Neural Attentive Bag-of-Entities Model for Text Classification

Ikuya Yamada[1,3]
ikuya@ousia.jp

Hiroyuki Shindo[2,3]
shindo@is.naist.jp

[1]Studio Ousia, Tokyo, Japan
[2]Nara Institute of Science and Technology, Nara, Japan
[3]RIKEN AIP, Tokyo, Japan

Abstract

This study proposes a *Neural Attentive Bag-of-Entities* model, which is a neural network model that performs text classification using entities in a knowledge base. Entities provide unambiguous and relevant semantic signals that are beneficial for capturing semantics in texts. We combine simple high-recall entity detection based on a dictionary, to detect entities in a document, with a novel neural attention mechanism that enables the model to focus on a small number of unambiguous and relevant entities. We tested the effectiveness of our model using two standard text classification datasets (i.e., the 20 Newsgroups and R8 datasets) and a popular factoid question answering dataset based on a trivia quiz game. As a result, our model achieved state-of-the-art results on all datasets. The source code of the proposed model is available online at https://github.com/wikipedia2vec/wikipedia2vec.

1 Introduction

Text classification is an important task, and its applications span a wide range of activities such as topic classification, spam detection, and sentiment classification. Recent studies showed that models based on neural networks can outperform conventional models (e.g., naïve Bayes) on text classification tasks (Kim, 2014; Iyyer et al., 2015; Tang et al., 2015; Dai and Le, 2015; Jin et al., 2016; Joulin et al., 2017; Shen et al., 2018). Typical neural network-based text classification models are based on words. They typically use words in the target documents as inputs, map words into continuous vectors (embeddings), and capture the semantics in documents by using compositional functions over word embeddings such as averaging or summation of word embeddings, convolutional neural networks (CNN), and recurrent neural networks (RNN).

Apart from the aforementioned approaches, past studies attempted to use entities in a knowledge base (KB) (e.g., Wikipedia) to capture the semantics in documents. These models typically represent a document by using a set of entities (or *bag of entities*) relevant to the document (Gabrilovich and Markovitch, 2006, 2007; Xiong et al., 2016). The main benefit of using entities instead of words is that unlike words, entities provide unambiguous semantic signals because they are uniquely identified in a KB. One key issue here is to determine the way in which to associate a document with its relevant entities. An existing straightforward approach (Peng et al., 2016; Xiong et al., 2016) involves creating a set of relevant entities using an entity linking system to detect and disambiguate the names of entities in a document. However, this approach is problematic because (1) entity linking systems produce disambiguation errors (Cornolti et al., 2013), and (2) entities appearing in a document are not necessarily relevant to the given document (Gamon et al., 2013; Dunietz and Gillick, 2014).

This study proposes the *Neural Attentive Bag-of-Entities* (NABoE) model, which is a neural network model that addresses the text classification problem by modeling the semantics in the target documents using entities in the KB. For each entity name in a document (e.g., *"Apple"*), our model first detects entities that may be referred to by this name (e.g., *Apple Inc.*, *Apple (food)*), and then represents the document using the weighted average of the embeddings of these entities. The weights are computed using a novel neural attention mechanism that enables the model to focus on a small subset of the entities that are less ambiguous in meaning and more relevant to the document. In other words, the attention mechanism is designed to compute weights by jointly addressing entity linking and entity salience detection (Gamon et al., 2013; Dunietz and Gillick, 2014) tasks. Furthermore, the attention mechanism improves the interpretability of the model because it enables us to inspect the small number of entities that strongly affect the classification decisions.

We validate the effectiveness of our proposed model by addressing two important natural language tasks: a text classification task using two standard datasets (i.e., the 20 Newsgroups and R8 datasets), and a factoid question answering task based on a popular dataset derived from the *quiz bowl* trivia quiz game. As a result, our model achieved state-of-the-art results on both tasks. The source code of the proposed model is available online at https://github.com/wikipedia2vec/wikipedia2vec.

2 Our Approach

Given a document, our model addresses the text classification task by using the following two steps: it first detects entities from the document, and then classifies the document using the proposed model with the detected entities as inputs.

2.1 Entity Detection

In this step, we detect entities that may be relevant to the document. Here, we use a simple method based on an *entity dictionary* that maps an entity name (e.g., “Washington”) to a set of possible referent entities (e.g., *Washington, D.C.* and *George Washington*). In particular, we first take all words and phrases in a document, treat them as entity names if they exist in the dictionary, and detect all possible referent entities for each detected entity name. Following past work (Hasibi et al., 2016; Xiong et al., 2016), the boundary overlaps of the names are resolved by detecting only those that are the earliest and the longest.

