# Enhancing Language Models with Knowledge Graph Embeddings

**Andrew Conley**
Rensselaer Polytechnic Institute (RPI)
110 Eighth Street
Troy, NY USA 12180
Email: conlea@rpi.edu

**Jugal Kalita**
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, Colorado 80918
Email: jkalita@uccs.edu

### Abstract

Most NLP tasks use word embeddings to improve performance. Breakthroughs like ELMo (Peters et al. 2018) and BERT (Devlin et al. 2018) have shown that state of the art results can be achieved in many NLP tasks through good language models, even without a task-specific architecture. Word vectors have been a simple, popular, and effective language model for years. Methods for generating these word vectors typically use unsupervised learning based on the context in which each word is used within the greater corpus. We propose new method of generating these word vectors. We use knowledge embeddings extracted from knowledge bases like Freebase and WordNet (Bordes et al. 2013) as a starting point, and introduce syntactic information captured from existing language models. By incorporating knowledge directly into the word embedding we aim to improve the task of natural language inference, similar to those achieved by applying knowledge bases to machine reading (Yang and Mitchell 2019). The new embeddings are judged primarily on their perfermance on the natural language inference model HBMP (Talman, Yli-Jyrä, and Tiedemann 2019). This performance is compared to that of Glove (Pennington, Socher, and Manning 2014), with the same architecture. No improvement was found to accuracy, however further steps to achieve the desired improvement are well defined.

## Introduction

Many NLP tasks see improved performance through the use of a pre-trained language model. Recently, BERT has been used to achieve state of the art results on many NLP tasks, even without the use of extensive task specific architectures (Devlin et al. 2018). Common methods of generating word embeddings like Word2vec (Mikolov et al. 2013), GloVe (Pennington, Socher, and Manning 2014), and Fasttext (Bojanowski et al. 2017), assign a vector representation to each word in a particular vocabulary. Metrics like the Euclidean distance or cosine similarity between vectors are thought of as a pseudo measurement of similarity between words in the language model. Something to note about these processes is that they are built solely from the context in which a word is used within its corpus, and do not incorporate outside knowledge. This leads to results that contradict this notion of similarity. For example, using Word2vec yields a very small distance between the vector for "$Hello''$ and the vector for "$Dolly''$ despite the meanings of the two words being unrelated. We propose using altered knowledge graph embeddings as a language model for natural language inference in order to capture semantic information instead of just syntactic information.

Knowledge graphs are large graphs of triples made up of two entities (nodes) and a relation (edge) connecting them. These graphs are often highly curated and contain specific information about how various entities are related. When dealing with knowledge graphs, we often talk about the task of link prediction. Link prediction takes an existing knowledge graph and tries to predict relations between entities not present in the graph. Graph embeddings, a method of modelling entities and relations, as well as scoring triples, are used to achieve good results in link prediction(Trouillon et al. 2016)(Ding et al. 2018)(Bordes et al. 2013).

TransE (Bordes et al. 2013) is a model for graph embedding that represents entities as low dimensional vectors and relations as translations (low dimensional vector similar to those of entities) on those vectors. For head entity $h$, tail entity $t$, and relation (link) $l$, training minimizes the euclidean distance between $h + l$ and $t$, while maximizing the distance between $h + l$ and other entities.

The TransE model is particularly suited for creating a general language model since it models entities as low dimensional vectors, the same form used by methods like Word2vec, GloVe, and Fasttext. This allows TransE embeddings to replace these common language models without needing to alter the architecture used for downstream tasks. TransE also includes a constraint on entity embeddings, restricting their L2 norm to be equal to 1 This was done to prevent artificial reduction of the loss function by increasing distance between entities that do not have a relation by arbitrarily increasing the size of entity embeddings. These restrictions keep the embeddings in a smaller space, which keeps related entities closer together, much like Word2vec and similar language models.

## Related Work

Generating good embeddings for rare words is a difficult task and big problem. Since generating word embeddings is often an unsupervised task, a large amount of data is required to create good embeddings. Rare words, by definition, do not appear often, and this lack of data often results in unreliable word vectors that do not accurately represent the

similarity of a rare word to other words in the vocabulary. Herbelot and Baroni (2017) describe an effective method to train these rare word vectors by altering word2vec. This method learns the majority of the vocabulary in accordance with the methods of word2vec, then holds the learned word vectors constant while learning the rare words with a higher rate of learning and a larger window size to take in as much context as possible. This paper also utilizes a smaller, curated subset of its corpus to train the rare words. This subset is made up of sentences from encyclopedic data, with context believed to be highly informative to the meaning of the rare word.

