Embedded Topic Models Enhanced by Wikification

Takashi Shibuya **Takehito Utsuro**
Degree Programs in Systems and Information Engineering,
Graduate School of Science and Technology, University of Tsukuba
s2430171@_u.tsukuba.ac.jp utsuro@_iit.tsukuba.ac.jp

Abstract

Topic modeling analyzes a collection of documents to learn meaningful patterns of words. However, previous topic models consider only the spelling of words and do not take into consideration the homography of words. In this study, we incorporate the Wikipedia knowledge into a neural topic model to make it aware of named entities. We evaluate our method on two datasets, 1) news articles of *New York Times* and 2) the AIDA-CoNLL dataset. Our experiments show that our method improves the performance of neural topic models in generalizability. Moreover, we analyze frequent terms in each topic and the temporal dependencies between topics to demonstrate that our entity-aware topic models can capture the time-series development of topics well.

1 Introduction

Probabilistic topic models such as latent Dirichlet allocation (LDA) (Blei et al., 2003) and embedded topic model (ETM) (Dieng et al., 2020) have been utilized for analyzing a collection of documents and discovering the underlying semantic structure. Such topic models have also been extended to dynamic topic models (Blei and Lafferty, 2006; Hida et al., 2018; Dieng et al., 2019; Cvejoski et al., 2023), which can capture the chronological transition of topics, motivated by the fact that documents (such as magazines, academic journals, news articles, and social media content) feature trends and themes that change with time.

However, previous (dynamic) topic models consider only the spelling of words and do not take into consideration the homography of words such as "*apple*" and "*amazon*". We hypothesize that this unawareness of the word homography harms the performance of topic models because one meaning of a word will tend to be used in some specific topics but another meaning of the same spelled word will appear in other topics more frequently. For instance, the entity "Amazon.com" will tend to appear in business news or technology articles, whereas documents about the environment will discuss the entity "Amazon rainforest" more often than "Amazon.com". Although the word "*Amazon*" can thus refer to a different entity depending on a context, existing topic models are not aware of such homography of the word "*Amazon*" and regard the word as unique.

To address the above issue, we propose a method of analyzing a collection of documents based on entity knowledge on Wikipedia. Our proposed method relies on two technologies: 1) entity linking (wikification) and 2) entity embedding (Wikipedia2Vec (Yamada et al., 2020)). Entity linking (wikification) is a natural language processing technique that assigns an entity mention in a document to a specific entity in a target knowledge base (Wikipedia). For example, an entity linker can recognize which a word "apple" in a document means, "Apple Inc.", "Big Apple", or another. We adopt entity linking as a preprocessing of topic modeling. Next, we incorporate entity embeddings (vector representations of entities in a knowledge base) into a neural topic model according to the result of the entity linking. Previous neural topic models utilize only conventional word embeddings, which are unaware of the homography of words. On the other hand, our proposed method uses not only word embeddings but also entity embeddings, which enables neural topic models to distinguish between multiple entities that share their spelling. We hypothesize that our entity-aware method improves the performance of neural topic models. We empirically show the effectiveness of our method on two datasets: 1) a collection of news articles of *New York Times* published between 1996 and 2020 and 2) the AIDA-CoNLL dataset (Hoffart et al., 2011). We adopt two topic models, ETM and dynamic ETM (Dieng et al., 2019), as baselines and quantitatively show that entity linking

Proceedings of the First Workshop on Advancing Natural Language Processing for Wikipedia, pages 80–90

improves the performance of neural topic models. Furthermore, we demonstrate that topics and their temporal change extracted by trained dynamic topic models are reasonable by manually analyzing frequent terms of each topic. We summarize our contributions as follows:

- We propose a method to make neural topic models aware of named entities. Our method utilizes entity linking (wikification) as preprocessing and incorporates entity embeddings (Wikipedia2Vec) into neural topic models.
- We quantitatively demonstrate that our proposed method improves the performance of neural topic models on a dataset containing many homographic words such as "apple".
- We manually analyze topics extracted by trained topic models and verify that our proposed method brings high interpretability because frequent terms in each topic are expressed with Wikipedia entries.
- We also show that our method does not harm the performance even on a dataset that does not include many homographic words (if entity linking is accurate enough).

2 Related Work

2.1 Neural Topic Models

Our method builds on a combination of topic models and word embeddings, following a surge of previous methods that leverage word embeddings to improve the performance of probabilistic topic models. Some methods incorporate word similarity into the topic model (Petterson et al., 2010; Xie et al., 2015; Zhao et al., 2017). Other methods combine LDA with word embeddings by first converting the discrete text into continuous observations of embeddings (Das et al., 2015; Batmanghelich et al., 2016; Xun et al., 2016, 2017). Another line of research improves topic modeling inference utilizing deep neural networks (Cong et al., 2017; Zhang et al., 2018; Card et al., 2018). These methods reduce the dimension of the text data through amortized inference and the variational auto-encoder (Kingma and Welling, 2014). Finally, Dieng et al. (2020) proposed the embedded topic model (ETM) that makes use of word embeddings and uses amortization in its inference procedure.

