Improving Topic Modeling for Textual Content with Knowledge Graph Embeddings

Marco Brambilla, Birant Altinel
Politecnico di Milano, DEIB
Piazza Leonardo da Vinci, 32. I-20133 Milano, Italy
{firstname.lastname}@polimi.it

Abstract

Topic modeling techniques has been applied in many scenarios in recent years, spanning textual content, as well as many different data sources. The existing researches in this field continuously try to improve the accuracy and coherence of the results. Some recent works propose new methods that capture the semantic relations between words into the topic modeling process, by employing vector embeddings over knowledge bases. In this paper we study various dimensions of how knowledge graph embeddings affect topic modeling performance on textual content. In particular, the objective of the work is to determine which aspects of knowledge graph embedding have a significant and positive impact on the accuracy of the extracted topics. In order to obtain a good understanding of the impact, all steps of the process are examined and various parameterization of the techniques are explored. Based on the findings, we improve the state of the art with the use of more advanced embedding approaches and parameterizations that produce higher quality topics. The work also include a set of experiments with 2 variations of the knowledge base, 7 embedding methods, and 2 methods for incorporation of the embeddings into the topic modeling framework, also considering a set of variations of topic number and embedding dimensionality.

Introduction

In the current age of information, larger and larger amounts of data are generated and collected every second around the world. A significant portion of this data is in the form of textual content.The need for understanding this vast amount of textual content keeps increasing as everything in the world becomes more data-driven but mostly because of the fact that it's impossible for us to do it manually.

The fields of Natural Language Processing and Machine Learning offer automated to understand large amounts of textual data. Vector representations of words (Mikolov et al. 2013) (Řehůřek and Sojka 2010) (Pennington, Socher, and Manning 2014) (Joulin et al. 2016) have been used for many Natural Language Processing tasks such as syntactic parsing (Socher et al. 2013a) and sentiment analysis (Socher et al. 2013b), and it also is being used in the Topic Modeling field (Hinton and Salakhutdinov 2009)(Srivastava, Salakhutdinov, and Hinton 2013)(Cao et al. 2015)(Nguyen et al. 2015)(Yao et al. 2017). One of these papers with a method called KGE-LDA (Yao et al. 2017) aims to improve the performance of topic modeling by obtaining the vector representations of words from external knowledge bases such as WordNet (Miller 1995) and FreeBase (Bollacker et al. 2008) instead of learning them from documents. According to their reported results, this approach is successful and improves the topic coherence by 9.5% to 44% and document classification accuracy by 1.6% to 5.4% compared to LDA (Blei, Ng, and Jordan 2003).

Their approach improves the results with one specific method to obtain the word representations, but it's not clear whether vectors obtained through other methods that can capture better semantics of networks are able to boost the accuracy of topic modeling. The vector embedding methods that have proven to be more successful in other fields such as Link Prediction can possibly capture the semantics of the external knowledge base more accurately.

Another question that remains to be answered in this context is whether a larger knowledge base in terms of entities or a denser knowledge base in terms of relations between entities can also contribute to better representations of words. The primary motive to this question lies in the fact that the knowledge graphs do not have the complete semantic representation of the real world, and can be improved with different relations between entities.

This paper presents two approaches to improve Topic Modeling. The first approach applies various **Multi-relational Network Embedding Methods** by computing the vectors on the same network, and incorporating the results into the topic modeling framework that has been taken as the base method of this work. The mentioned embedding methods all follow a translation-based approach to vectors with incremental improvements over the original work which is TransE (Bordes et al. 2013). Since knowledge embeddings are increasingly used for topic modeling, there is lack of a comprehensive study that discovers the effects of knowledge encoded by various methods. Therefore, the primary motive of this work is to push the state of the art in this field forward by the application of more advanced methods and knowledge bases for obtaining better knowledge graph

embeddings in order to improve topic modeling.

The second approach **modifies the network of the knowledge graph** itself, and manages to significantly increase the density of the network by adding syntactic dependency relations between words in a sentence that are computed from the same text corpus used for the topic modeling. This combination is performed by computing the dependency trees of the sentences in the text corpus, and adding each relation to the knowledge graph between the corresponding entities, thus updating and enlarging the network. It studies the knowledge encoded by this denser network in terms of relations between entities, and how it affects the overall performance of embeddings, and consecutively topic modeling.

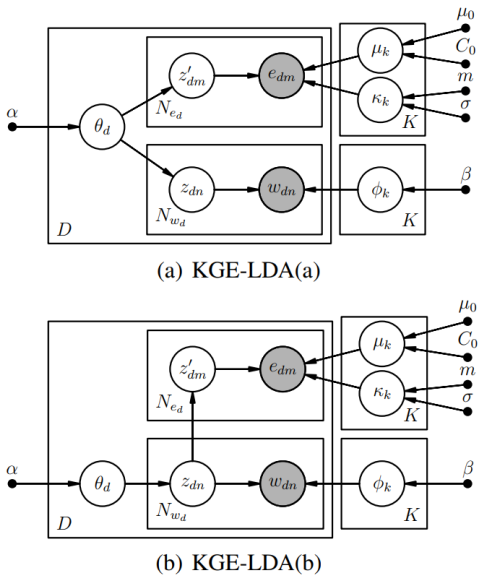
The paper is organized as follows: Section 2 presents the related works. Section 3 contains the details of the employed methods that are used in this paper. Section 4 describes the source codes and implementations of the used methods. Section 5 presents the results of the experiments and discusses these outcomes. Section 6 concludes and draws some possible future work.

Related Work

In this section, the existing works in the literature that are discussed constitute the basis for the main focus and direction of this paper. KGE-LDA (Yao et al. 2017) is directly the baseline work about topic modeling with knowledge graph embeddings that this paper is focused on. On the other hand, LF-LDA (Nguyen et al. 2015) is an older method that introduced the idea of using embeddings of words to improve topic modeling. The discussion of these methods is aimed towards creating a general perspective for the main idea and experiments that are proposed in this paper.

KGE-LDA (Yao et al. 2017) is a knowledge-based topic model that combines the well-known LDA model with entity embeddings obtained from knowledge graphs. It proposes two topic models that incorporate the vector representations of words, by obtaining them from the knowledge bases such as WordNet (Miller 1995) and Freebase (Bollacker et al. 2008). The two topic models are based on the previous works CI-LDA (Newman, Chemudugunta, and Smyth 2006) and Corr-LDA (Blei and Jordan 2003). The contributions of this paper create the foundations that this paper studies and attempts to improve. In this paper, the topic models of KGE-LDA are used. Their claim and results show that knowledge encoded from the knowledge graphs capture the semantics better than the compared methods. In order to handle the embeddings, they propose a Gibbs Sampling inference method.

KGE-LDA extends two entity topic models, namely CI-LDA (Newman, Chemudugunta, and Smyth 2006) and Corr-LDA (Blei and Jordan 2003) in order to incorporate the learned entity embeddings into the topic model. The model based on CI-LDA is referred to as **KGE-LDA(a)** and the model based on Corr-LDA is referred to as **KGE-LDA(b)** in the paper and also throughout this work. The details regarding these approaches are discussed in the following subsections. The graphical representation of the models can be seen in Figure 1.