A La Carte Embedding (Khodak et al. 2018) builds off of this paper applying similar techniques to phrases. Notably, this word achieves low computational cost embedding chimeras (a combination of words to simulate a new rare word), and excellent results. Chimera representations for idioms have been shown to be very similar to their meaning, for example, the representation for beef up was (by cosine similarity) most similar to need and improve, and the representation for cutting edge was most similar to innovative and technology.

Another approach for improving rare word embeddings is the Fasttext language model (Bojanowski et al. 2017). Fasttext is a character based language model where words vectors are a sum of n-gram vectors. One major advantage of this is that out of vocabulary words, a huge problem for dictionary based models, can have approximate meanings guessed from the vector representation of prefixes, suffixes, and root words. This model creates word vectors that are very related to words that share a root, which both strongly indicates similarity linguistically and allows detection of similarity across languages that share roots (i.e. romance languages sharing Latin roots). This model has been combined with a Gaussian Distribution model (Athiwaratkun and Wilson 2018) which uses probability densities to be able to model multiple meanings of a single word. The result of the combination, Probabilistic Fasttext (Athiwaratkun, Wilson, and Anandkumar 2018) is able to accurately predict the meaning of rare, misspelt, and unseen words. This model seperates the representation of a word into various mixture component, each representing a different sense or meaning of the word. In doing so, this model has achieved state of the art results on tasks that require the use and differentiation of different meanings of a word.

There are currently many approaches to generating graph embeddings for the task of link prediction. One notable technique applies constraints to limit irrelevant or unwanted information taken from the knowledge graph when generating embeddings (Ding et al. 2018). These constraints came in two forms: a non-negativity constraint and approximate entailment constraints. The non-negativity constraint does not allow negative relations to be considered (i.e. a cat is not a car, or a frog is not an instrument). The effect of this constraint was a more sparse and interpretable set of entity vectors. The approximate entailment constraint allowed detection of entailment between relations. Using the ComplEx (Trouillon et al. 2016) model, Ding et al. were able to formulate an equation for detecting entailment with probability $\lambda$. This constraint alters the imaginary component of each vector to encode this entailment information without greatly affecting metrics of entity similarity in the vector space (distance and cosine similarity).

| Embedding Model | Accuracy |
|---|---|
| GloVe | 85.89% |
| TransE ConceptNet | 71.35% |
| Word2vec | 71.35% |
| TransE ConceptNet+Word2vec | 71.35% |
| TransH ConceptNet+Word2vec | 71.35% |
| DistMult ConceptNet+Word2vec | 71.35% |
| Randomly Initialized | 71.35% |

Table 1: Results of each language model on snli dataset using HBMP model

## Results

We propose a new method of altering knowledge graph embeddings to act as a language model for natural language inference. We approach this by replacing GloVe word vectors with standard knowledge embeddings of the knowledge graph Concept Net, and replacing those vectors with knowledge embeddings generated from altered knowledge graph. The quality of created language models is evaluated by performance on the natural language inference model described in (Talman, Yli-Jyrä, and Tiedemann 2019).

Initially, standard TransE embeddings were trained on Concept Net triples and applied to the HBMP model. Using OpenKE packages (Han et al. 2018), entities and relations are indexed and triples are split into a text file for training and a text file for testing, both of the form, entityid entityid relationid. TransE was chosen for this initial task because it models entities and relations from the knowledge graph as low dimensional vectors, similar to the GloVe embeddings to be replaced.

In an attempt to improve these embeddings, a new, modified set of triples was created in an attempt to include syntactic information from the successful language model word2vec. New triples were generated and added to Concept Net to incorporate the similarity found in an existing word2vec model. These triples were created by linking each word in the model's vocabulary (entities) with the word closest to it by cosine similarity, via a new context relation. In practice, this yields connections such as "$unamused''$ is related by context to "$bemused''$ or "$pastured''$ is related by context to "$Simmental_cattle''$. The word2vec model from which these connections were generated was downloaded pre-trained on the google news groups corpus.

As an extension to the previous improvement, we attempted to expand this to a general $n$ most similar words by cosine similarity. These words would be connected by relations, Ċontext0, Context1, ..., Contextn̈respectively. Unfortunately, this proved too time consuming for the scope of this project, as these new sets of triples would have taken weeks to create.

We then moved away from TransE graph embeddings to TransH and DistMult. Both of these new embedding mod-

els have proven more effective than TransE at the task of link prediction, which would suggest it might generate a better general language model. TransH models entities in the same way as TransE and GloVe, allowing for simple substitution of the newly created vectors for the default GloVe vectors. TransH differs from TransE in that it models relations as a hyperplane, instead of just a translation. This allows for more intricate combinations of relations to be modeled and tested successfully in the task of link prediction. DistMult models relations as a bilinear operator between entities. This model is also a multiplicative graph embedding model, meaning in the task of link prediction, the scoring and prediction of triples is based on a multiplication of vectors instead of addition in the case of TransE and TransH. Multiplicative models tend to perform better than additive ones at link prediction, but are less similiar to conventional language models like GloVe.

