Leveraging linguistic resources for improving neural text classification

Ming Liu
FIT, Monash University
ming.m.liu@monash.edu

Gholamreza Haffari
FIT, Monash University
gholamreza.haffari@monash.edu

Wray Buntine
FIT, Monash University
wray.buntine@monash.edu

Michelle R. Ananda-Rajah
Alfred Health and Monash University
michelle.ananda-rajah@monash.edu

Abstract

This paper presents a deep linguistic attentional framework which incorporates word level concept information into neural classification models. While learning neural classification models often requires a large amount of labelled data, linguistic concept information can be obtained from external knowledge, such as pre-trained word embeddings, WordNet for common text and MetaMap for biomedical text. We explore two different ways of incorporating word level concept annotations, and show that leveraging concept annotations can boost the model performance and reduce the need for large amounts of labelled data. Experiments on various data sets validate the effectiveness of the proposed method.

1 Introduction

Text classification is an important task in natural language processing, such as sentiment analysis, information retrieval, web page ranking and document classification (Pang et al., 2008). Recently, deep neural models have been widely used in this area due to their abstract framework and good performance. While these models are being used frequently, they require a large amount of labelled data and training time.

The core idea of text neural classification models is that text signals are fed into composition and activation functions via deep neural networks, and then a softmax classifier generates the final label as a probability distribution. Unlike standard n-gram models, word representation (Mikolov et al., 2013) is distributed and manual features are not usually necessary in deep neural models.

Though promising, most current text neural classification models still lack the ability of modeling linguistic information of the language, especially in domains where annotations are time-consuming and expensive such as biomedical text. In this work, we use some prior knowledge from pre-trained word embeddings or knowledge bases, and explore different ways of incorporating this prior knowledge into existing deep neural classification models. Our model is an integration of a simple neural bag of words model, which works in 2 steps:

1. create mappings from a sequence of word tokens into concept tokens (based on the given pre-trained word embeddings or knowledge bases),

2. combine the embeddings of both word and concept tokens and pass the resulting embedding through a deep feed-forward classification model to make the final prediction.

The motivation of our work is to incorporate extra knowledge from pre-trained word embeddings or knowledge bases such as WordNet for common text, MetaMap for biomedical text. Our main contributions are: (1) creating linguistically-related concepts of words from external knowledge bases; (2) incorporating the concept information either through what we call direct or gated mappings. We show that leveraging concept annotations can boost the model performance and reduce the need for large amounts of labelled data, and the concept information can be incorporated more effectively in a gated mapping manner.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the architecture of incorporating concept information. Data sets and implementation details are described in section 4. Section 5 demonstrates the effectiveness of our method with experiments. Finally, section 6 offers concluding remarks.

Ming Liu, Gholamreza Haffari, Wray Buntine and Michelle Ananda-Rajah. 2017. Leveraging linguistic resources for improving neural text classification. In *Proceedings of Australasian Language Technology Association Workshop*, pages 34–42.

2 Related Work

This section describes some related work on deep neural models for text classification and several common knowledge bases.

2.1 Text classification with deep neural models

Composition functions play a key role in many deep neural models. Generally, composition functions fall into two categories: unordered and syntactic. Unordered functions regard input text as bags of word embeddings (Iyyer et al., 2015), while syntactic models take word order and sentence structure into account (Mikolov et al., 2010; Socher et al., 2013b). Previously published results have shown that syntactic models have outperformed unordered ones on many tasks. RecNN-based approaches (Socher et al., 2011, 2013a,b) rely on parsing trees to construct the semantic function, in which each leaf node in the tree corresponds to a word. Recursive neural models then compute parent vectors in a bottom up fashion using different types of compositionality functions. While parsing is the first step, RecNNs are restricted to modelling short text like sentences rather than documents. Recurrent neural networks (RNNs) (Mikolov et al., 2010) are another natural choice to model text due to their capability of processing arbitrary-length sequences. Unfortunately, a problem with RNNs is that the transition function inside can cause the gradient vector to grow or decay exponentially over long sequences. The LSTM architecture (Hochreiter and Schmidhuber, 1997) addresses this problem by introducing a memory cell that is able to preserve state over a long period of time. Tree-LSTM (Tai et al., 2015) is an extension of standard LSTM in that Tree-LSTM computes its hidden state from the current input and the hidden states of arbitrarily many child units. Convolutional networks (Kalchbrenner et al., 2014) also model word order in local windows and have achieved performance comparable or better than that of RecNNs or RNNs on many tasks.