We use Wikipedia as the target KB, and the entity dictionary is built by using the names and their referent entities of all internal anchor links in Wikipedia (Guo et al., 2013). We also collect two statistics from Wikipedia, namely *link probability* and *commonness* (Mihalcea and Csomai, 2007; Milne and Witten, 2008). The former is the probability of a name being used as an anchor link in Wikipedia, whereas the latter is the probability of a name referring to an entity in Wikipedia.

We generate a list of entities by concatenating all possible referent entities contained in the dictionary for each detected entity name, and feed it to the model presented in the next section. Note that we do not disambiguate entity names here, but detect all possible referent entities of the entity names.

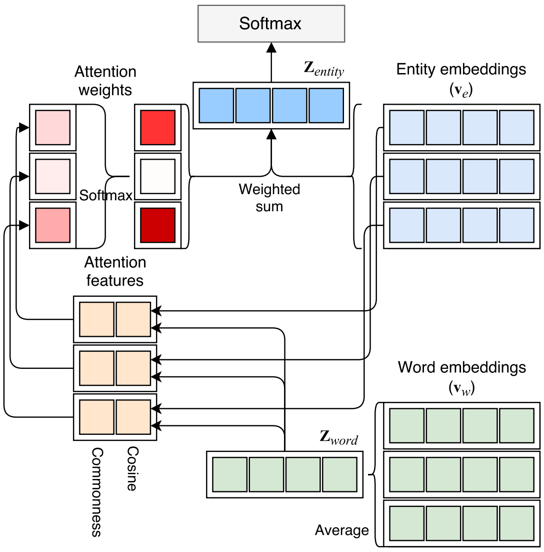

Figure 1: Architecture of the NABoE-entity model.

2.2 Model

Figure 1 shows the architecture of our model. Given words $w_1, ..., w_N$, and entities $e_1, ..., e_K$ detected from target document D, we first compute the word-based representation of D:

$$\mathbf{z}_{word} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_{w_i}, \tag{1}$$

where $\mathbf{v}_w \in \mathbb{R}^d$ is the embedding of word w. We then derive the entity-based representation of D as a weighted average of the embeddings of the entities:

$$\mathbf{z}_{entity} = \sum_{i=1}^{K} a_{e_i}\mathbf{v}_{e_i}, \tag{2}$$

where $\mathbf{v}_e \in \mathbb{R}^d$ is the embedding of entity e and a_e the normalized attention weight corresponding to e computed using the following softmax-based attention function:

$$a_e = \frac{\exp(\mathbf{w}_a^\top \Phi(e, D) + b_a)}{\sum_{i=1}^{K}\exp(\mathbf{w}_a^\top \Phi(e_i, D) + b_a)}, \tag{3}$$

where $\mathbf{w}_a \in \mathbb{R}^l$ is a weight vector, $b_a \in \mathbb{R}$ is the bias, and $\Phi(e, D)$ is a function that generates an l-dimensional vector consisting of the features of the attention function.

We use the following two features in the attention function:

- **Cosine**: the cosine similarity between the embedding of the entity $\mathbf{v}_e$ and the word-based representation of the document $\mathbf{z}_{word}$.
- **Commonness**: the probability that the entity name refers to the entity in KB.

Here, our aim is to capture the relevance and the unambiguity of entity e in document D using the attention function. Thus, the problem is related to the tasks of entity salience detection (Gamon et al., 2013; Dunietz and Gillick, 2014), which aims to detect entities relevant (or salient) to the document, and entity linking, which aims to resolve the ambiguity of entities. The key assumption relating to these two tasks in the literature is that if an entity is semantically related to the given document, it is relevant to the document (Dunietz and Gillick, 2014), and it is likely to appear in the document (Milne and Witten, 2008; Ratinov et al., 2011). With this in mind and following past work (Yamada et al., 2016), we use the cosine similarity between $\mathbf{v}_e$ and $\mathbf{z}_{word}$ as a feature. Further, as in past entity linking studies, we also use the commonness of the name referring to the entity.

Moreover, we derive a representation based both on entities and words by simply adding $\mathbf{z}_{entity}$ and $\mathbf{z}_{word}$[1]:

$$\mathbf{z}_{full} = \mathbf{z}_{entity} + \mathbf{z}_{word}. \tag{4}$$

We then solve the task using a multiclass logistic regression classifier with the computed representation (i.e., with $\mathbf{z}_{entity}$ or $\mathbf{z}_{full}$) as features. In the remainder of this paper, we denote our models based on $\mathbf{z}_{entity}$ and $\mathbf{z}_{full}$ by **NABoE-entity** and **NABoE-full**, respectively.