(a) KGE-LDA(a)

(b) KGE-LDA(b)

Figure 1: The representation of both KGE-LDA(a) and KGE-LDA(b) models (Yao et al. 2017)

LF-LDA, which stands for Latent Feature LDA, aims to improve topic modeling by incorporating latent feature vectors with a similar point of view as KGE-LDA. The difference is that, apart from being published before KGE-LDA, this paper obtains the latent feature representations directly from the text corpus itself. It uses the famous word2vec (Mikolov et al. 2013) method to compute the embeddings on a large text corpus, to be used later on a smaller corpus for topic modeling. Its main contribution that is relevant to this paper consists of using a large external data to compute the word embeddings. LF-LDA extends two topic models, LDA (Blei, Ng, and Jordan 2003) and DMM (Nigam et al. 2000) by adding a latent feature component to the Dirichlet multinomial component that generates the words from topics in each topic model (Nguyen et al. 2015). The extended methods are called LF-LDA and LF-DMM. The graphical representation of LF-LDA can be seen in the Figure 2.

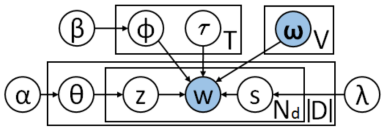


Figure 2: Representation of the LF-LDA model (Nguyen et al. 2015)

Improving Knowledge Graph Embeddings for Topic Modeling

The focus of this paper is to explore the improvements in knowledge graph embeddings and their effects in topic modeling performance. There are three explored dimensions

in the knowledge graph embedding process that are presumed to have direct effect on performance. These dimensions are embedding method performance, the information in the knowledge base, and the vector dimension of the embeddings. This chapter describes these dimensions and how to explore them.

Embedding Methods Application

The following models are chosen for running the experiments. TransE (Bordes et al. 2013) is the model used by the authors of KGE-LDA (Yao et al. 2017), whereas the following models of the respective papers are chosen because they are either directly or indirectly are compared with TransE and each other, which provides us with a better understanding of the difference in their performance.

The mentioned papers improve the state-of-the-art in knowledge graph embedding in their respective papers. The presumption is that the models which improve upon the results of TransE on other grounds such as Link Prediction, should also deliver similar improvements in Topic Modeling results. To create a comparison of equal grounds, all of these models should be trained with the same dataset, the same parameters, and produce an output of the same embedding dimension. By keeping all other variables same, it's possible to directly observe the quality of the embeddings for the purpose of topic modeling. The result of this approach helps determine the methods and the configurations which moves the state of the art further by producing the highest accuracy in topic modeling.

The following subsections explain the main characteristics and differences of the compared embedding methods:

TransE TransE model represents the relations in the graph as translations in the embedding space (Bordes et al. 2013). For example, in a triple (head, relation, tail); the vector arithmetic equation ***head + relation = tail*** should hold true. In this model, a null vector would represent the equivalence of the head and tail entity. This also means that if the semantics of the graph are captured correctly, the result of the vector arithmetic vector("France") - vector("Paris") + vector("Rome") should create a vector that is closest to the vector("Italy") in the knowledge graph (Mikolov et al. 2013), with the assumption that the triples (Paris, capitalof, France) and (Rome, capitalof, Italy) or similar semantic relations exist.

As stated before, TransE is part of the baseline method KGE-LDA that the following methods are compared to in the experiments.

TransH TransH model, models relations as hyperplanes in addition to the translation operations as TransE does (Wang et al. 2014). The motive is the fact that there are properties like reflexive, one-to-many, many-to-one and many-to-many; and there is a need to represent these mapping properties. Their claim is that TransE was not successful in preserving these properties.

DistMult This model also directly aims to improve on TransE model, and the main difference is the composition of vectors. Different from TransE where vectors are composed by addition as explained in previous subsections **(head + relation = tail)**; DistMult composes vectors by weighted element-wise dot product, in other words the following multiplicative operation: **head** x **relation = tail** (Yang et al. 2014).

TransR TransR model attempts to tackle the problem that the same semantic space to model embeddings for all entities and relations is insufficient (Lin et al. 2015b). Building on TransE and TransR; it build entity and relation embeddings in seperate semantic spaces.

PTransE PTransE builds upon the previous methods by utilizing multiple-step relation paths in the knowledge graph. Their approach is similar to TransE, with the addition of relation path-based learning (Lin et al. 2015a). In a few simple words, they join consecutive relations in the path into a single relation such as **relation1** $\circ$ **relation2 = relation path** and use these paths in the model.

HolE Short for "Holographic Embeddings", the difference that this model adopts is the learning of the compositional vector space representation of entire knowledge graphs (Nickel et al. 2016). It uses correlation as the compositional operator. The results of HolE are compared to TransE, TransR and other embedding methods in the published paper.

One interesting fact is that, HolE was proved to be equivalent to another method called ComplEx (Trouillon et al. 2016), which was also published the same year (Hayashi and Shimbo 2017). Because of this fact, ComplEx was excluded in this work from the experimentation.

Analogy Analogy proposes the optimization of latent feature representations with respect to the analogical properties of the embeddings of both entities and relations (Liu, Wu, and Yang 2017). It also unifies several methods in multi-relational embedding which are DistMult (Yang et al. 2014), ComplEx (Trouillon et al. 2016) and HolE (Nickel et al. 2016). It's also compared to all previous methods mentioned in this paper in the experiments of the published paper.

In Table 1, the time and space complexities along with the scoring functions of the described methods are compared.

Knowledge Graph Extension with Dependency Trees

While the previous two sections observe the effects of the embedding models and process; this section focuses on the density and quality of the knowledge graphs, which the embedding models are trained with.

Therefore, as a source of new information for the knowledge graph, the text corpus itself is a great answer. The dependency relations in sentences constitute meaningful semantics, and a quite massive source of information. The question that remains to be answered is the fact that are semantic relations in a knowledge graph and a dependency graph are compatible with each other? Are they able to create a richer knowledge base? Are the current embedding methods able to capture the information encoded in the resulting massive graph?

Table 1: Characteristics of the different Embedding Methods. Parameters: d: Embedding size, n_e: Number of entities, n_r: Number of relations, h: head entity, r: relation, t: tail entity, w_r: vector representation of r, p: path

	Time Complexity	Space Complexity	Scoring Function
TransE	$O(d)$	$O(n_e d + n_r d)$	$-\|h + r - t\|_{1/2}$
TransH	$O(d)$	$O(n_e d + 2n_r d)$	$-\|(h - w_r^\top h w_r) + r - (t - w_r^\top t w_r)\|_2^2$
DistMult	$O(d)$	$O(n_e d + n_r d)$	$h^\top diag(r) t$
PTransE	$O(d)$	$O(n_e d + n_r d)$	$-\|p - (t - h)\|$
TransR	$O(d^2)$	$O(n_e d + n_r d + n_r d^2)$	$-\|(M_r h) + r - (M_r t)\|_2^2$
HolE	$O(d \log d)$	$O(n_e d + n_r d)$	$r^\top (h \star t)$
Analogy	$O(d)$	$O(n_e d + n_r d)$	$h^\top M_r t$

To answer these questions, the Knowledge Graph used in this paper (WN18) was merged with the Dependency Graph obtained by the 20NG text corpus which is also used in this paper for topic modeling. As the details can be seen in the Datasets subsection of the Experiments section; the density of the graph increased about 5 times, which surely created a more complex semantic structure.