2.2 Dynamic Topic Models

The seminal work of Blei and Lafferty (2006) introduced dynamic latent Dirichlet allocation (D-LDA), which uses a state space model on the parameters of a topic distribution, thus allowing the distribution parameters to change with time. Dieng et al. (2019) proposed an extension of D-LDA, dynamic embedded topic model (D-ETM), that better fits the distribution of words via the use of distributed representations for both the words and the topics. Furthermore, Miyamoto et al. (2023) introduced the self-attention mechanism into the neural network used in amortized variational inference.

2.3 Entity Embeddings

Entity embeddings have been studied mainly in the context of named entity disambiguation (NED). Bordes et al. (2011); Socher et al. (2013); Lin et al. (2015) focus on knowledge graph embeddings and propose vector representations of entities to primarily address the knowledge base (KB) link prediction task. Wang et al. (2014) proposed the joint modeling of the embedding of words and entities and revealed that such joint modeling improves performance in several entity-related tasks including the link prediction task. Yaghoobzadeh and Schütze (2015) built embeddings of words and entities on a corpus with annotated entities using the skip-gram model to address the entity typing task. Finally, Yamada et al. (2016) proposed an embedding method that consists of three models: 1) the conventional skip-gram model that learns to predict neighboring words given the target word in text corpora, 2) the anchor context model that learns to predict neighboring words given the target entity using anchors and their context words in the KB, and 3) the KB graph model that learns to estimate neighboring entities given the target entity in the link graph of the KB. To the best of our knowledge, our study is the first attempt to incorporate entity embeddings into embedded topic models.

2.4 Topic Models with Wikipedia

There have been several works where topic models are applied to Wikipedia. Most such studies worked on cross-lingual topic modeling by harnessing Wikipedia's cross-linguality (Ni et al., 2009; Boyd-Graber and Blei, 2009; Zhang et al., 2013; Hao and Paul, 2018; Piccardi and West, 2021). In Wikipedia, each article describes a concept, and each concept is usually described in multiple

languages. They proposed formulations of cross-lingual topic models and verified the efficacy of their proposed topic models trained on Wikipedia articles and links. Aside from the above studies, Miz et al. (2020) applied topic models to Wikipedia for analyzing popular topics in different language editions. In contrast to these works, our method utilizes Wikipedia entities identified by entity linking to make embedded topic models capable of dealing with the homography of words in arbitrary documents.

3 Topic Models

Here, we review topic models on which our method is based: LDA, ETM, D-ETM. In the following, we consider a collection of D documents, where the vocabulary contains V distinct terms. Let $w_{dn} \in \{1, \dots, V\}$ denote the n-th word in the d-th document.

3.1 Latent Dirichlet Allocation (LDA)

LDA is a probabilistic generative model of documents (Blei et al., 2003). It posits K topics, and the distribution over the vocabulary for each topic k is represented $\boldsymbol{\beta}_k \in \mathbb{R}^V$. It assumes each document comes from a mixture of topics, where the topics are shared across the given documents and the mixture proportions are unique for each document. Specifically, LDA considers a vector of topic proportions $\boldsymbol{\theta}_d \in \mathbb{R}^K$ for each document d; each element θ_{dk} expresses how prevalent the k-th topic is in the document d. In the generative process of LDA, each word is assigned to topic k with the probability θ_{dk}, and the word is then drawn from the distribution $\boldsymbol{\beta}_k$. The generative process for each document is as follows:

1. Draw topic proportion: $\boldsymbol{\theta}_d \sim \text{Dirichlet}(\boldsymbol{\eta}_\theta)$
2. For each word n in d:
   (a) Draw topic assignment: $\boldsymbol{z}_{dn} \sim \text{Cat}(\boldsymbol{\theta}_d)$
   (b) Draw word: $w_{dn} \sim \text{Cat}(\boldsymbol{\beta}_{\boldsymbol{z}_{dn}})$.

Here, $\text{Cat}(\cdot)$ denotes a categorical distribution. LDA places a Dirichlet prior on the topics, $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\boldsymbol{\alpha}_\beta)$. The two concentration parameters of the Dirichlet distributions, $\boldsymbol{\alpha}_\beta$ and $\boldsymbol{\eta}_\theta$, are fixed model hyperparameters.

3.2 Embedded Topic Model (ETM)

ETM (Dieng et al., 2020) is a neural topic model powered by word embeddings (Mikolov et al., 2013) and a neural network. Here, let $\boldsymbol{\rho}$ be an $L \times V$ matrix, which contains L-dimensional embeddings of the words in the vocabulary. Each column $\boldsymbol{\rho}_v \in \mathbb{R}^L$ corresponds to the embedding of the v-th term. ETM uses this embedding matrix $\boldsymbol{\rho}$ to define the word distribution of each topic, $\boldsymbol{\beta}_k = \text{softmax}(\boldsymbol{\rho}^\top \boldsymbol{\alpha}_k)$. $\boldsymbol{\alpha}_k$ is an embedding representation of the k-th topic in the semantic space of words, called topic embedding. The generative process of ETM is analogous to LDA as follows:

1. Draw topic proportion: $\boldsymbol{\theta}_d \sim \mathcal{LN}(\boldsymbol{0}, \boldsymbol{I})$
2. For each word n in d:
   (a) Draw topic assignment: $\boldsymbol{z}_{dn} \sim \text{Cat}(\boldsymbol{\theta}_d)$
   (b) Draw word: $w_{dn} \sim \text{Cat}(\boldsymbol{\beta}_{\boldsymbol{z}_{dn}})$.