3 Experimental Setup

In this section, we describe our experimental setup used both in the text classification and the factoid question answering experiments presented below.

3.1 Entity Detection

As the target KB, we used the September 2018 version of Wikipedia, which contains a total of 7,333,679 entities.[2] Regarding the entity dictionary described in Section 2.1, we excluded an entity name if its link probability was lower than 1% and a referent entity if its commonness given the entity name was lower than 3% for computational efficiency. Entity names were treated as case-insensitive. As a result, the dictionary contained 18,785,550 entity names, and each name had 1.14 referent entities on average.

Furthermore, to detect entities from a document, we also tested two publicly available entity linking systems, **Wikifier** (Ratinov et al., 2011; Cheng and Roth, 2013) and **TAGME** (Ferragina and Scaiella, 2012), instead of using dictionary-based entity detection.[3] We selected these systems because they are capable of detecting non-named entities (e.g., technical terms) that are useful for addressing the text classification task.[4] Here, we used the entities detected and disambiguated by these systems as inputs to our neural network model.

3.2 Pretrained Embeddings

We initialized the embeddings of words ($\mathbf{v}_w$) and entities ($\mathbf{v}_e$) using pretrained embeddings trained on KB. To learn embeddings from the KB, we used the method adopted in the open source Wikipedia2Vec tool (Yamada et al., 2016, 2018a). In particular, we generated an entity-annotated corpus from Wikipedia by treating entity links in Wikipedia articles as entity annotations, and trained skip-gram embeddings (Mikolov et al., 2013a,b) of 300 dimensions with negative sampling using the generated corpus as inputs. The learned embeddings place similar words and entities close to one another in a unified vector space. Here, we used the same version of Wikipedia described in Section 3.1.

4 Text Classification

To evaluate the effectiveness of our proposed model, we first conducted the text classification

[1] We also tested concatenating $\mathbf{z}_{entity}$ and $\mathbf{z}_{word}$ to derive $\mathbf{z}_{full}$; however, adding them generally achieved enhanced performance in our experiments presented below.

[2] We downloaded the Wikipedia dump from Wikimedia Downloads: `https://dumps.wikimedia.org/`

[3] In our experiments, we simply used all entities detected by the entity linking systems.

[4] In our preliminary experiments, we also tested three other state-of-the-art entity linking systems: AIDA (Hoffart et al., 2011), WAT (Piccinno and Ferragina, 2014), and the commercial Entity Analysis API in Google's Cloud Language service. However, these systems achieved lower overall performance compared to Wikifier and TAGME because they tended to ignore non-named entities.

task on two standard datasets, namely the 20 Newsgroups (20NG) (Lang, 1995) and R8 datasets (Debole and Sebastiani, 2005).

4.1 Setup

Our experimental setup described in this section follows that in past work (Liu et al., 2015; Jin et al., 2016; Yamada et al., 2018b). In particular, we used the 20NG and R8 datasets to train and test the proposed model. The 20NG dataset was created using the documents obtained from 20 Newsgroups and contained 11,314 training documents and 7,532 test documents.[5] The R8 dataset consisted of news documents from the eight most popular classes of the Reuters-21578 corpus (Lewis, 1992) and comprised 5,485 training documents and 2,189 test documents. We created the development set for each dataset by selecting 5% of the documents for training. Note that the class distribution of the R8 dataset is highly imbalanced. For example, the number of documents in the largest and smallest classes is 3,923 documents and 51 documents, respectively.

We report the accuracy and macro-average F1 scores. The model was trained using mini-batch stochastic gradient descent (SGD) with its batch size set to 32 and its learning rate controlled by Adam (Kingma and Ba, 2014). We used words and entities that were detected three times or more in the dataset and ignored the other words and entities. The size of the embeddings of words and entities was set to $d = 300$. We used early stopping based on the accuracy of the development set of each dataset to avoid overfitting of the model.

4.2 Baselines

We used the following models as our baselines:

- **BoW-SVM** (Jin et al., 2016): This model is based on a conventional linear support vector machine (SVM) with bag of words (BoW) features. It outperformed the conventional naïve Bayes-based model.