The general structure of the merging phase is illustrated in Figure 3. The process finds the dependency trees of each sentences. Then, the corresponding entity of each word in the knowledge graph is found. If the words and computed dependencies pass the filtering stage, a new link is added between the corresponding entities in the knowledge graph with the name of the dependency relation.

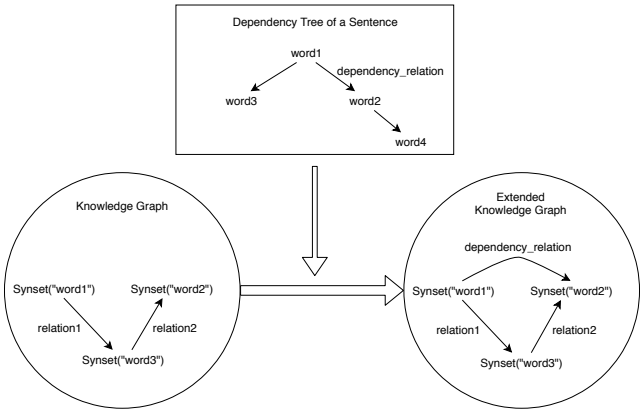


Figure 3: Visualization of the Knowledge Graph Extension

Further Exploration of Parameters

This section aims to increase the primary parameters to measure their effects on the final outcome. The motive is that as long as computational limits and feasibility allow, better parameters and settings should be used if it provides considerable improvements in the performance. In the light of this motive, the following aspects are considered.

The first aspect to be investigated is the effects of the embedding dimension on the Topic modeling performance. The motive for this aspect is the fact that the larger and denser the knowledge graph or the dataset gets, it creates more information to be stored in the embeddings. Larger vector dimensions offer more space to encode the semantics, but naturally it comes with performance costs.

Furthermore, the effects of topic number chosen for the topic model also has a direct effect on the performance. Considering the results in KGE-LDA (Yao et al. 2017), where the accuracy increases with topic number, a significantly increased topic number and its impact should be observed.

Lastly, the extended knowledge graph method that was described in the previous section should also be examined with the increased parameters as the information encoded from a larger graph might even provide greater performance with higher dimensional embeddings and higher topic numbers.

Implementation

Base Topic Modeling Framework

To merge learned embeddings with the process of the topic modeling, the original implementation of KGE-LDA by its authors was used[1]. The original implementation was chosen, because KGE-LDA is the baseline work that this paper follows; thus it's the best choice for running the experimentations.

The source code is structured as a Java project, and has a dependency for the Stanford CoreNLP library. Along with KGE-LDA, the project contains the implementations for LDA (Blei, Ng, and Jordan 2003) and CTM (Blei and Lafferty 2006). Several alterations and additions were made in the implementation for the third part of the experiments(Knowledge Graph Extension). The additions are as follows:

- Parsing 20NG dataset with the CoreNLP DependencyParser and to obtain dependency trees.
- Updating the WN18 graph with the obtained dependencies.
- Various minor alterations throughout the source code.

Embedding Methods

For the purpose of the experimentations for Embedding Method Comparison, the implementations of the chosen embedding methods were needed. Therefore, implementations of TransE, TransH, TransT and PTransE were taken from the open-source project KB2E[2]. The implementations of Dist-

[1] https://github.com/yao8839836/KGE-LDA

[2] https://github.com/thunlp/KB2E/

Mult, HolE and Analogy were taken from the open-source project OpenKE[3].

Dependency Parser

For the purpose of the Knowledge Graph Extension part of this paper, Stanford CoreNLP DependencyParser Annotator was used. Using DependencyParser, the code for the Knowledge Graph Extension part was implemented in Java. The process and the implementation follows this algorithm:

Algorithm 1: Knowledge Graph Extension with Dependency Trees

```
KnowledgeGraph ⟵ WN18;
DependencyNetwork ⟵ Empty Graph;
for Document d in 20NG do
    for Sentence s in d do
        t ⟵ DepencencyParser(s);
        DependencyNetwork append t;
    end
end
KnowledgeGraph merge DependencyNetwork;
return KnowledgeGraph;
```

To visualize how the dependency relations are merged with the knowledge graph, please refer to the Figure 4.

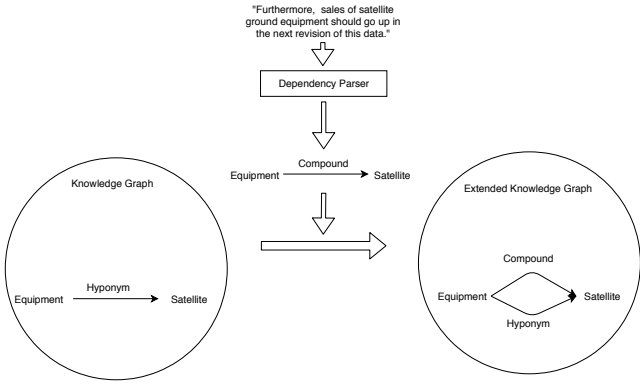


Figure 4: An Example of Merging a Dependency Relation from a Sentence with the Knowledge Graph

The example in the Figure 4 shows how a dependency relation extracted from a sentence updates the knowledge graph. In this specific example, there is a "Hyponym" relation from "Equipment" entity to "Satellite" entity in the knowledge graph. The dependency parser finds out that these two words are used in a compound in the corresponding sentence, and updates the knowledge graph with the "Compound" relation.

Experiments

In this section, a series of experiments that involve different methods and variations of parameters are presented. The used datasets along with the chosen parameters are stated for each of the different experiment sets.

The experiments are conducted to find answers to following questions:

1. Are newer and improved embedding models able to capture better semantics for the purpose of topic modeling?
2. How does the number of topics affect the performance of these sets of methods?
3. Does a denser and more complex knowledge base create a better or worse encoding of entities?
4. What is the importance of the vector dimensions in capturing and encoding information? Do we need larger vectors for more accurate representations for the used datasets?

The experiments are grouped into three categories that each try to answer the corresponding questions stated above. We proceed with three sets of experiments: (1) Embedding Method Application and Comparison; (2) Knowledge Graph Extension; (3) Further Exploration of Parameters.

Baselines

Two topic models are chosen to compare the results of experiments with:

- LDA (Blei, Ng, and Jordan 2003)
- KGE-LDA (Yao et al. 2017)

LDA was chosen as the primary indicator of performance, because it's the most widely used topic model which is considered as the baseline method for many other works in the field. KGE-LDA was chosen as the main indicator of performance since it is the baseline method and starting point of this paper work.

Datasets

Text Corpus The datasets in the context of this work refer to the text corpus that is used to run the topic models. For this purpose, 20-Newsgroups (20NG) dataset was used. The dataset includes 18,846 documents, split into 20 categories, with a vocabulary of 20,881 distinct words. In the text preprocessing phase, the following steps are applied to the data: Tokenization (with Stanford CoreNLP), stopwords removal, and rare words removal (for words that appear less than 10 times throughout the dataset).

External Knowledge The external knowledge refers to the knowledge graph that was used to train the representation learning methods to obtain the word embeddings. WN18, which is a subset of a widely used lexical knowledge graph WordNet, was used for this purpose. WN18 has the following characteristics in the training set: 141,442 triplets (the missing 10,000 triplets of WN18 are in the test and validation sets); 40,943 entities; 18 types of relations; 8,819 common entities with the 20NG vocabulary.