While models that use syntactic functions need large training time and data, unordered functions allow a tradeoff between training time and model complexity. Unlike some of the previous syntactic approaches, paragraph vector (Le and Mikolov, 2014) is capable of constructing representations of input sequences of variable length. It does not require task-specific tuning of the word weighting function nor does it rely on the parse trees. A compatible unordered method is also used in DANs (Iyyer et al., 2015), which averages the embeddings for all of a document's tokens and feeds that average through multiple layers. They show nonlinearly transforming the input is more important than tailoring a network to incorporate word order and syntax.

2.2 Exploiting linguistic resources

Besides distributed word representation, there exist many large-scale knowledge bases (KBs) in general or specific domains that can be used as prior information for text classification models. WordNet (Miller, 1995) is the most widely used lexical reference system which organizes nouns, verbs, adjectives and adverbs into synonym sets (synsets). Synsets are interlinked by a number of conceptual-semantic and lexical relations such as hypernym, synonym and meronym, etc. WordNet has already been used in reducing vector dimensionality for many text clustering tasks and showed that the lexical categories within it is quite useful. It includes a core ontology and a lexicon. The latest version is WordNet 3.0 which consists of 155,287 lexical entries and 117,659 synsets.

In the medical domain, some domain knowledge that may be useful to classifiers is also available in the form of existing knowledge sources (Baud et al., 1996). The UMLS (Bodenreider, 2004) knowledge sources provide huge amounts of linguistic information readily available to the medical community. SNOMED (Spackman et al., 1997) is today the largest source of medical vocabulary (132,643 entries) organised in a systematic way. The GALEN (Rector, 1995) consortium is working together since 1992 and has produced, using the GRAIL representation language, a general model of medicine with nearly 6,000 concepts. The MED (Medical Entities Dictionary) (Cimino, 2000) is a large repository of medical concepts that are drawn from a variety of sources either developed or used at the New York Presbyterian Hospital, including the UMLS, ICD9-CM and LOINC. Currently numbering over 100,000, these concepts correspond to coded terms used in systems and applications throughout both medical centers (Columbia-Presbyterian and New York-Cornell). MetaMap (Aronson, 2001) was developed to map biomedical free text to biomedical knowledge representation in which concepts were classified by semantic type and both hierarchical and non-hierarchical relationships among the concepts. In spite of the fact that KBs play an important role for biomedical NLP tasks, to the best of our knowledge, there is little work on integrating KBs with word embedding models for biomedical NLP tasks.

In this paper, we propose models which incorporate concept information from such external knowledge as word clusters in pre-trained word embeddings or different knowledge bases. This prior concept knowledge is leveraged and fed into a neural bag of words model through a weighted composition. We explore two different ways of incorporation and show that our model can achieve near state of art performance on different text classification tasks.

3 The Model

In this paper, we investigate the feasibility of incorporating prior knowledge from pre-trained word embeddings and various knowledge bases into a traditional neural classification model. As an initial task, we aim to find out what kind of knowledge bases can be used for different domains and how the model can benefit from the additional common and specific concept information.

Assume that we have L training examples $\{X^d, y^d\}_{d=1}^{|L|}$, X^d is composed of a word sequence$\{x_i^d\}_{i=1}^{|X^d|}$. Suppose we have M knowledge bases $\mathcal{C}^{(j)}$, $j \in \{1, 2, 3, ..., M\}$ and define a mapping: $\mathcal{V} \rightarrow \mathcal{C}^{(j)}$ from a word into a specific concept or topic, i.e. $\mathcal{C}^{(j)} = \{c_1^j, ..., c_K^j\}$, where $x \in \mathcal{V}$ and $c_k^j \in \mathcal{C}^{(j)}$. With each knowledge base $\mathcal{C}^{(j)}$, similar words are to be gathered in the same group with the same topic or concept. For instance, given the sentence *Since the previous examination much of the ground-glass opacity identified has resolved.* We could have such concept annotations based on different lexical resources:

- **WordNet** *Previous[adj.pertainyms] examination[noun.quantity] opacity[noun.state] identify[verb.peception] resolve[verb.change]*

- **MetaMap** *Previous[Temporal Concept] examination[Therapeutic or Preventive Procedure] opacity[Finding] identified[Qualitative Concept] resolved[Conceptual Entity]*

The question is how to incorporate the word level concept information into existing neural classification models. In the following, we first describe a simple and effective neural bag-of-words model, and explore two different ways of incorporating linguistic concept information into the model. We also find the sources which can provide different concept annotations.