- **BoE** (Jin et al., 2016): This model extends the skip-gram model; It learns different word embeddings per target class from the dataset, and a linear model based on learned word embeddings is used to classify the documents. The performance of this model was superior to that of many state-of-the-art models, including those based on the skip-gram and CBOW models (Mikolov et al., 2013b), and the paragraph vector model (Le and Mikolov, 2014).

- **SWEM-concat** (Shen et al., 2018): This model is based on a neural network model with simple pooling operations (i.e., average and max pooling) over pretrained word embeddings.[6] Despite its simplicity, it outperformed many neural network-based models such as the word-based CNN model (Kim, 2014) and RNN model with LSTM units (Shen et al., 2018).

- **TextEnt** (Yamada et al., 2018b): This model learns entity-aware document embeddings from Wikipedia, and uses a neural network model with the learned embeddings as pretrained parameters to address text classification.

As described in Section 2.1, we also tested the variants of our NABoE-entity and NABoE-full models for which **Wikifier** and **TAGME** were used as the entity detection methods.

4.3 Results

Table 1 shows the results of our models and those of our baselines. Here, *w/o att.* and *w/o emb.* signify the model without the neural attention mechanism (all attention weights a_e are set to $\frac{1}{K}$, where K is the number of entities in the document) and the model without the pretrained embeddings (the embeddings are initialized randomly), respectively.

Relative to the baselines, our models yielded enhanced overall performance on both datasets. The NABoE-full model outperformed all baseline models in terms of both measures on both datasets. Furthermore, the NABoE-entity model outperformed all the baseline models in terms of both measures on the 20NG dataset, and the F1 score on the R8 dataset. Moreover, our attention mechanism consistently improved the performance. These results clearly highlighted the effectiveness of our approach, which addresses text

[5] We used the by-date version downloaded from the author's web site: http://qwone.com/~jason/20Newsgroups/.

[6] We also tested all four models proposed in Shen et al. (2018) (i.e., SWEM-aver, SWEM-max, SWEM-concat, and SWEM-hier). These models generally delivered comparable performance, with SWEM-concat slightly outperforming the other models on average.

	20NG		R8	
	Acc.	F1	Acc.	F1
NABoE-entity	.863	.856	.962	.915
NABoE-entity w/o att.	.822	.817	.943	.869
NABoE-entity w/o emb.	.844	.838	.957	.892
NABoE-full	**.868**	**.862**	**.971**	**.917**
Wikifier (NABoE-entity)	.735	.729	.896	.803
Wikifier (NABoE-entity w/o att.)	.728	.723	.844	.782
Wikifier (NABoE-entity w/o emb.)	.727	.722	.861	.755
Wikifier (NABoE-full)	.797	.789	.953	.839
TAGME (NABoE-entity)	.844	.838	.942	.871
TAGME (NABoE-entity w/o att.)	.826	.821	.924	.857
TAGME (NABoE-entity w/o emb.)	.842	.836	.942	.865
TAGME (NABoE-full)	.860	.853	.958	.889
BoW-SVM	.790	.783	.947	.851
BoE	.831	.827	.965	.886
SWEM-concat	.853	.855	.967	.898
TextEnt	.845	.839	.967	.910

Table 1: Results of the text classification task on the 20NG and R8 datasets. Here, *w/o att.* and *w/o emb.* represent the model without the neural attention mechanism and the model without the pretrained embeddings, respectively.

classification by using a small number of unambiguous and relevant entities detected by the proposed attention mechanism. Moreover, the pretrained embeddings improved the performance on both datasets.

Further, the models based on the dictionary-based entity detection (see Section 2.1) generally outperformed the models based on the entity linking systems (i.e., Wikifier and TAGME). We consider that this is because these entity linking systems failed to detect or disambiguate entity names that were useful to address the text classification task. Moreover, our attention mechanism consistently improved the performance for Wikifier- and TAGME-based models because the attention mechanism enabled the model to focus on entities that were relevant to the document.