Table 2 shows the top 10 occurring relation types in the knowledge graph, their occurring counts, and percentages in size over the whole graph.

[3]https://github.com/thunlp/OpenKE

Table 2: Occurence Counts and Percentages of Top 10 Relations in the Original WN18 Dataset

Relation	Count	Percentage of Graph
Hyponym	34832	24.6%
Hypernym	34796	24.6%
Derivationally Related Form	29715	21.0%
Member Meronym	7402	5.23%
Member Holonym	7382	5.22%
Has part	4816	3.40%
Part of	4805	3.40%
Member of Domain Topic	3118	2.20%
Synset Domain Topic of	3116	2.20%
Instance Hyponym	2935	2.08%

Extended Knowledge Graph As mentioned before, the Knowledge Graph in the previous subsection was merged with the dependency graph obtained from the 20NG text corpus. The resulting graph has the following characteristics that have increased relative to the original knowledge graph(WN18):

- 817,568 triplets, with respect to the original 141,442;
- 55 types of relations, increased from the original 18.

There were new relations introduced to the knowledge graph, but no new entities. To demonstrate how the knowledge graph changed, here are the top 10 occurring relation types, their occurring counts, and percentages in size over the whole graph:

Table 3: Occurence Counts and Percentages of Top 10 Relations in the Extended WN18 Dataset

Relation	Count	Percentage of Graph
Root	30117	15.0%
Nominal Modifier	90531	11.1%
Compound	78654	9.62%
Direct Object	56423	6.90%
Adjectival Modifier	53819	6.58%
Dependent	35930	4.39%
Hyponym	34832	4.26%
Hypernym	34796	4.26%
Conjunct	33223	4.06%
Auxiliary	30775	3.76%

It can be seen that the structure of the knowledge graph has changed substantially, with the high number of additions. With the extension, the size of the graph grew by 578% compared to the original knowledge graph, and 37 new relation types were added.

Settings

A set of settings of the different parameters have been defined for the execution and validation of the approach. Some parameters have been adopted with a constant value across the experiments, while others have been varying across experiments. The settings considered include:

Table 4: Embedding Methods Comparison Settings

Parameter Name	Parameter Value
Embedding Dimension	50
Gibbs Sampling Iterations	1000
Learning Rate	0.001
Hyperparameter α	50/K (#Topics)
Hyperparameter β	0.01
Number of Topics (K)	20, 30, 40, 50

Table 5: Further Exploration Experiment Settings

Parameter Name	Parameter Value
Embedding Dimension	100
Topic Number	50, 100

1. Settings for Embedding Methods Comparison: all parameters have been fixed, except for the number of topics (and the respective parameter α), as reported in Table 4;

2. Settings for Knowledge Graph Extension: The settings are the same as the settings of Embedding Methods Comparison group.

3. Settings for Further Exploration of Parameters: with the aim of delving into detailed investigation of the parameter values, a further set of experiment with new variations of the settings have been launched, with values as reported in Table 5. With respect to the initial experiments (parametrized as in point 1 of this list), the embedding dimension is increased to 100 and the number of topics is increased to 100.

Results

The results are obtained through two different evaluation mechanisms, namely Topic Coherence and Document Classification. UCI method which uses Pointwise mutual information (Newman et al. 2010) was used for Topic Modeling, and LIBLINEAR linear classification library (Fan et al. 2008) was used for Document Classification. In the rest of the subsection, these results will be presented and discussed.

Embedding Methods Comparison

Topic Coherence Results As stated before PMI based topic coherence was used to obtain these results. To compute PMI, a dataset of 4,776,093 Wikipedia articles were used. For each method and topic, the results were run 5 times, after which the average and the standard deviation was calculated. The results can be found in Table 6.

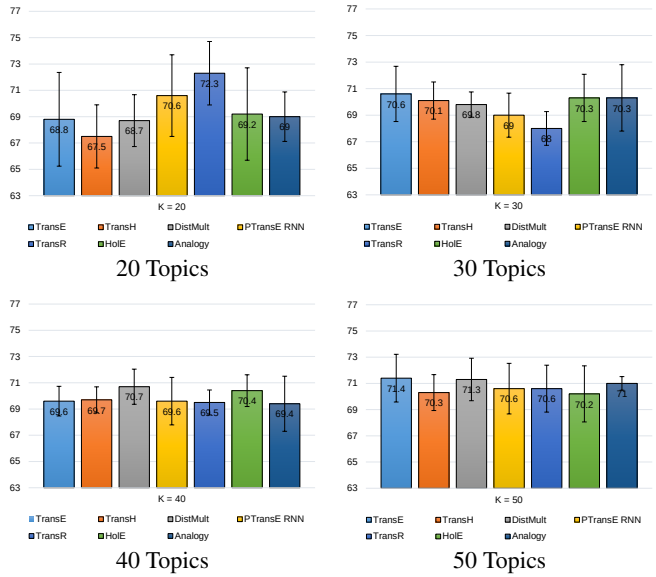


Figure 5: Topic Coherence Scores of Topic Modeling Obtained Through Different Embedding Methods with the Incorporation Model A, Separated by Topic Number

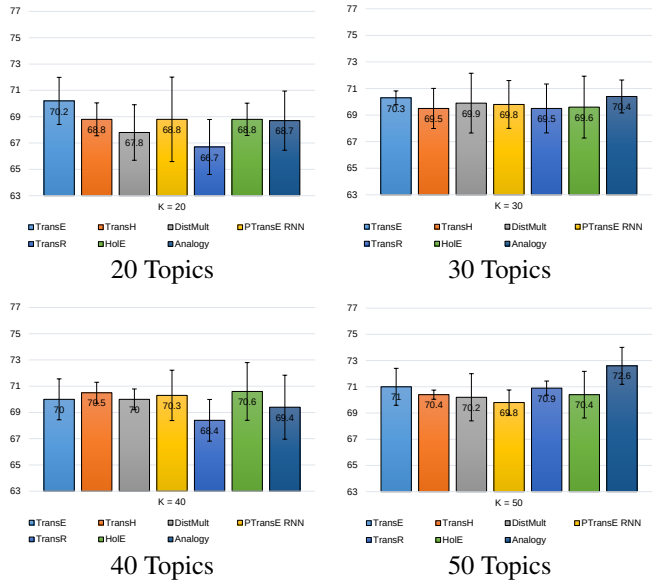


Figure 6: Topic Coherence Scores of Topic Modeling Obtained Through Different Embedding Methods with the Incorporation Model B, Separated by Topic Number

Table 6: Topic Coherence Results of Embedding Methods on Topic Modeling. The best results are reported in **bold**.