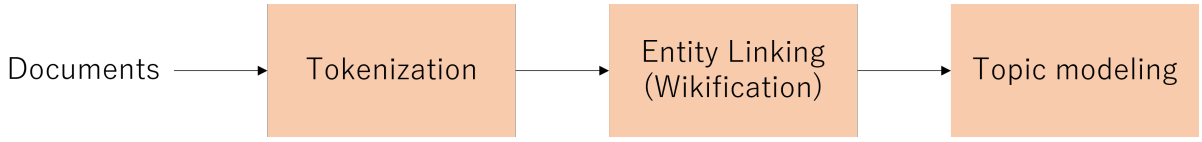
Here, $\mathcal{LN}(\cdot, \cdot)$ denotes a logistic-normal distribution (Atchison and Shen, 1980). The intuition behind ETM is that the embedding representations of semantically related words are similar to each other, they will interact with the topic embeddings $\boldsymbol{\alpha}_k$ similarly, and then they will be assigned to similar topics.

3.3 Dynamic Embedded Topic Model (D-ETM)

D-ETM (Dieng et al., 2019) analyzes time-series documents by introducing Markov chains to the topic embeddings $\boldsymbol{\alpha}_k$ and the topic proportion mean. As in ETM, D-ETM considers an embedding matrix $\boldsymbol{\rho} \in \mathbb{R}^{L \times V}$, such that each column $\boldsymbol{\rho}_v \in \mathbb{R}^L$ corresponds to the embedding of the v-th term. D-ETM posits an topic embedding $\boldsymbol{\alpha}_k^{(t)} \in \mathbb{R}^L$ for each topic k at a time stamp $t \in \{1, \dots, T\}$. This means D-ETM represents each topic with a time-varying vector. Then, the word distribution for the k-th topic in the time step t is defined by $\boldsymbol{\beta}_k^{(t)} = \text{softmax}(\boldsymbol{\rho}^\top \boldsymbol{\alpha}_k^{(t)})$. Here, the generative process of D-ETM for documents is described as follows:

1. For time step $t = 0$:
   (a) Draw initial topic embedding:
   $\boldsymbol{\alpha}_k^{(0)} \sim \mathcal{N}(\boldsymbol{0}, I)$ for $k \in \{1, \dots, K\}$
   (b) Draw initial topic proportion mean:
   $\boldsymbol{\eta}_0 \sim \mathcal{N}(\boldsymbol{0}, I)$
2. For each time step $t \in \{1, \dots, T\}$:
   (a) Draw topic embedding:
   $\boldsymbol{\alpha}_k^{(t)} \sim \mathcal{N}(\boldsymbol{\alpha}_k^{(t-1)}, \sigma^2 I)$
   for $k \in \{1, \dots, K\}$

(b) Conventional topic models



(b) Our proposed topic model

Figure 1: Processing flows of conventional topic models and our proposed topic model.

(b) Draw topic proportion mean:
$\boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{\eta}_{t-1}, \delta^2 I)$

3. For each document $d \in \{1, \ldots, D\}$:

(a) Draw topic proportion:
$\boldsymbol{\theta}_d \sim \mathcal{LN}(\boldsymbol{\eta}_{t_d}, \gamma^2 I)$

(b) For each word n in d:

i. Draw topic assignment:
$\boldsymbol{z}_{dn} \sim \mathrm{Cat}(\boldsymbol{\theta}_d)$

ii. Draw word:
$w_{dn} \sim \mathrm{Cat}(\boldsymbol{\beta}_{\boldsymbol{z}_{dn}}^{(t_d)})$,

where $\mathcal{N}(\cdot, \cdot)$ denotes a normal distribution distribution. σ, δ, and γ are model hyperparameters, each of which controls the variance of the corresponding normal distribution. t_d denotes the time stamp of the document d. Step 2(a) encourages smooth variations of the topic embeddings, and Step 2(b) describes time-varying priors over the topic proportions $\boldsymbol{\theta}_d$.

In this study, we incorporate entity knowledge into ETM or D-ETM by utilizing not only word embeddings but also entity embeddings, which enables topic models to be aware of named entities. To the best of our knowledge, our study is the first attempt to apply entity embeddings to embedded topic models. In the next section, we will explain how we introduce entity embeddings into embedded topic models.

4 Proposed Method

In this study, we propose a method of incorporating word disambiguation results into a neural topic model. We depict the processing flows of conventional topic models and our proposed method in Figure 1. In previous embedded topic models such as ETM and D-ETM, given documents are first tokenized, and then the word embedding matrix $\boldsymbol{\rho}$ is built by tiling the pretrained word embeddings such as skip-gram (Mikolov et al., 2013) corresponding to tokenized words. On the other hand, we incorporate entity information extracted by entity linking (EL) into the word embedding matrix $\boldsymbol{\rho}$ of an ETM/D-ETM. We explain the details of our method below.