The vectors we have been dealing with so far have all used 300 dimensions. However, DistMult embeds its entities with 100 dimensional vectors by default. Scaling the size of these vectors up to 300 during training would be possible, but not feasible given time constraints. In order to allow these embeddings to match the HBMP architecture, they were padded with values of 0 to reach 300 dimensions.

Table 1 shows the accuracy achieved on the NLI task for each set of embeddings described above. A set of randomly initialized vectors was also tested to use a baseline of a language model expected to cause no improvement. Every set of embeddings except the original GloVe embeddings achieved an accuracy of 71.35, while the GloVe embeddings achieved 85.89. The interpretation of these results will be discussed in the next section.

## Discussion

Every novel language model tested produced the same result of 71.35. This accuracy is notably worse than the 85.89 observed using GloVe. We believe this occurred either due to little overlap of key vocabulary between the knowledge graph and NLI corpus, or because of unexpected behavior of the HBMP model when changing embeddings.

Assuming the HBMP model performed as expected, the poor results were likely due to the novel language models lacking reliable representations of common words from the NLI corpus. The knowledge graph Concept Net is particularly useful when it has been applied previously because it captures semantic information. General language models merely capture syntactic information, the general context each word is used in, but lacks any complex or deeper understanding. Concept Net is able to capture complex ideas used in speach. This is apparent by the presence of idioms as entities in its graph. For example, the phrase "$few_cards_shy_of_full_deck$" is present in Concept Net. The meaning of this phrase is not the same as the literal interpretation of those words in sequence. Therefore, the ability to leverage this complex concept should prove invaluable to natural language inference, a task that inherently requires some kind of understanding. The problem is that in Concept Net this idiom is represented as written above, several words connected by underscores and stored as a single entity. This is not the same format that would emerge in the NLI task. If this phrase was present at all, it would be a series of words without underscores, which the model would not equate to the representation of the idiom found in Concept Net. When we consider how common phrases or expressions are in Concept Net, and that many of connections present for individual words in Concept Net connect those words to phrases, it seems likely that Concept Net fails to create a densely connected network of words found in our downstream task. This interpretation is also supported by the performance of the randomly initialized vectors. Randomly initialized vectors should contain no useful information whatsoever, and therefore not enhance the performance of the NLI task. Since the same results are seen using our Concept Net models and random vectors, our vectors provide no improvement to the model.

If these results were caused primarily by an issue of vocabulary, we would expect including information from Word2vec to improve results. Word2vec trained on the google news corpus should share a large portion of vocabulary with the GloVe embeddings as well as the NLI task. However we observe the same poor results on each model that includes Word2vec based triples. In practice, merely 1 context relation did not add any significant information. When looking through the added triples by hand, it becomes clear that very often the nearest vector is too similar to the original vector to add any meaningful information. We see many relations connecting words like peice to piece (a misspelling) or connecting mage to mages (plural). If these words were present in Concept Net to begin with, it is likely that these relations are already represented. Also, it becomes evident that a single relation for each word cannot create good representations. If a word from Word2vec is not present in Concept Net, it is not meaningfully added to the vocabulary. Although it technically becomes included in the vocabulary, it will only be connected to its closest word, which is also likely not in the vocabulary. This yields a very sparse set of connections, which will net create useful vectors. However, even if the word is present in Concept Net, we are still only adding a single relation. It is possible that many relations (although expensive with regard to time) may improve these results, but that is not certain. Even with hundreds of relations, it is likely that the words from Word2vec will become significantly connected with other words from Word2vec and entirely unconnected to entities from Concept Net. In that case, Concept Net would not add anything to the language model, and performance would strive to match plain Word2vec, not surpass it.

All issues discussed above would also apply DistMult and TransH. If the issues already presented were not the cause of these poor results, it is possible that existing graph embedding methods are inherently ill-suited for acting as a general language model. Typical language models use unsupervised learning to group words used in similar contexts. This keeps similar words very close together in the vector space. Graph embeddings, on the other hand, require more specific information than just whether or not two words are related. Because a graph embedding needs to know the way in which

two entities are related, it often does not group similar vectors very close together. Instead these embeddings rely on using specific directions or other similar methods to distinguish between relations. Ambiguity is unwelcome in graph embeddings, as it is as important to be confident two entities do not share a relation as it is to know the entities do share a relation. Because of this, a greater distance between entities can often prove benefitial. The TransE paper specifical mentioned needing to add a constraint to prevent entities from drifting too far apart, which was found to happen often to arbitrarily reduce the loss function during training.