Model	K = 20	K = 30	K = 40	K = 50
LDA	68.4 ± 2.63	**72.5±1.87**	**70.9±1.74**	71.6±0.45
TransE(a)	68.8±3.56	70.6±2.08	69.6 ±1.13	71.4±1.82
TransE(b)	70.2±1.79	70.3±0.52	70 ±1.56	71±1.41
TransH(a)	67.5±2.4	70.1±1.4	69.7 ±0.99	70.3±1.37
TransH(b)	68.8±1.25	69.5±1.51	70.5 ±0.8	70.4±0.35
DistMult(a)	68.7±1.97	69.8±0.95	70.7±1.34	71.3±1.62
DistMult(b)	67.8±2.11	69.9±2.25	70 ±0.79	70.2±1.8
PTransE RNN(a)	70.6 3.1	69±1.66	69.6 ±1.81	70.6±1.93
PTransE RNN(b)	68.8±3.21	69.8±1.8	70.3 ±1.92	69.8±0.96
TransR(a)	**72.3 ±2.41**	68±1.27	69.5 ±0.95	70.6±1.79
TransR(b)	66.7±2.09	69.5±1.84	68.4 ±1.59	70.9±0.54
HolE(a)	69.2±3.51	70.3±1.78	70.4 ±1.21	70.2±2.14
HolE(b)	68.8±1.23	69.6±2.33	70.6 ±2.2	70.4±1.78
Analogy(a)	69±1.88	70.3±2.5	69.4 ±2.1	71±0.52
Analogy(b)	68.7±2.25	70.4±1.24	69.4 ±2.44	**72.6±1.41**

Overall Topic Coherence Results The best and second coherence scores for each topic number are different, and it should be noted that the performance of the original LDA is consistently good. TransR leads to more coherent topics with lower topic numbers, and Analogy performs best with higher topic numbers. The general trend shows improvement with higher topic numbers.

Model A on Topic Coherence For 30,40 and 50 topics the topic coherence results are close and in the same range with each other. The only significant visual difference in coherence can be observed with topic number 20 where we see TransR performing better than other methods. It is also worth mentioning that TransR performs best with the topic number of 20 than higher numbers, and performs worst on 30 topics. With 20 topics, the standard deviation seems to be higher than higher topic numbers with the best(TransR) and worst(TransH) scores of all the combinations.

Model B With Model B, there is also a general trend of improvement with topic number. The standard deviation in the general trend also gets smaller with increasing topic number. TransR scores the lowest on 20 topics, even though it scored the highest on 20 topics with Model A. The highest score combination is Analogy method with 50 topics.

Document Classification Results The documents have been classified using LIBLINEAR (Fan et al. 2008). For each method and topic, the results were run 5 times. The average and the standard deviation are reported in the Table 7 for each method and topic number.

Overall Document Classification Results The Table 7 show that in overall results with topic numbers 20,30,40 and 50; HolE and Analogy perform the best. Also on average, Model A results in slightly better scores than Model B; even though Analogy performs better with Model B. Another observation is that, performance almost always increases with topic number; noting that with 40 and 50 topics, the results are closer to each other than with other topic number increments.

Table 7: Classification Results of Embedding Methods on Topic Modeling. The best results are reported in **bold**.

Model	K = 20	K = 30	K = 40	K = 50
LDA	0.539± 0.028	0.633±0.022	0.695 ±0.022	0.69±0.022
TransE(a)	0.57± 0.024	0.677±0.013	0.705 ±0.011	0.694±0.017
TransE(b)	0.554±0.017	0.670±0.017	0.676 ±0.022	0.714±0.006
TransH(a)	0.567± 0.032	0.668± 0.027	**0.71 ±0.019**	0.714±0.009
TransH(b)	0.555±0.014	0.666±0.035	0.694 ±0.013	0.697±0.024
DistMult(a)	0.59±0.021	0.644±0.015	0.706±0.019	0.702±0.026
DistMult(b)	0.587±0.017	0.667±0.014	0.687 ±0.014	0.694±0.025
PTransE RNN(a)	0.567±0.024	0.667±0.024	0.701 ±0.012	0.709±0.010
PTransE RNN(b)	0.576±0.016	0.659±0.015	0.684 ±0.024	0.701±0.021
TransR(a)	0.574±0.012	0.656±0.018	0.687 ± 0.022	0.716±0.011
TransR(b)	0.555±0.035	0.662±0.022	0.692 ±0.005	0.695±0.026
HolE(a)	**0.597±0.032**	0.679±0.032	0.697 ±0.021	0.707±0.004
HolE(b)	0.563±0.022	0.668±0.034	0.684 ±0.026	0.713±0.017
Analogy(a)	0.579±0.014	0.641±0.037	0.704 ±0.022	0.715±0.009
Analogy(b)	0.554±0.004	**0.687±0.017**	0.676 ±0.022	**0.719±0.006**

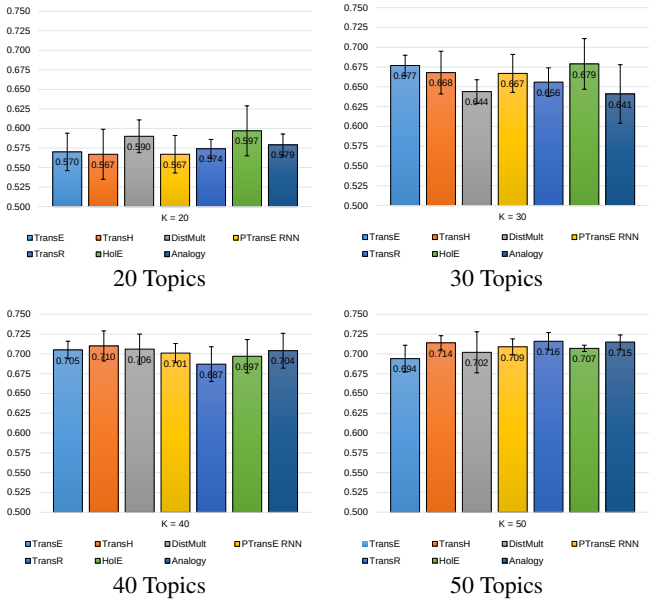


Figure 7: Document Classification Accuracy of Topic Modeling Obtained Through Different Embedding Methods with the Incorporation Model A, Separated by Topic Number

Model A The results of Model A show that in 20 topics, HolE and DistMult perform the best. Their approach apparently is better for small number of topics. Analogy also performs close to them. In 30 topics, the results show that HolE again scores best. However, this time DistMult scores low, and TransE, TransH and TransR which employ an addition based translation score better. In 40 topics, the performance of all methods converge, with all of them scoring more similarly than they do in other topic numbers. 50 topic results are also relatively similar, with TransR, Analogy and TransH scoring best.

The outcomes show that HolE is the best performer overall with Model A. Looking at the standard deviations, it seems that with Model A; the methods have similar consistency in their results.

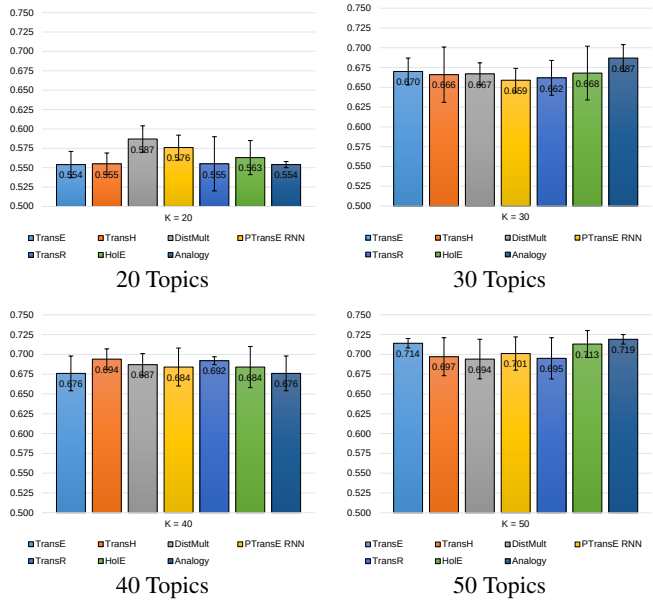


Figure 8: Document Classification Accuracy of Topic Modeling Obtained Through Different Embedding Methods with the Incorporation Model B, Separated by Topic Number

Model B The main difference of Model B generates the entity embeddings by topics in the same document, so it's important to state that the embeddings of the best methods are a better fit for this approach.