4.1 Incorporation of Entity Linking

Here, we explain a way of building the embedding matrix $\boldsymbol{\rho}$ based on EL results. EL is a task that assigns a unique identity to an entity mention in text. In this study, we use an entity embedding instead of a word embedding if an entity linker identifies a phrase in a document as an entry in a knowledge base (KB) as depicted in Figure 2. Specifically, we utilize entity embedding trained with the Wikipedia2Vec toolkit (Yamada et al., 2020). The Wikipedia2Vec toolkit can learn the embeddings of both words and entities by using Wikipedia's text and hyperlinks. We can incorporate distributed representations of not only words but also entities into neural models with them. For example, if a word "amazon" is identified as a KB entry "Amazon (company)" in a document, we adopt the entity embedding corresponding to "Amazon (company)". If "*amazon*" is identified as a KB entry "Amazon rainforest" in another document (or another place of the same document), we use the entity embedding for "Amazon rainforest". If "*amazon*" is not identified to any KB entry, we adopt the word embedding corresponding to "*amazon*". Thus we deal with the entity "Amazon (company)", the entity "Amazon rainforest", and the word "*amazon*" as distinct items. Through the above procedure, we

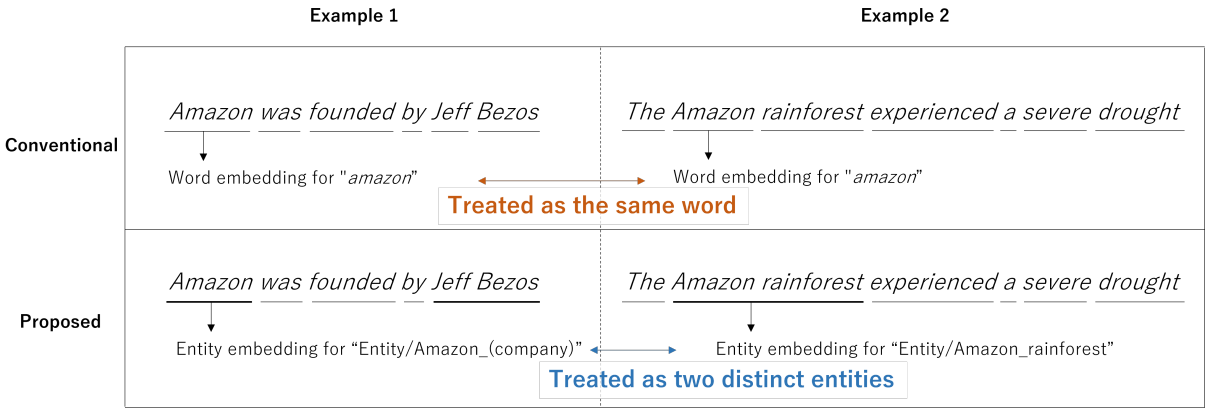


Figure 2: Difference between conventional embedded topic models and our proposed topic model.

can incorporate EL results into a neural topic model and make it aware of named entities. In the next section, we will evaluate the performance of our proposed method.

5 Experiments

In this section, we conduct two experiments. First, we evaluate our method on our original dataset, which requires a topic model to be aware of named entities. Our first experiment aims to verify that our method is effective in a case where word disambiguation is important. Next, we evaluate our method on the AIDA-CoNLL dataset (Hoffart et al., 2011). The AIDA-CoNLL dataset provides manual entity annotations. In this second experiment, we aim to assess 1) whether our method of incorporating entity information does not harm the performance of topic models even in a case where word disambiguation is not necessarily required and 2) how largely the off-the-shelve entity linker used in our pipeline deteriorates the performance in comparison with the use of the gold entity annotations.

5.1 Fine-Grained Topic Modeling

5.1.1 Experimental Setup

Dataset. In this experiment, we use archive news articles of *New York Times*[1]. We extract two subsets of articles published between the years 1996 and 2020: 1) a collection of 6,651 documents that include the word "*apple*" and 2) a collection of 3,070 documents that include "*amazon*". We regard each of the two collections as a single dataset and assess if our proposed method can train a more generalizable topic model by disambiguating homographic words, "*apple*" and "*amazon*". We randomly split each collection into 3:1:1 for training, validation, and test sets. Following Miyamoto et al. (2023), we filter out words that appear in 70% or more of documents and words included in a predefined stop-word list before building an embedding matrix $\boldsymbol{\rho}$. We group documents published within five consecutive years into a single time step. For example, news articles published between 1996 and 2000 are grouped.

Compared Models. We use **ETM** (Dieng et al., 2020) and **DSNTM** (Miyamoto et al., 2023) (one implementation of D-ETM (Dieng et al., 2019)) as baseline models, where only tokenization is applied to documents. ETM is not a dynamic topic model and does not consider time stamp information, whereas DSNTM is a dynamic topic model and can capture the chronological transition of topics. We assess if our method is effective in each model. We compare **ETM+EL** and **DSNTM+EL** (where we use entity embeddings for entities identified by an entity linker) with their corresponding baselines to see if our proposed method is effective.