Alternatively, these poor results may be due to unexpected behavior involving changing the language model, or a failure to import the new vectors. If this is the case, many of the previously discussed issues are still relevant.

Concept Net includes many phrases and idioms. Assuming those phrases are connected to many other entities that can illustrate their meaning, those representations contain valuable information. However, that information cannot be accessed in its current state. Merely introducing connections between words in Word2vec will not cause the representation of a Concept Net phrase to be used in a NLI task if the phrase does not appear in the same form (underscores).

We also would not know if the graph embedding models are suitable to be used as a general language model. As stated earlier, graph embeddings often have more euclidean distance between vectors. This makes the vectors themselves more varied between related words. Since the layers of a neural network use matrix multiplication to generate outputs, intuition would suggest vectors that are made up of very similary values would produce very similar results with a neural NLI model. It is desireable to have similar words behave in similar ways, so loosing that quality may hurt performance when switching to a graph embedding.

## Future Work

Given adequate time, we would like to explore including more relations from Word2vec as triples. We would also like to develop a method of connecting Word2vec entities to existing Concept Net entities. Recent work (Yang and Mitchell 2019) has shown good results applying attention to knowledge graph embeddings. Specifically their method of choosing embeddings to attend to, searching the graph for substring matches, is a promising system for choosing entities to relate to our vocabulary.

Once serious improvement is seen, we will experiment with restricting the information used from our knowledge graph. Work by (Ding et al. 2018) has shown that simple constraints, such as not including negative relations (is not a, does not have, etc.), can greatly improve performance in link prediction. One would expect a negative relation contradict the advantages of a language model, keeping similar words near each other in vector space, by reducing the distance between words that are known to be related by their dissimilarity, for instance, an Antonym relation.

We would still like to experiment with othergraph embedding models. Although the models we tried in this paper did not show any change, different embedding models can represent entities and especially relations in a myriad of ways, some of which should vastly outperform the others. In particular, a character based embedding may help to address the issue of formatting phrases, by being less particular about the presence or absence of underscores.

It would also be worthwile to look into a method of preprocessing the NLI dataset to isolate phrases and format them ahead of time to match their appearance in Concept Net. Concept Net is a large graph containing a large amount of semantic world knowledge. If we could specifically tailor our task to match Concept Net, much more of that information could likely be used.

We would also like to look into ways of incorporating information from knowledge graphs into language models beyond introducing new triples and generating a graph embedding. The original intent was to build off the work of nonce2vec (Herbelot and Baroni 2017) and from an existing set of graph embeddings, additionally selectively train the remainder of a desired vocabulary using word2vec. This approach was hidered by the requirement of a word2vec model to base training off of, instead of just vectors, where knowledge embedding vectors would have been easily substitutable. Should a solution to this issue be found, a definition dataset, required for nonce2vec, has already been prepared from a general corpus. There has also been discussion of simply concatonating the representation for each word created by glove and by a knowledge embedding into a single vector to be fed into the downstream model. This method should also capture semantic information from the knowledge graph, but would require modification of the downstream architecture and involve higher dimensional vectors.

## Acknowledgement

The work reported in this paper is supported by the National Science Foundation under Grant No. 1659788. Any opinions, findings and conclusions or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

Athiwaratkun, B., and Wilson, A. G. 2018. Hierarchical density order embeddings. *arXiv preprint arXiv:1804.09843*.

Athiwaratkun, B.; Wilson, A. G.; and Anandkumar, A. 2018. Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv:1806.02901*.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, B.; Wang, Q.; Wang, B.; and Guo, L. 2018. Improving knowledge graph embedding using simple constraints. *arXiv preprint arXiv:1805.02408*.

Han, X.; Cao, S.; Lv, X.; Lin, Y.; Liu, Z.; Sun, M.; and Li, J. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*, 139–144.

Herbelot, A., and Baroni, M. 2017. High-risk learning: acquiring new word vectors from tiny data. *arXiv preprint arXiv:1707.06556*.

Khodak, M.; Saunshi, N.; Liang, Y.; Ma, T.; Stewart, B.; and Arora, S. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. *arXiv preprint arXiv:1805.05388*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Talman, A.; Yli-Jyrä, A.; and Tiedemann, J. 2019. Sentence embeddings in nli with iterative refinement encoders. *JNLE*.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2071–2080.

Yang, B., and Mitchell, T. 2019. Leveraging knowledge bases in lstms for improving machine reading. *arXiv preprint arXiv:1902.09091*.