Class	SWEM -concat	NABoE -full	NABoE -entity
20NG:			
alt.atheism	.780	**.820**	.804
comp.graphics	.787	.818	**.822**
comp.os.ms-windows.misc	.746	.802	**.811**
comp.sys.ibm.pc.hardware	.735	**.754**	.752
comp.sys.mac.hardware	.857	**.865**	.861
comp.windows.x	.837	.867	**.870**
misc.forsale	**.854**	.834	.805
rec.autos	.916	**.929**	.917
rec.motorcycles	.954	**.968**	.956
rec.sport.baseball	.946	**.969**	.966
rec.sport.hockey	.971	**.981**	.975
sci.crypt	**.942**	.940	.940
sci.electronics	.794	**.806**	.783
sci.med	.878	.900	**.905**
sci.space	.921	**.923**	.918
soc.religion.christian	.905	**.906**	.905
talk.politics.guns	.826	**.828**	.819
talk.politics.mideast	.921	**.940**	.935
talk.politics.misc	.689	**.694**	.680
talk.religion.misc	.657	.702	**.706**
R8:			
grain	.750	**.889**	**.889**
ship	.781	.817	**.822**
interest	.910	.885	.885
money-fx	**.909**	.894	.898
trade	.894	**.924**	**.924**
crude	**.971**	.958	.954
acq	.979	**.980**	.966
earn	.989	**.990**	.980

Table 2: Class-level F1 scores in each class on the 20NG and R8 datasets.

	20NG		R8	
	Acc.	F1	Acc.	F1
Commonness only	.849	.843	.949	.894
Cosine only	.846	.840	.956	.898
Both	**.863**	**.856**	**.962**	**.915**

Table 3: Feature study of the neural attention mechanism of the NABoE-entity model.

4.4 Analysis

In this section, we provide a detailed analysis of the performance of our model in terms of conducting the text classification task. We first provide a comparison of the SWEM-concat, NABoE-entity, and NABoE-full models using class-level F1 scores on both of the datasets (see Table 2). Here, we aim to compare the detailed performance of the word-based model (SWEM-concat), entity-based model (NABoE-entity), and the model based on both words and entities (NABoE-full). Compared with the SWEM-concat model, the NABoE-full and NABoE-entity models performed more accurately in 23 out of 28 and 17 out of 28 classes, respectively. This result clearly demonstrates the ability of the model to successfully capture strong semantic signals that can only be obtained from entities. Moreover, we observed that the NABoE-entity model achieved weaker performance especially for the *misc.forsale* class in the 20NG dataset and several classes in the R8 dataset. Regarding the *misc.forsale* class, because documents in this class contain a wider variety of entities (i.e., objects users want to sell) than other classes, the model failed to capture the effective semantic signals from the entities. Further, as described in the error analysis provided below, it often appeared to be difficult to distinguish pairs of

Class	Top entities
20NG:	
alt.atheism	Christian ethics, Atheism, Moral agency, Gregg Jaeger, Fred Rice
comp.graphics	Algorithm, Ray tracing (graphics), Framebuffer, Image file formats, TIFF
comp.os.ms-windows.misc	Windows 3.1x, Microsoft Windows, Windows NT, CONFIG.SYS, BMP file format
comp.sys.ibm.pc.hardware	BIOS, Don't Copy That Floppy, SCSI host adapter, Nonvolatile BIOS memory, Parallel SCSI
comp.sys.mac.hardware	PowerBook, Macintosh Quadra 610, Macintosh Quadra 650, FirstClass, Macintosh SE/30
comp.windows.x	X-Perts, Xterm, OPEN LOOK, OpenWindows, Man page
misc.forsale	Freight transport, Make Me an Offer, AC adapter, Plaque reduction neutralization test, Outline of working time and conditions
rec.autos	Manual Shift, Chassis, Automotive industry, Nissan, Ford Probe
rec.motorcycles	United States Department of Defense, Motorcycle, ZX8302, Honda motorcycles, Pillion, Hawk GT
rec.sport.baseball	Pitcher, Inning, The Jays, Home run, Bullpen
rec.sport.hockey	National Hockey League, Goaltender, ESPN, The Penguins, Achkar
sci.crypt	Cryptography, Algorithm, Escrow, Considered harmful, Encryption
sci.electronics	Solvent, Copy protection, Electronics, Leadacid battery, Printed circuit board
sci.med	Infection, Antibiotics, Kirlian photography, Allergy, Kirlian
sci.space	Spacecraft, SunOS, Vandalism, VIA International, Space station
soc.religion.christian	Rutgers University, Geneva, Byler, Immaculate Conception, Original sin
talk.politics.guns	Ranch, BD's Mongolian Grill, Firearm, Second Amendment to the United States Constitution, Feustel
talk.politics.mideast	Serdar Argic, Israelis, Palestinians, Palestine Liberation Organization, Arabs
talk.politics.misc	Clayton Cramer, Janet Reno, Police state, Ronzone, Federal Bureau of Investigation
talk.religion.misc	Christian ethics, Thomas George Lanphier, David Koresh, Albert Sabin, Josephus
R8:	
grain	Grain, Tonne, Price support, Oil reserves, United States Senate
ship	Freight transport, Shipbuilding, Flag of convenience, Cargo, Persian Gulf
trade	Balance of trade, Export, International trade, Economic sanctions, Import
interest	Interest rate, Prime rate, Repurchase agreement, Balance of trade, Money market
money-fx	Exchange rate, Currency, Money market, Foreign exchange market, Monetary policy
crude	Petroleum, West Texas Intermediate, Price of oil, OPEC, Oil platform
acq	Common stock, Tender offer, Privately held company, Preferred stock, Shares outstanding
earn	QTR, Dividend, Stock split, Net profit, Income fund