3.1 Neural bag of words model

The Neural bag-of-words model (NBOW) differs from traditional bag-of-words model in that each word in a sequence is represented by a distributed rather than one-hot representation. With the above assumption, the model maps an input document $\{x_i\}_{i=1}^{|X^d|}$ into y with m labels. We first apply a composition function to average the sequence of word embeddings $\boldsymbol{e}(x_i)$ for $x_i \in X$. The output of this composition function is fed into a logistic regression function.

To be specific, in an initial setting of NBOW, we can get an averaged word embedding $\boldsymbol{z}$ for any set of words $\{x_i\}_{i=1}^{|X|}$:

$$\boldsymbol{z} = \frac{1}{|\boldsymbol{X}|} \sum_{i=1}^{|\boldsymbol{X}|} \boldsymbol{e}(x_i).$$

Feeding $\boldsymbol{z}$ to a softmax layer gives probability for each output label:

$$\hat{y} = \text{softmax}(\boldsymbol{W}_s \cdot \boldsymbol{z} + \boldsymbol{b}).$$

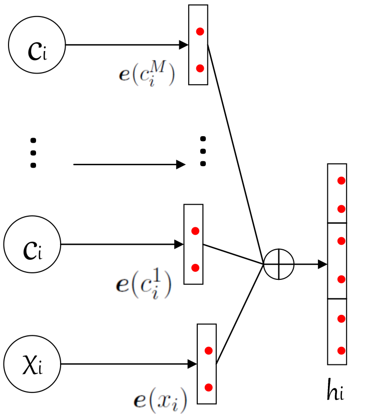
Alternatively, more layers can be created on top of $\boldsymbol{z}$ to generate more abstract representations. The objective function is to minimize the cross entropy error, which for a single training example with true label y is:

$$\ell \ (\hat{y}) = -\sum_{p=1}^{m} y_p \log(\hat{y_p}).$$

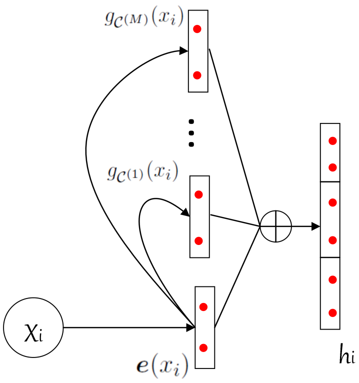
The following section will describe how we extend this NBOW model by integrating linguistic concept information into $\boldsymbol{z}$.

3.2 Incorporating Linguistic Concept Information

Direct mapping: Given a document $\{x_i\}_{i=1}^{|X|}$, we can get the corresponding annotations $\{c_i^j\}_{i=1}^{|X|}$ based on $\mathcal{C}^{(j)}, j = \{1, ..., M\}$, which means additional input is available for the classifier. The question is how we can effectively make use of these annotations based on various $\mathcal{C}^{(j)}, j = \{1, ..., M\}$. In order to represent these concept information, we design two model variants, the first



(a) Direct mapping.



(b) Gated mapping.

Figure 1: Direct and gated mapping.

one is conducted by direct mapping, and the second one is done through gated mapping.

With direct mapping, the embeddings for a specific token x_i and its concept annotation c_i are initialized separately. Therefore, the input for the following composition function is the concatenation of $\boldsymbol{e}(x_i)$ and $\boldsymbol{e}(c_i^j), j = \{1, ..., M\}$. In this case, the new hidden representation for x_i is $\boldsymbol{h}_i$:

$$\boldsymbol{h}_i = \boldsymbol{e}(x_i) \oplus \boldsymbol{e}(c_i^1) \oplus ... \oplus \boldsymbol{e}(c_i^M).$$

Gated mapping: Gated mapping leads to a concept representation by sharing weight with the word representation, the mapping is conducted through a non-linear transformation $g(x)$ instead of direct initialization.:

$$g_{\mathcal{C}^{(j)}}(x_i) = \tanh(\boldsymbol{W}_{\mathcal{C}^{(j)}} \cdot \boldsymbol{e}(x_i) + \boldsymbol{b}_{\mathcal{C}^{(j)}}),$$

where $\boldsymbol{W}_{\mathcal{C}^{(j)}}$ is a three dimensional weight indexing matrix which corresponds to different knowledge bases, $\boldsymbol{b}_{\mathcal{C}^{(j)}}$ is the bias vector. Hence, the new hidden representation is $\boldsymbol{h}_i$:

$$\boldsymbol{h}_i = \boldsymbol{e}(x_i) \oplus g_{\mathcal{C}^{(1)}}(x_i) \oplus ... \oplus g_{\mathcal{C}^{(M)}}(x_i).$$

The resulted gated representation thus computes concept embedding by transforming the original word embeddings from a word semantic space into a concept semantic space based on the given concept annotations.