The results of Model B reveal that in 20 topics, DistMult is the best performer along with PTransE RNN. In 30 topics, Analogy outperforms others, as all the other methods score similar to each other. In 40 topics, TransH and TransR score better than others by a landslide. In 50 topics, Analogy seems to outperform others with TransE and HolE scoring close.

The outcomes show that Analogy and DistMult are the best performers overall with Model B. It's also important to note that Analogy gives more consistent results with multiple runs, which can be seen in lower standard deviation than other methods.

Knowledge Graph Extension

Topic Coherence Results The Topic Coherence experiments were run according to the parameters specified before. Each experiment was run 5 times, with averages and standard deviations reported in the Table 8.

Table 8: Topic Coherence Results with Knowledge Graph Extension. The best results are reported in **bold**.

	K = 20	K = 30	K = 40	K = 50
Orig. K.G. (a)	68.8±3.56	**70.6±2.08**	69.6±1.13	**71.4±1.82**
Orig. K.G. (b)	70.2±1.79	70.3±0.52	70±1.56	71±1.41
Ext. K.G. (a)	**70.5±2.44**	69.5±1.08	70.4±1.44	70.7±2.56
Ext. K.G. (b)	68.1±0.48	70.1±2.13	**71.5±3.01**	71.3±0.7

Topic Coherence with Knowledge Graph Extension Overview The results in Table 8 show that the Extended Knowledge Graph led to similar results with the Original Knowledge Graph. With an overall inspection of the table, it can be seen that the best performance are distributed to different models and graphs. The version with the Extended Knowledge Graph provided better average scores for 20 topics and 40 topics. Also, the overall trend is similar to the topic coherence results of the previous section, as 30, 40 and 50 topics resulted in the same range of performance with each other.

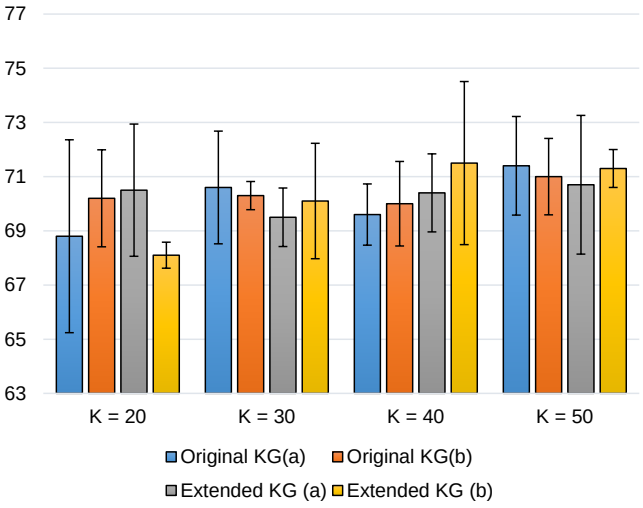


Figure 9: Topic Coherence Scores of Topic Modeling Obtained Through Original Knowledge Graph and Extended Knowledge Graph, On Different Topic Numbers

Document Classification Results The experiments in this section were also run 5 times as the ones before. The averages with the standard deviations are reported in Table 9.

Table 9: Document Classification Results with Knowledge Graph Extension. The best results are reported in **bold**.

	K = 20	K = 30	K = 40	K = 50
Orig. K.G. (a)	0.57±0.024	0.677±0.013	**0.705±0.011**	0.694±0.017
Orig. K.G. (b)	0.554±0.017	0.670±0.017	0.676±0.022	0.714±0.006
Ext. K.G. (a)	**0.582±0.017**	**0.683±0.032**	0.692±0.010	0.711±0.027
Ext. K.G. (b)	0.566±0.015	0.656±0.014	0.695±0.018	**0.716±0.010**

Document Classification with Knowledge Graph Extension Overview Results in Table 9 show that the knowledge graph extension created better semantics in the graph which in turn reflected to the classification results. We see an overall improvement with both Model A and Model B, whereas improvements with Model A are larger. Extended Graph with Model A performs better with smaller topic numbers, where as the extended graph with Model B is more accurate on larger topic numbers.

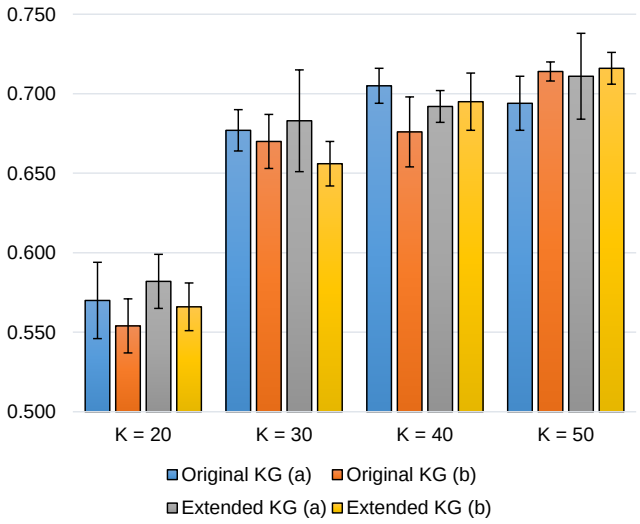


Figure 10: Document Classification Accuracies of Topic Modeling Obtained Through Original Knowledge Graph and Extended Knowledge Graph, on Different Topic Numbers

Increased Topic Number and Embedding Dimension The experiment in this section corresponds to the previous subsections about further exploration of parameters. For this purpose, an increased topic number of 100 and an increased embedding dimension of 100 was used with TransE and Analogy on the original knowledge graph, and furthermore TransE on Extended Knowledge Graph.

The average and standard deviations obtained from 5 runs of each combinations are reported in Tables 10 and 11.

Table 10: Topic Coherence Results with 100 Dimensional Embeddings. The best results are reported in **bold**.

	K = 50	K = 100
TransE on Orig. K.G. (a)	70.1±1.1	73.4±1.71
TransE on Orig. K.G. (b)	71.5±1.89	73.5±1.11
Analogy on Orig. K.G. (a)	69.4±0.82	72.7±1.73
Analogy on Orig. K.G. (b)	70±0.81	**75.1±2.21**
TransE on Ext. K.G. (a)	70.1±1.44	72.7±0.8
TransE on Ext. K.G. (b)	**71.5±0.19**	73.4±0.84

According to the Topic Coherence scores, the extended knowledge graph provides a better performance on 50 topics than both TransE and Analogy on the original graph. Even

though it scores the equal as the same configuration with Original Knowledge Graph, its standard deviation is 90% lower. On 100 topics, Analogy with Model B stands out with the highest coherence score that was obtained throughout the experiments of this paper work by scoring 2.18% higher than the closest coherence score. Figure 11 offers a clear comparison of these results in a visual way.