Table 4: Top five influential entities for each class of the NABoE-entity model in the 20NG and R8 datasets.

similar classes in the R8 dataset based only on entities.

Next, we conducted a feature study of the attention mechanism by excluding one feature at a time from the NABoE-entity model (Table 3). We found both of the features to make an important contribution to the performance.

Furthermore, to investigate the attention mechanism in more detail, we computed the top influential entities in the attention mechanism for each class on the 20NG and R8 datasets. In particular, we calculated the number of times each entity obtained the highest attention weight in the test documents in each class and selected the five most frequent ones. Table 4 presents the results. Overall, our attention mechanism successfully selected entities that were highly relevant to each class. For example, *Cryptography*, *Algorithm*, *Escrow*, *Considered harmful*, and *Encryption* were selected for the *sci.crypt* class. Furthermore, although we did not explicitly perform entity disambiguation, the model successfully overcame the ambiguity issues in the entity names and attended to the entities that were relevant to the classes.

Subsequently, we conducted an error analysis by selecting 50 random test documents for which the NABoE-entity model made wrong predictions. Most of the errors were caused by two pairs of classes: 22 errors were caused by misclassifying documents of *acq* (corporate acquisitions) and those of *earn* (corporate earnings), and 13 errors were caused by misclassifying documents of *interest* and those of *money-fx*. Furthermore, the model tended to perform poorly if a document contained entities that strongly indicate an incorrect class. For example, a *money-fx* document containing the entity *interest rate* multiple times was classified into the *interest* class, and a document in the *acq* class reporting news related to oil companies (i.e., ExxonMobil and ZENEX) was classified into the *crude* class.

5 Factoid Question Answering

In this section, we address factoid question answering based on a dataset consisting of questions of the *quiz bowl* trivia quiz game. Factoid question answering is one of the common settings of question answering that aims to predict an entity (e.g., events, authors, and books) that is described in a given question. The players of quiz bowl solve questions consisting of sentences that describe an entity. Quiz bowl questions have frequently been used for evaluating neural network-based models in recent studies (Iyyer et al., 2014, 2015; Yamada et al., 2017).

This task has a significantly larger number of target classes compared to the task addressed in the previous experiment. Our main aim here is to evaluate the effectiveness of using entities to capture the finer-grained semantics required to perform the task of factoid question answering effectively.

5.1 Setup

Our experimental setup described in this section follows that in past work (Xu and Li, 2016; Yamada et al., 2017). We address this task as a text classification problem that selects the most relevant answer from the possible answers observed in the dataset. We obtained the dataset proposed in Iyyer et al. (2014)[7]. We only used questions in the history and literature categories. Furthermore, we excluded questions of which the answers appear fewer than six times in the dataset. As a result, the number of candidate answers was 303 and 424 in the history and literature categories, respectively. We used 20% of questions each for the development set and test sets, and the remaining 60% for the training set. As a result, the training, development, and test sets consisted of 1,535, 511, and 511 questions for the history category, and 2,524, 840, and 840 questions for the literature category.

The settings we used to train the model were the same as those in the previous experiment (see Section 4.1). The model was trained using mini-batch SGD with its learning rate controlled by Adam (Kingma and Ba, 2014) and its mini-batch size set to 32. We used words and entities that were detected three times or more in the dataset, and ignored the other words and entities. The size of the embeddings of words and entities was set to $d = 300$. As in past work, we report the accuracy score, and the score on the development set was used for early stopping.

Name	History	Literature
NABoE-full	**.949**	**.985**
NABoE-entity	.941	.979
NABoE-entity w/o att.	.845	.943
NABoE-entity w/o emb.	.941	.973
Wikifier (NABoE-full)	.935	.967
Wikifier (NABoE-entity)	.930	.952
Wikifier (NABoE-entity w/o att.)	.924	.941
Wikifier (NABoE-entity w/o emb.)	.934	.949
TAGME (NABoE-full)	.941	.977
TAGME (NABoE-entity)	.930	.963
TAGME (NABoE-entity w/o att.)	.922	.961
TAGME (NABoE-entity w/o emb.)	.932	.962
BoW	.508	.462
FTS-BRNN	.881	.931
NTEE	.947	.951
SWEM-concat	.900	.966

Table 5: Accuracy of the proposed and baseline methods for the factoid QA task.