Figure 1 shows the difference between these two methods. The steps for feeding the newly concatenated word-concept vector $\boldsymbol{h}_i$ into the following layers is the same. But not all words contribute equally to the representation of the document meaning, we further introduce an attention mechanism to extract such words that are important to the meaning of the document and aggregate the representation of those informative words to form a single hidden vector. Specifically, we introduce a context vector $\boldsymbol{q}$,

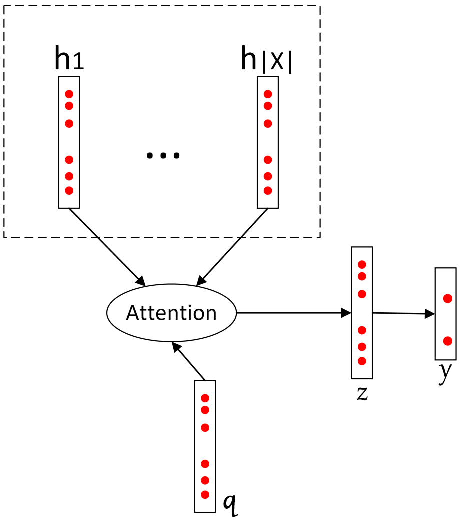


Figure 2: Framework of our model

$$\boldsymbol{u}_i = \tanh(\boldsymbol{W}_q \cdot \boldsymbol{h}_i + \boldsymbol{b}_q),$$

$$\alpha_i = \frac{\exp(\boldsymbol{u}_i^T \boldsymbol{q})}{\sum_{i=1}^{|X|} \exp(\boldsymbol{u}_i^T \boldsymbol{q})},$$

$$\boldsymbol{z} = \textstyle\sum_{i=1}^{|X|} \alpha_i \boldsymbol{h}_i.$$

With $\boldsymbol{z}$, the final prediction is made with a softmax layer:

$$\hat{y} = \text{softmax}(\boldsymbol{W}_s \cdot \boldsymbol{z} + \boldsymbol{b}).$$

Figure 2 gives the framework of our model. The two variants of the model are neural bag of words with either direct or gated mapping.

3.3 Sources of concept information

We collect concept annotation from three sources: the word clusters returned by GloVe word embeddings, lexical categories from WordNet, and biomedical concepts from MetaMap.

Word clusters from GloVe word embeddings Global K-means clustering algorithm (Likas et al., 2003) is used to create K word clusters from pre-trained GloVe word embeddings (Pennington et al., 2014). The algorithm is conducted in an incremental approach: To create K word clusters, all intermediate problems with $1, 2, ..., K - 1$ clusters are sequentially solved. The core idea of this method is that an optimal solution for a clustering problem with K clusters can be obtained by using a series of local optimal searches. We tested different K which varies from 50 to 200.

WordNet lexical categories By using WordNet lexical categories we have mapped each word remained after the preprocessing to lexical categories. WordNet 3.0 (Miller, 1995) offers categorization of 155,287 words into 44 WordNet lexical categories. Since many words may have different categories, a word sense disambiguation technique is required in order to not add noise to the later concept mapping. We use disambiguation by context (Hotho et al., 2003). This technique returns the concept which maximizes a function depending on the conceptual vicinity.

MetaMap concepts MetaMap (Aronson, 2001) provides 133 specific concepts for biomedical words.

4 Datasets and implementation details

In this section, we introduce our experimental datasets and some implementation details.

4.1 Datasets

We select 3 datasets of different sizes, corresponding to varying classification tasks. Some statistics about these datasets is summarized in Table 1.

20 Newsgroups This is a news categorization dataset (Lang, 1995). It has a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Some of the newsgroups are very closely related to each other, while others are highly unrelated. Each news belongs to one out of 20 labels.

IMDB This core dataset (Maas et al., 2011) contains 50,000 reviews which are divided evenly into 25k train and 25k test sets. The overall distribution of labels is balanced (25k positive and 25k negative).

CT reports Additionally, we use 1000 CT scan reports (Martinez et al., 2015) with either positive or negative labels for fungal disease. These reports have technical medical content and highly specialized conventions, which are arguably the most distant genre from the above three datasets.