Implementation Details. We set the number of topics $K = 10$ for all models. The variances of the prior distributions are set $\delta^2 = \sigma^2 = 0.005$ and $\gamma^2 = 1$. We use 500-dimensional word/entity embeddings (window size: 10)[2] pretrained with the Wikipedia2Vec toolkit (Yamada et al., 2020)[3]. Regarding other hyperparameters, we follow the official implementation of DSNTM[4]. For the preprocessing of documents, we utilize the tokenizer and entity linker implemented in the Stanford CoreNLP

[1] https://developer.nytimes.com

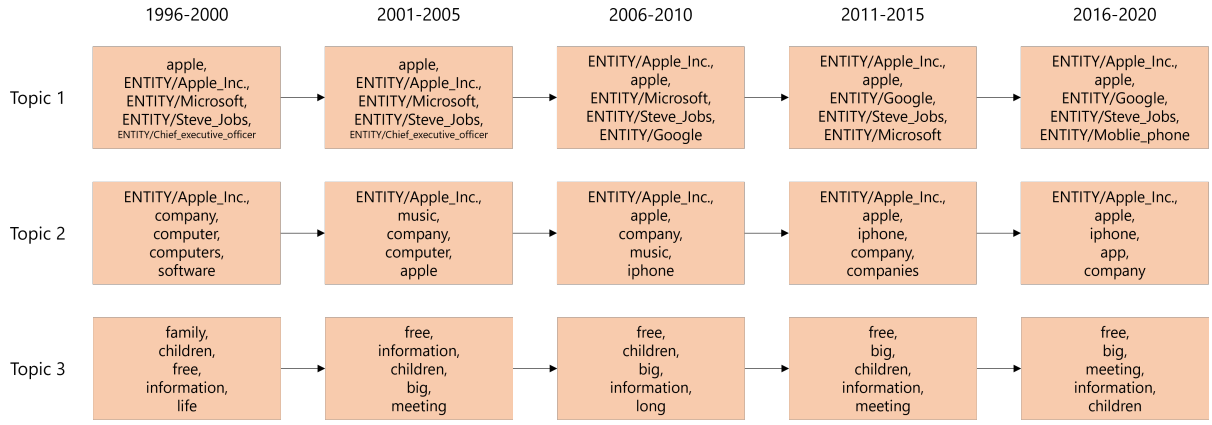
[2] http://wikipedia2vec.s3.amazonaws.com/models/en/2018-04-20/enwiki_20180420_win10_500d.txt.bz2

[3] https://wikipedia2vec.github.io/wikipedia2vec/

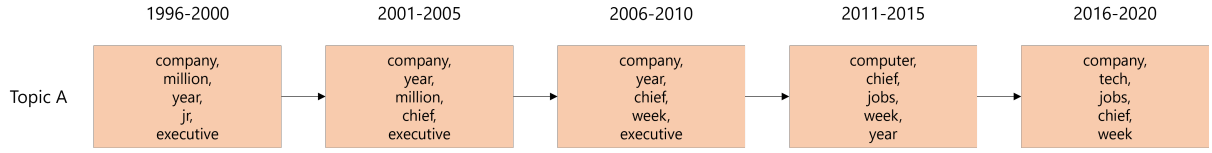
[4] https://github.com/miyamotononno/DSNTM

Method	"*apple*"	"*amazon*"
ETM	5753.3 ± 227.2	5086.7 ± 304.8
ETM+EL	5228.9 ± 730.9	6412.4 ± 731.3
DSNTM (Miyamoto et al., 2023)	4597.9 ± 270.0	4587.6 ± 349.0
DSNTM+EL	**3578.6** ± 141.4	4038.7 ± 65.9

Table 1: Results for perplexity with 95% confidence interval (CI) on our *New York Times* dataset. The lower, the better. EL means entity linking.



(a) DSNTM+EL



(b) DSNTM

Figure 3: Examples of topic transition. We present the top five most frequent terms in each topic.

toolkit (Manning et al., 2014).[5] We call these CoreNLP analyzers through the Stanza library (Qi et al., 2020)[6].

5.1.2 Quantitative Evaluation

We use perplexity (Rosen-Zvi et al., 2004) to evaluate the generalizability of a topic model. Although there is a discussion on how to properly evaluate topic models (Chang et al., 2009; Hoyle et al., 2021), perplexity is still a widely-used objective metric (Hida et al., 2018; Miyamoto et al., 2023). It measures the ability to predict words in unseen documents. In training, we apply early stopping based on the performance of a validation set. We train each model eight times with different random seeds and report the average performance and its 95% confidence interval on a test set.

[5]Although more accurate entity linkers (Shavarani and Sarkar, 2023; Wang et al., 2024) are publicly available, we choose the one implemented in the Stanford CoreNLP due to the limitation of computing resources.