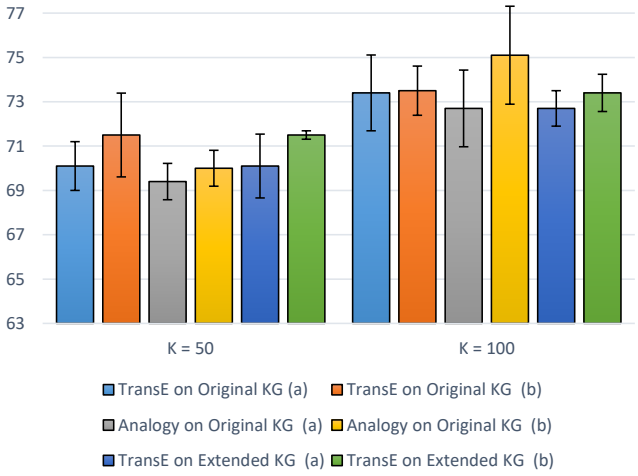


Figure 11: Topic Coherence Scores of Topic Modeling Obtained Through Specified Method and Knowledge Graph Combinations, with 100 Dimensional Embeddings on Different Topic Numbers

Table 11: Document Classification Results with 100 Dimensional Embeddings. The best results are reported in **bold**.

	K = 50	**K = 100**
TransE on Orig. K.G. (a)	0.712±0.020	0.725±0.009
TransE on Orig. K.G. (b)	0.705 ±0.009	0.724±0.006
Analogy on Orig. K.G. (a)	0.711 ±0.010	0.73±0.010
Analogy on Orig. K.G. (b)	0.706 ±0.010	0.727±0.010
TransE on Ext. K.G. (a)	**0.712 ±0.011**	**0.734±0.002**
TransE on Ext. K.G. (b)	0.693 ±0.019	0.726±0.013

The extended knowledge graph scores the highest Document Classification accuracy for both 50 topics and 100 topics with Model A. In fact, the Extended Graph with Model A on 100 topics scored the highest accuracy for Document Classification throughout the experiments of this paper by scoring 1.24% higher than the same configuration with the Original Knowledge Graph. On 50 topics, it scored the same average with the Original Knowledge Graph but with a smaller standard deviation. According to these results, the Extended Knowledge Graph leads to better accuracy than the Original Knowledge Graph with the exception of 50 topics with Model B. It also performs better than Analogy with Model A. These results can also be clearly seen in Figure 12.

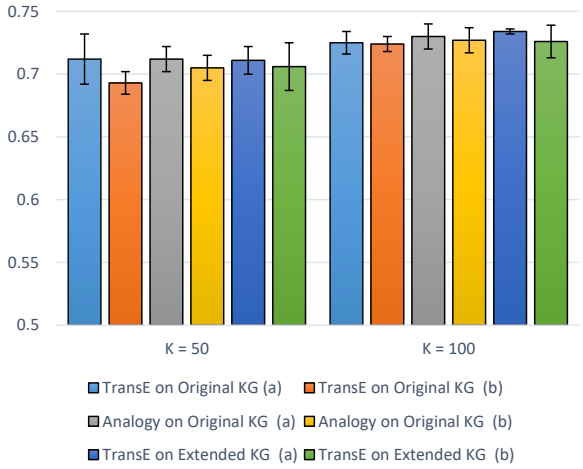


Figure 12: Document Classification Accuracy of Topic Modeling Obtained Through Specified Method and Knowledge Graph Combinations, with 100 Dimensional Embeddings on Different Topic Numbers

Runtime Duration

The experiments were conducted on a computer with the following relevant technical specifications:

- Intel Core i5-8250U CPU @ 1.60GHz
- 8 GB of DDR4 RAM @ 1866 MHz

Throughout the experiments, the elapsed execution time was measured. Embedding methods were run only once to obtain the representations from the knowledge graph. The fastest embedding happened to be TransE with approximately 1 hour of computation, and the slowest was HolE with approximately 17 hours of computation. All of other methods ran for a duration between 1 hour and 2 hours. It is safe to say that HolE was exceptionally slow during the training phase compared to other methods.

The more crucial and overall time consuming part was running the topic models with the obtained representations. The duration of topic modeling phase was not affected by the representations obtained by different methods, as they all provide an output of the same size. However, the topic number and embedding size had a significant effect on the execution time. The average durations are reported in two separate tables. For embedding size of 50 the results can be seen in Table 12 and for embedding size of 100 the results can be seen in Table 13.

Table 12: Average execution time of Topic Modeling with 50-dimensional embeddings (in minutes) depending on the number of topics K.

	K = 20	**K = 30**	**K = 40**	**K = 50**
Model A	133	142	164	189
Model B	121	145	162	216

Table 13: Average execution time of Topic Modeling with 100-dimensional embeddings (in minutes) depending on the number of topics K.

	K = 50	K = 100
Model A	217	340
Model B	209	314

To more clearly interpret the execution times, Figure 13 provides a visual representation. It can be seen in the figure on K = 50 that the runtime duration decreases by 3.3% with the embedding size with Model B, and increases 14.5% with Model A from 50 dimensional embeddings to 100 dimensional embeddings.

However an increase from 50 topics to 100 topics increases runtime duration by 79.9% with Model A and 45.4% with Model B. Considering these facts with the general trend of growth in the figure; it is safe to say that topic number has a larger impact on runtime duration than the embedding size during topic modeling.

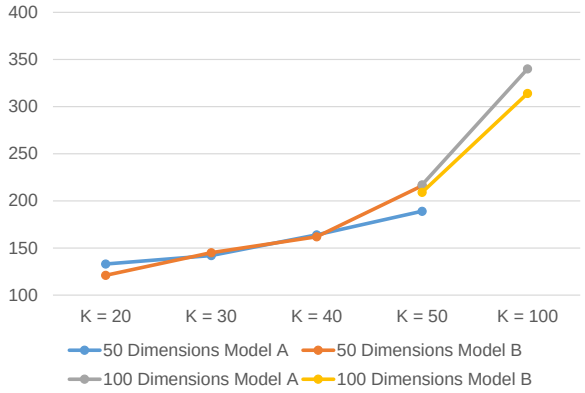


Figure 13: Execution times (in minutes).

Discussion

The results on topic coherence throughout the three experiments share a similar pattern. From 30 topics to upwards, the scores are really similar with the consideration of standard deviation, with a few results having significant difference. These scores also do not vary much between different methods both for the incorporation models A and B. For the results in 20 topics, we have larger difference between methods. With the increased parameters of 100 topics and 100 dimensional embeddings, the highest score achieved is 75.1±2.21 by Analogy, which scored 72.6±1.41 with 50 topics and 50 dimensional embeddings. Topic Coherence with 100 topics shows that Analogy with Model B configuration proves to be successful also on higher dimensional embedding and higher topic numbers.

Therefore, some inferences can be made for the effects of embedding methods on topic coherence: The coherence increases with topic number on average, but inconsistently. This means that a general trend of increase is seen, except on 40 topics which resulted in lower coherence scores in general than 30 topics. The usage of different embedding methods create topic coherence results that are in 1.2% range of each other on average. Analogy with Model B leads to the highest coherence scores with high topic numbers. The extended knowledge graph clearly improved the Document Classification accuracy with the exception of 40 topics. The improvements on Topic Coherence is on 20 and 40 topics.