5.2 Baselines

We used the following baseline models:

- **BoW** (Xu and Li, 2016) This model is based on a logistic regression classifier with conventional binary BoW features.
- **FTS-BRNN** (Xu and Li, 2016) This model is based on a bidirectional RNN with gated recurrent units (GRU). It uses the logistic regression classifier with the features derived by the RNN.
- **NTEE** (Yamada et al., 2017) This model is a state-of-the-art model that uses a multi-layer perceptron classifier with the features computed using the embeddings of words and entities trained on Wikipedia using the neural network model proposed in their paper.

Similar to our previous experiment, we also add **SWEM-concat**, and the variants of our NABoE-entity and NABoE-full models based on **Wikifier** and **TAGME** (see Section 4.2). Note that all the baselines address the task as a text classification problem.

5.3 Results and Analysis

Table 5 provides the results of our models and those of our baselines. Overall, our models achieved enhanced performance on this task. In particular, the NABoE-full model successfully outperformed all the baseline models, and the NABoE-entity model achieved competitive performance and outperformed all the baseline models in the literature category. These results clearly

[7] This dataset was downloaded from the authors' web page: https://cs.umd.edu/œmiyyer/qblearn/.

highlighted the effectiveness of our model for this task.

Furthermore, similar to the previous text classification experiment, the attention mechanism and the pretrained embeddings consistently improved the performance. Moreover, the models based on dictionary-based entity detection outperformed the models based on the entity linking systems.

We also conducted an error analysis using the NABoE-entity model and the test questions in the history category. We found nearly 70% of the errors to be caused by questions of which the answers were country names. This is because these questions tended to provide indirect clues (e.g., describing a notable person born in the country) and most entities used in these clues do not directly indicate the answer (i.e., country names). Furthermore, our model failed in difficult cases such as predicting *Tokugawa shogunate* instead of *Tokugawa Ieyasu*.

6 Related Work

KB entities have been conventionally used to model the semantics in texts. A representative example is Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2006, 2007), which represents a document using a bag of entities, namely a sparse vector of which each dimension corresponds to the relevance score of the text to each entity. This simple method is shown to be effective for various NLP tasks including text classification (Gabrilovich and Markovitch, 2006; Gupta and Ratinov, 2008; Negi and Rosner, 2013) and information retrieval (Egozi et al., 2011; Xiong et al., 2016),

Several neural network models that use KB entities to capture the semantics in texts have been proposed. These models typically depend on an additional preprocessing step that extracts the relevant entities from the target texts. For example, Wang et al. (2017) used the Probase conceptualization API for short text classification by retrieving the Probase entities that were relevant to the target text and used them in a model based on CNN. Pilehvar et al. (2017) also extracted entities using a graph-based linking algorithm and used these entities in a neural network model. A similar approach was adopted in Yamada et al. (2018b,c); they extracted entities from the target text using an entity linking system and simply used the detected entities in a neural network model. However, unlike these models, our proposed model addresses the task in an *end-to-end* manner; i.e., entities that are relevant to the target text are automatically selected using our neural attention mechanism. Furthermore, we also used the model proposed by Yamada et al. (2018b) as a baseline in our text classification experiments.

Additionally, our work is also related to studies on entity linking. Entity linking models can be roughly classified into two groups: *local* models, which resolve entity names independently using the contextual relevance of the entity given a document, and *global* models, in which all the entity names in a document are resolved simultaneously to select a topically coherent set of results (Ratinov et al., 2011). Recent state-of-the-art models typically combine both of these models (Yamada et al., 2016; Ganea and Hofmann, 2017; Cao et al., 2018; Kolitsas et al., 2018). However, several studies also showed that the local model alone can achieve results competitive to those of the global and combined models (Eshel et al., 2017; Ganea and Hofmann, 2017; Yamada et al., 2017; Cao et al., 2018; Kolitsas et al., 2018). In this study, we adopt a simple but effective local model, which uses cosine similarity between the embedding of the target entity and the word-based representation of the document to capture the relevance of an entity given a document.