[6]https://github.com/stanfordnlp/stanza

The results are shown in Table 1. We can find two tendencies in the results. The first one is that EL tends to improve the performance except for ETM on the "*amazon*" dataset. In particular, DSNTM+EL achieves lower perplexity than DSNTM. This demonstrates that word disambiguation by EL is effective in analyzing a collection of documents with a topic model. We will discuss the reason why our method does not work well with ETM on the "*amazon*" dataset in a later section. The second tendency is that DSNTM+EL performs better than ETM+EL. This means that modeling a temporal change of topics is effective even when EL is combined.

5.1.3 Qualitative Analysis

Visualization of Topic Transition. We present an overview of the topic transition process extracted by a trained DSNTM+EL model on the "*apple*" dataset in Figure 3(a). The topics in the first, second, and third rows (Topics 1, 2, and 3)

Word/Entity 1	Word/Entity 2	Cosine similarity
ENTITY/Apple_Inc.	*apple*	0.67
ENTITY/Apple_Inc.	ENTITY/Steve_Jobs	0.59
ENTITY/Apple_Inc.	*steve*	0.27
ENTITY/Apple_Inc.	*jobs*	0.28
apple	ENTITY/Steve_Jobs	0.52
apple	*steve*	0.30
apple	*jobs*	0.30

Table 2: Word similarities of two words/entities on Wikipedia2Vec (Yamada et al., 2020).

represent business/management, products/services, and *New York City*, respectively. When we look into Topic 1, the word "*apple*", the entity "ENTITY/Apple_Inc.", and the entity "ENTITY/Steve_Jobs" are frequently used constantly between 1996 and 2020, whereas the entity "ENTITY/Google" emerges after 2006. This is reasonable because Google was founded in 1998 and went public via an initial public offering (IPO) in 2004. Google was never mentioned before 1998 and not often before 2004. This demonstrates that DSNTM+EL successfully finds the transition of frequent terms in each topic and that we can easily understand the trends of topics by visualization. This is true for "*iphone*" (released in 2007) in Topic 2 as well. Regarding Topic 3, one might think this topic has nothing to do with the word "*apple*" at a glance, but this topic is related to *New York City*. *New York City* sometimes is called its nickname, "*Big Apple*". This topic consists of articles about *New York City*, especially entertainment such as *Big Apple Circus* and *Big Apple Chorus*. Then, the word "*big*" is listed as a frequent term.[7]

We also show the transition of a topic (Topic A) extracted by a trained DSNTM model in Figure 3(b). According to the frequent terms, Topic A is similar to Topic 1 in Figure 3(a). This means that a conventional topic model can analyze documents in a similar way. However, our method involving entity linking into its preprocessing comes with higher interpretability as frequent terms are expressed with not only words but also entities. The word "*jobs*" in Topic A means *Steve Jobs* in almost all cases, but DSNTM+EL shows that Topic 1 is related to *Steve Jobs* in a much easier-to-understand manner. This high interpretability is another advantage of our proposed method in addition to lower perplexities.

Influence of Entity Embedding. We investigate why entity linking (EL) boosts the performance of neural topic models. Some words have multiple meanings, whereas previous topic models deal with such words without being aware of meanings, considering only their spelling. In such an approach, a topic model can take into consideration neither who "*steve*" is nor whether "*jobs*" is a person's name or a common noun. In our proposed method, we try to disambiguate words, and use entity embedding trained with the Wikipedia2Vec toolkit (Yamada et al., 2020) instead of conventional word embedding if a word is linked to a KB entry.

Here, let us show some properties of the entity embedding used. We show the cosine similarities between some words/entities in Table 2. As shown, "ENTITY/Steve_Jobs" is much closer to "ENTITY/Apple_Inc." than the words "*steve*" and "*jobs*". This is because the word "*jobs*" can be a noun word (the plural form of "*job*"), and even "*steve*" can be the name of another person. Then, their embedding vectors are trained in various contexts. On the other hand, "ENTITY/Steve_Jobs" tends to appear in articles relevant to *Apple Inc.*, and then its entity embedding is trained in a narrow range of contexts. As a result, the entity embedding of "ENTITY/Steve_Jobs" has a large similarity to the entity embedding of "ENTITY/Apple_Inc.", while the word embeddings of "*steve*" and "*jobs*" go far from the entity embedding of "ENTITY/Apple_Inc.".

In ETM and DSNTM, a topic embedding $\boldsymbol{\alpha}_k^{(t)}$ is multiplied with a static word/entity embedding matrix $\boldsymbol{\rho}$ to estimate a distribution of terms, $w_{dn} \sim \mathrm{Cat}(\mathrm{softmax}(\boldsymbol{\rho}^\top \boldsymbol{\alpha}_{z_{dn}}^{(t_d)}))$ (See Section 3). This means that, if word/entity embedding vectors cluster based on their used context, topic embedding can be easily trained. Actually, entity embedding has such a property as we explained in the previous paragraph. Thus, entity embedding can help neural topic models extract topics from documents.

