Learning. In: *IEEE Transactions on Knowledge and Data Engineering*, vol. 22(10), pp. 1345–1359, 2010.

[37] Papineni K., Roukos S., Ward T., Zhu W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311–318. 2001.

[38] Robbins H., Monro S.: A stochastic approximation method. In: *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[39] Rumelhart D.E., Hinton G.E., Williams R.J.: Learning representations by back-propagating errors. In: *Nature*, vol. 323, pp. 533–536, 1986.

[40] Sennrich R., Haddow B.: Linguistic Input Features Improve Neural Machine Translation. In: *Proceedings of the First Conference on Machine Translation*, pp. 83–91. 2015.

[41] Sennrich R., Haddow B., Birch A.: Neural Machine Translation of Rare Words with Subword Units. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725. 2016.

[42] Shi C., Liu S., Ren S., Feng S., Li M., Zhou M., Sun X., Wang H.: Knowledge-Based Semantic Embedding for Machine Translation. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 2245–2254. 2016.

[43] Song Y., Roth D.: Machine Learning with World Knowledge: The Position and Survey. In: *CoRR*, vol. abs/1705.02908, 2017.

[44] Sorokin D., Gurevych I.: Modeling semantics with gated graph neural networks for knowledge base question answering. In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3306–3317. 2018.

[45] Sutskever I., Vinyals O., Le Q.V.: Sequence to sequence learning with neural networks. In: *Proceedings of the Internationa Conference on Neural Information Processing Systems*, pp. 3104–3112. 2014.

[46] Tang G., Cap F., Pettersson E., Nivre J.: An Evaluation of Neural Machine Translation Models on Historical Spelling Normalization. In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1320–1331. 2018.

[47] Tang G., Muller M., Rios A., Sennrich R.: Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4263–4272. 2018.

[48] Tang G., Sennrich R., Nivre J.: An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation. In: *Proceedings of the Third Conference on Machine Translation*, pp. 26–35. 2018.

[49] Tran K., Bisazza A., Monz C.: The Importance of Being Recurrent for Modeling Hierarchical Structure. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4731–4736. 2018.

[50] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I.: Attention is all you need. In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 5998–6008. 2017.

[51] Wang Z., Zhang J., Feng J., Chen Z.: Knowledge Graph Embedding by Translating on Hyperplanes. In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 1112–1119. 2014.

[52] Wu L., Tian F., Qin T., Lai J., Liu T.Y.: A Study of Reinforcement Learning for Neural Machine Translation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 3612–3621. 2018.

[53] Yang B., Mitchell T.: Leveraging Knowledge Bases in LSTMs for Improving Machine Reading. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1436–1446. 2017.

[54] Yin W., Kann K., Yu M., Schütze H.: Comparative Study of CNN and RNN for Natural Language Processing. In: *CoRR*, vol. abs/1702.01923, 2017.

Affiliations

Benyamin Ahmadnia
Department of Computer Science, Tulane University, New Orleans, LA, United States,
ahmadnia@tulane.edu, ORCID ID: https://orcid.org/0000-0003-2992-0188

Bonnie J. Dorr
Institute for Human and Machine Cognition (IHMC), Ocala, FL, United States,
bdorr@ihmc.us

Parisa Kordjamshidi
Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, United States, kordjams@msu.edu

Received: ??.??.20??
Revised: ??.??.20??
Accepted: ??.??.20??