7 Conclusions

This study proposed NABoE, which is a neural network model that performs text classification using entities in Wikipedia. We combined simple dictionary-based entity detection with a neural attention mechanism to enable the model to focus on a small number of unambiguous and relevant entities in a document. We achieved state-of-the-art results on two important NLP tasks, namely text classification and factoid question answering, which clearly verified the effectiveness of our approach. As a future task, we intend to more extensively analyze our model and explore its effectiveness for other NLP tasks. Furthermore, we would also like to test more expressive neural network models for example by integrating global entity coherence information into our neural attention mechanism.

References

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural Collective Entity Linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 675–686.

Xiao Cheng and Dan Roth. 2013. Relational Inference for Wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796.

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. 2013. A Framework for Benchmarking Entity-annotation Systems. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 249–260.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised Sequence Learning. In *Advances in Neural Information Processing Systems 28*, pages 3079–3087.

Franca Debole and Fabrizio Sebastiani. 2005. An Analysis of the Relative Hardness of Reuters-21578 Subsets: Research Articles. *Journal of the American Society for Information Science and Technology*, 56(6):584–596.

Jesse Dunietz and Daniel Gillick. 2014. A New Entity Salience Task with Millions of Training Examples. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 205–209.

Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. 2011. Concept-Based Information Retrieval Using Explicit Semantic Analysis. *ACM Trans. Inf. Syst.*, 29(2):8:1—-8:34.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. 2017. Named Entity Disambiguation for Noisy Text. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 58–68.

Paolo Ferragina and Ugo Scaiella. 2012. Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *Software, IEEE*, 29(1):70–75.

Evgeniy Gabrilovich and Shaul Markovitch. 2006. Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume 2, pages 1301–1306.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness Using Wikipedia-Based Explicit Semantic Analysis. In *International Joint Conference on Artificial Intelligence*, pages 1606–1611.

Michael Gamon, Tae Yano, Xinying Song, Johnson Apacible, and Patrick Pantel. 2013. Identifying Salient Entities in Web Pages. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 2375–2380.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030.

Rakesh Gupta and Lev Ratinov. 2008. Text Categorization with Knowledge Transfer from Heterogeneous Data Sources. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, pages 842–847.

Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2016. Exploiting Entity Linking in Queries for Entity Retrieval. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 209–218.

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 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A Neural Network for Factoid Question Answering over Paragraphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 633–644.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. 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 1681–1691.

Peng Jin, Yue Zhang, Xingyuan Chen, and Yunqing Xia. 2016. Bag-of-embeddings for Text Classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2824–2830.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-End Neural Entity Linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529.

Ken Lang. 1995. NewsWeeder: Learning to Filter Netnews. *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339.

Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning*, volume 32, pages 1188–1196.

David D. Lewis. 1992. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2418–2424.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 233–242.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 2013 International Conference on Learning Representations*, pages 1–12.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

David Milne and Ian H. Witten. 2008. Learning to Link with Wikipedia. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pages 509–518.

Sapna Negi and Michael Rosner. 2013. UoM: Using Explicit Semantic Analysis for Classifying Sentiments. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, pages 535–538.

Hao Peng, Jing Liu, and Chin-Yew Lin. 2016. News Citation Recommendation with Implicit and Explicit Semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–398.

Francesco Piccinno and Paolo Ferragina. 2014. From TagME to WAT: A New Entity Annotator. In *Proceedings of the First International Workshop on Entity Recognition and Disambiguation*, pages 55–62.

Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a Seamless Integration of Word Senses into Downstream NLP Applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1857–1869.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2915–2921.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2016. Bag-of-Entities Representation for Ranking. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 181–184.

Dong Xu and Wu-Jun Li. 2016. Full-Time Supervision based Bidirectional RNN for Factoid Question Answering. *arXiv preprint arXiv:1606.05854v2*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2018a. Wikipedia2Vec: An Optimized Tool for Learning Embeddings from Wikipedia. *arXiv preprint arXiv:1812.06280v2*.

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.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning Distributed Representations of Texts and Entities from Knowledge Base. *Transactions of the Association for Computational Linguistics*, 5:397–411.

Ikuya Yamada, Hiroyuki Shindo, and Yoshiyasu Takefuji. 2018b. Representation Learning of Entities and Documents from Knowledge Base Descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 190–201.

Ikuya Yamada, Ryuji Tamaki, Hiroyuki Shindo, and Yoshiyasu Takefuji. 2018c. Studio Ousia's Quiz Bowl Question Answering System. In *The NIPS '17 Competition: Building Intelligent Systems*, pages 181–194.