Dependency on Entity Linking. In contrast to our aim, entity linking (EL) does not boost the

[7] Ideally, entity linkers should recognize those entities correctly, but the entity linker used in our pipeline is not so accurate. As a result, the word "*big*" is listed.

Method	Tokenization & entity linking	Perplexity
ETM	Gold annotation	5380.5 ± 246.2
ETM+EL (ours)	Gold annotation	**5010.1** ± 448.8
ETM	Stanford CoreNLP	5404.9 ± 225.0
ETM+EL (ours)	Stanford CoreNLP	6558.1 ± 979.4

Table 3: Results for perplexity with 95% confidence interval (CI) on the AIDA-CoNLL dataset (Hoffart et al., 2011). The lower, the better. EL means entity linking.

performance of ETM on the "*amazon*" dataset, different from the "*apple*" dataset. We find that the accuracy of entity linking is not so good on the "*amazon*" dataset and that the entity linker fails to assign entity mentions to correct KB entries. Our proposed method is a pipeline of 1) preprocessing with an entity linker and 2) neural topic modeling. If the preprocessing is not accurate, the successive topic modeling will naturally be affected. We hypothesize that the latest, more accurate entity linkers (Shavarani and Sarkar, 2023; Wang et al., 2024) can boost the performance of neural topic models more. To verify our hypothesis, we will conduct an experiment on a dataset that contains manual entity annotations in the next section.

5.2 Coarse-Grained Topic Modeling

In this section, we evaluate our method on a dataset accompanied with gold entity annotations, to assess 1) whether our method of incorporating entity information does not harm the performance of topic models even in a case where word disambiguation is not necessarily required and 2) how largely the off-the-shelf entity linker used in our pipeline deteriorates the performance in comparison with the use of the gold entity annotations.

5.2.1 Experimental Setup

Dataset. In this experiment, we use the AIDA-CoNLL dataset (Hoffart et al., 2011)[8]. This dataset contains manual Wikipedia annotations for the 1,393 Reuters news stories originally published for the CoNLL-2003 Named Entity Recognition Shared Task (Tjong Kim Sang and De Meulder, 2003). The number of Wikipedia annotations is 27,817. The dataset consists of `train`, `testa`, and `testb` splits, which contain 946, 216, and 231 documents, respectively. We utilize the three splits as training, validation, and test sets. As in our previous experiment, we filter out words that appear in 70% or more of documents and words included in the predefined stop-word list before building an embedding matrix $\boldsymbol{\rho}$. In contrast to the *New York Times* dataset used in the previous experiment, which is created by collecting news articles that include a specific word such as "*apple*", the AIDA-CoNLL dataset was made without such an intention. It should include much less ambiguous words.

Compared Models. We use **ETM** (Dieng et al., 2020) as a baseline model. As the AIDA-CoNLL dataset provides gold annotations of entity linking (including tokenization), we can assess the influence of the off-the-shelf tokenizer and entity linker on the performance of our entire pipeline by comparing results from using gold annotations and results from using annotations by the tokenizer and entity linker. Therefore, we evaluate the following four models. 1) **ETM** that utilizes the gold annotations, 2) **ETM+EL** that uses the gold annotations, 3) **ETM** that utilizes annotations provided by Stanford CoreNLP, and 4) **ETM+EL** that uses annotations given by Stanford CoreNLP. Since the AIDA-CoNLL dataset does not include time stamp information, we do not adopt a dynamic topic model in this experiment.

Implementation Details. In this experiment, we use 300-dimensional word/entity embeddings (window size: 10)[9] because we encountered training instability with 500-dimensional word/entity embeddings. Regarding all other hyperparameters and implementations, we follow the previous experiment.

5.2.2 Results

The results are shown in Table 3. First, we can see that when the gold annotations are provided, entity linking improves the performance of ETM, even though the used AIDA-CoNLL dataset does not include as many homographic words as our *New York Times* dataset used in the previous experiment. This demonstrates that our method is potentially generalizable and can perform well on various data. Second, we observe that using information anno-

[8] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/ambiverse-nlu/aida

[9] http://wikipedia2vec.s3.amazonaws.com/models/en/2018-04-20/enwiki_20180420_win10_300d.txt.bz2

tated by the Stanford CoreNLP entity linker deteriorates the performance. As the knowledge base supported by the entity linker is not identical to that used for the annotations in the AIDA-CoNLL dataset, the accuracy of the entity linker can not be calculated so easily. However, we can attribute the performance gap between the two cases, 1) gold annotations and 2) the CoreNLP entity linker, to the accuracy of the entity linker. We believe that the latest, more accurate entity linkers (Shavarani and Sarkar, 2023; Wang et al., 2024) can boost the performance of neural topic models.

6 Conclusion

In this study, we proposed a method of analyzing a collection of documents after disambiguating homographic words. We incorporated entity information extracted by entity linking into neural topic models. Our experimental results demonstrated that entity linking improves the generalizability of topic models by disambiguating words such as "*apple*" and "*amazon*". In addition, our method offers higher interpretability as frequent terms in each topic are represented with not only words but also entities.

Limitations

Our models heavily rely on word/entity embedding as with other neural topic models. If the word/entity embedding contains some bias, our models will be affected by the bias.