For a general purpose use, Analogy is a clear choice over DistMult and HolE. The first reason is the fact that Analogy is a generalized method which can reproduce DistMult and HolE with a selection of parameters; it allows a higher range of performance and parameters. This should allow a grid search to find a configuration which is better than DistMult and HolE. The second reason is the fact that even though HolE and Analogy with the same parameters perform quite similar to each other, it takes much longer to train HolE ($\sim$ 17 hours) compared to Analogy($\sim$ 1-2 hours). A much faster training with theoretically being able to produce the same results as HolE, makes Analogy more feasible.

The Document classification Evaluation produced results that are clearer and easier to interpret in general. With small exceptions, increased topic number produced better results. In the embedding method comparison section, it can be seen that some of the newer and more complex embedding methods like DistMult, HolE and Analogy led to higher classification accuracy. Model A seems to be on average 1% better than Model B, but they produce equally consistent results with the same standard deviation at 1.9% on average.

On the other hand, there are clear improvements in the accuracy of the document classification when the Extended Knowledge Graph was used to train the embedding methods. This means that the semantic structure of the knowledge graph was enhanced, which reflected into better vector representations of entities and relations.

In the last group of experiments, the Extended Knowledge Graph provides better results than TransE and Analogy on the Original Knowledge Graph with an accuracy of 0.734 ± 0.002 which is the highest accuracy recorded throughout the experiments in this paper work.

In light of these outcomes, the following inferences are made for the effects of embedding methods on document classification. The accuracy consistently increases with topic number. Changes on the embedding method performance reflects on the document classification accuracy. Analogy with Model B leads to the highest accuracy scores on high topic numbers. The extended knowledge graph led to increased accuracy, and showed that dependency trees enhanced the semantics of the knowledge graph.

Conclusion

This paper explored the incorporation of knowledge graph embeddings into topic modeling, by experimenting on various aspects and identifying the ways for improvements. These aspects were the semantic information in the source knowledge graph, different embedding methods, performance effects of topic numbers and embedding dimensions. performance of 7 embedding methods, 2 topic models, 2 variations of the knowledge base and various parameters

have been explored in the context of Topic Modeling. 2 evaluation methods, namely Topic Coherence and Document Classification, have been used to measure the success of the experimentations. In the light of these results, this paper work has made several contributions.

On Embedding Methods Comparison, Topic Coherence and Document Classification yields different performance by each method, but the results have similarities. The most obvious pattern is the performance of Analogy. It outperforms all other methods on higher topic numbers with Model B. For lower topic numbers, simpler methods like TransE and TransR produce the best results. Overall, the best average scores come from HolE.

The Knowledge Graph Extension scores similar results to the original graph on Topic Coherence, but on Document Classification it clearly improves the accuracy. With increased parameters and embedding dimension, the improvements of the Knowledge Graph Extension are clearer, especially in Document Classification.

The best performing embedding method Analogy with Model B achieves an average improvement of 0.50% over the baseline method (KGE-LDA using TransE) in Topic Coherence, and an average improvement of 1.01% over the baseline method (KGE-LDA using TransE) in Document Classification. The Knowledge Graph Extension achieves an average improvement of 0.52% over the Original Knowledge Graph in Topic Coherence, and an average improvement of 0.77% over the Original Knowledge Graph in Document Classification.

As the closing remark, the best embedding method, incorporation model and parameter combination is Analogy with Model B on high topic numbers, with high embedding dimension. The extension of the knowledge base along with high embedding dimension enables more information to be encoded into the vectors, which in turn creates a more accurate representation of the entities compared to the Original Knowledge Graph. This performance improvement of the Extended Knowledge Graph comes with a 578% growth in the size of the graph.

It has been shown that Analogy is the most optimal embedding method Secondly, the results clearly show that the Extended Knowledge Graph has improved both Topic Coherence score and Document Classification accuracy.

Deeper investigations on a few points can provide further improvements on the solution. For the embedding method comparison part, the different methods have been tested with the same parameters. This provided an equal ground for the methods to compete with each other. However, a comprehensive parameter grid search for each embedding method can increase their performance and reveal more realistic values. Finally, as the specific knowledge graph extension in the experiments yielded better results, there can be further exploration on the knowledge graph capabilities.

References

[Blei and Jordan 2003] Blei, D. M., and Jordan, M. I. 2003. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 127–134. ACM.

[Blei and Lafferty 2006] Blei, D., and Lafferty, J. 2006. Correlated topic models. *Advances in neural information processing systems* 18:147.

[Blei, Ng, and Jordan 2003] Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

[Bollacker et al. 2008] Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. AcM.

[Bordes et al. 2013] 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.

[Cao et al. 2015] Cao, Z.; Li, S.; Liu, Y.; Li, W.; and Ji, H. 2015. A novel neural topic model and its supervised extension. In *AAAI*, 2210–2216.

[Fan et al. 2008] Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research* 9(Aug):1871–1874.

[Hayashi and Shimbo 2017] Hayashi, K., and Shimbo, M. 2017. On the equivalence of holographic and complex embeddings for link prediction. *arXiv preprint arXiv:1702.05563*.

[Hinton and Salakhutdinov 2009] Hinton, G. E., and Salakhutdinov, R. R. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, 1607–1614.

[Joulin et al. 2016] Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *CoRR* abs/1607.01759.

[Lin et al. 2015a] Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; and Liu, S. 2015a. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*.

[Lin et al. 2015b] Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015b. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, 2181–2187.

[Liu, Wu, and Yang 2017] Liu, H.; Wu, Y.; and Yang, Y. 2017. Analogical inference for multi-relational embeddings. *arXiv preprint arXiv:1705.02426*.

[Mikolov et al. 2013] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J.; Sutskever, L.; and Zweig, G. 2013. word2vec. *URL https://code. google. com/p/word2vec*.

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

[Newman et al. 2010] Newman, D.; Lau, J. H.; Grieser, K.; and Baldwin, T. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 100–108. Association for Computational Linguistics.

[Newman, Chemudugunta, and Smyth 2006] Newman, D.; Chemudugunta, C.; and Smyth, P. 2006. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 680–686. ACM.

[Nguyen et al. 2015] Nguyen, D. Q.; Billingsley, R.; Du, L.; and Johnson, M. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics* 3:299–313.

[Nickel et al. 2016] Nickel, M.; Rosasco, L.; Poggio, T. A.; et al. 2016. Holographic embeddings of knowledge graphs. In *AAAI*, volume 2, 3–2.

[Nigam et al. 2000] Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using em. *Machine learning* 39(2-3):103–134.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

[Řehůřek and Sojka 2010] Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA. `http://is.muni.cz/publication/884893/en`.

[Socher et al. 2013a] Socher, R.; Bauer, J.; Manning, C. D.; et al. 2013a. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 455–465.

[Socher et al. 2013b] Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

[Srivastava, Salakhutdinov, and Hinton 2013] Srivastava, N.; Salakhutdinov, R.; and Hinton, G. 2013. Fast inference and learning for modeling documents with a deep boltzmann machine.

[Trouillon et al. 2016] 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.

[Wang et al. 2014] Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, volume 14, 1112–1119.

[Yang et al. 2014] Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

[Yao et al. 2017] Yao, L.; Zhang, Y.; Wei, B.; Jin, Z.; Zhang, R.; Zhang, Y.; and Chen, Q. 2017. Incorporating knowledge graph embeddings into topic modeling. In *AAAI*, 3119–3126.