Besides, topic models, including our models, sometimes infer incorrect information about topics, such as the frequent terms appearing in topics, the topic proportion in each document, and the dependencies among topics. There would be the potential risk of inducing misunderstandings among users.

Ethics Statement

Our study complies with the ACL Ethics Policy. We used *PyTorch* (BSD-style license), *New York Times* articles[10], the AIDA-CoNLL dataset (Creative Commons Attribution 3.0 license). Our study was conducted under their licenses and terms.

Acknowledgments

We thank anonymous reviewers for helpful feedback on our draft.

[10] https://developer.nytimes.com/terms

References

J. Atchison and S.M. Shen. 1980. Logistic-normal distributions:Some properties and uses. *Biometrika*, 67(2):261–272.

Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. 2016. Nonparametric spherical topic modeling with word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 537–542, Berlin, Germany. Association for Computational Linguistics.

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 113–120, New York, NY, USA. Association for Computing Machinery.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1):301–306.

Jordan Boyd-Graber and David M. Blei. 2009. Multilingual topic models for unaligned text. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 75–82, Arlington, Virginia, USA. AUAI Press.

Dallas Card, Chenhao Tan, and Noah A. Smith. 2018. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2031–2040, Melbourne, Australia. Association for Computational Linguistics.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Yulai Cong, Bo Chen, Hongwei Liu, and Mingyuan Zhou. 2017. Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 864–873. PMLR.

Kostadin Cvejoski, Ramsés J. Sánchez, and César Ojeda. 2023. Neural dynamic focused topic model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12719–12727.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on*

Natural Language Processing (Volume 1: Long Papers), pages 795–804, Beijing, China. Association for Computational Linguistics.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2019. The dynamic embedded topic model. *Preprint*, arXiv:1907.05545.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Shudong Hao and Michael J. Paul. 2018. Learning multilingual topics from incomparable corpora. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2595–2609, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Rem Hida, Naoya Takeishi, Takehisa Yairi, and Koichi Hori. 2018. Dynamic and static topic model for analyzing time-series document collections. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 516–520, Melbourne, Australia. Association for Computational Linguistics.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken? the incoherence of coherence. In *Advances in Neural Information Processing Systems*, volume 34, pages 2018–2033. Curran Associates, Inc.

Diederik Kingma and Max Welling. 2014. Efficient gradient-based inference through transformations between Bayes nets and neural nets. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1782–1790, Bejing, China. PMLR.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Nozomu Miyamoto, Masaru Isonuma, Sho Takase, Junichiro Mori, and Ichiro Sakata. 2023. Dynamic structured neural topic model with self-attention mechanism. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5916–5930, Toronto, Canada. Association for Computational Linguistics.

Volodymyr Miz, Joëlle Hanna, Nicolas Aspert, Benjamin Ricaud, and Pierre Vandergheynst. 2020. What is trending on Wikipedia? capturing trends and language biases across Wikipedia editions. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 794–801, New York, NY, USA. Association for Computing Machinery.

Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. 2009. Mining multilingual topics from Wikipedia. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 1155–1156, New York, NY, USA. Association for Computing Machinery.

James Petterson, Wray Buntine, Shravan Narayanamurthy, Tibério Caetano, and Alex Smola. 2010. Word features for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Tiziano Piccardi and Robert West. 2021. Crosslingual topic modeling with WikiPDA. In *Proceedings of the Web Conference 2021*, WWW '21, page 3032–3041, New York, NY, USA. Association for Computing Machinery.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, page 487–494, Arlington, Virginia, USA. AUAI Press.

Hassan Shavarani and Anoop Sarkar. 2023. SpEL: Structured prediction for entity linking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11123–11137, Singapore. Association for Computational Linguistics.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor

networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Junxiong Wang, Ali Mousavi, Omar Attia, Ronak Pradeep, Saloni Potdar, Alexander Rush, Umar Farooq Minhas, and Yunyao Li. 2024. Entity disambiguation via fusion entity decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6524–6536, Mexico City, Mexico. Association for Computational Linguistics.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar. Association for Computational Linguistics.

Pengtao Xie, Diyi Yang, and Eric Xing. 2015. Incorporating word correlation knowledge into topic modeling. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 725–734, Denver, Colorado. Association for Computational Linguistics.

Guangxu Xun, Vishrawas Gopalakrishnan, Fenglong Ma, Yaliang Li, Jing Gao, and Aidong Zhang. 2016. Topic discovery for short texts using word embeddings. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1299–1304.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A correlated topic model using word embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4207–4213.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Lisbon, Portugal. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30, Online. Association for Computational Linguistics.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany. Association for Computational Linguistics.

Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. 2018. WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In *International Conference on Learning Representations*.

Tao Zhang, Kang Liu, and Jun Zhao. 2013. Cross lingual entity linking with bilingual topic model. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 2218–2224. AAAI Press.

He Zhao, Lan Du, Wray Buntine, and Gang Liu. 2017. MetaLDA: A topic model that efficiently incorporates meta information. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 635–644.