4.2 Implementation details

Preprocessing The same preprocessing steps were used for all the datasets. We lower-cased all the tokens, removed stop words and replaced those low-frequency tokens with a UNK representation. All the numbers were replaced with a NUM symbol. Specifically, since all the CT reports were obtained from local hospitals, any potentially identifying information such as name, address, age, birthday and gender were removed. For each CT report, we used the free-text section, which contains the radiologist's interpretation of the scan and the reason for the requested scan as written by clinicians.

Word embeddings For the first 2 datasets, we initialized word embeddings with the GloVe word vectors with 400 thousand vocabulary and 6 billion tokens. For the out-of-vocabulary words, we initialized their word embeddings randomly. For pathology reports, we have another 6000 CT documents which are unannotated by doctors. Therefore, a specific biomedical word embedding was randomly initialized with both unlabelled and labelled training data alongside other model parameters. The embedding dimension is set to be 100 for biomedical text and 300 for news and review text.

Learning and hyperparameters To avoid overfitting, a dropout rate 0.3 is used on the word embedding layer (Srivastava et al., 2014). Mini-batch size is 32, the update method is AdaGrad (Duchi et al., 2011), the initial learning rate is 0.01. During training, we conduct experiments in the following to see if word embedding update during training can have an effect on the model performance. For all experiments, we iterate over the training set for 10 times, and pick the model which has the least training loss as the final model, all the

Dataset	Classes	Number of docs.				Vocab. size
		Total	Training	Development	Test	
20News	20	18.8k	10.3k	1k	7.5k	218k
IMDB	2	50k	23k	2k	25k	116K
CT reports	2	1.0k	-	-	-	4.4k

Table 1: Statistics about the four datasets used in our experiments.

results on the test sets are performed from the final models.

5 Experiments

We evaluate the two variants of our model with 5 types of concept information incorporation: word clusters returned by applying K-means to GloVe word vectors, lexical categories returned from WordNet, biomedical concepts from MetaMap, both clusters returned from GloVe clusters and WordNet, and all the concepts from the three knowledge sources. We first do concept annotation from GloVe word clusters and manage to find out the best K for clustering GloVe words, then see whether concept information from different knowledge bases help and compare in each case to several strong baselines.

5.1 Choosing the number of GloVe embedding clusters

We assumed the number of concept clusters (K) would have an impact on the model performance, therefore we have to test K for different datasets accordingly. For 20News and IMDB, we used 10% of the training set as development data. For CT reports, we used 10-fold cross validation. In the following, we use NBOW-DM and NBOW-GM to represent our two model variants, the direct mapping and gated mapping variants, respectively. Figure 3 and 4 show the test accuracy of our two model variants with various K from 50 to 200, we find that the best results can be got when K is 120,150 for 20News and IMDB respectively. This scenario is in our expectation that for larger datasets, there tend to be more groups of concepts. For CT reports, we notice there is a large fluctuation, partly because the GloVe embeddings we used are trained on top of Wikipedia text which are not specific for the biomedical terms in CT reports. In the following comparison experiment, the most appropriate K (120, 150, 90) is set accordingly for these 3 datasets.

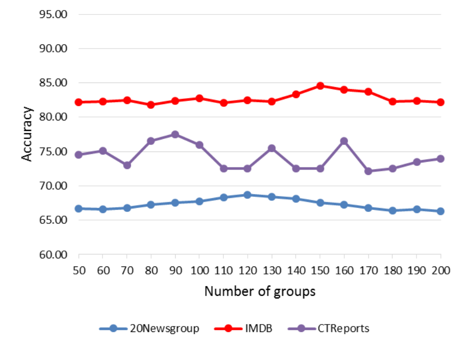


Figure 3: Accuracy of NBOW-DM-GloVe clusters

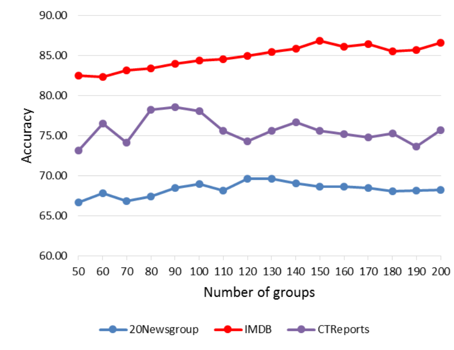


Figure 4: Accuracy of NBOW-GM-GloVe clusters

5.2 Model effectiveness with fixed word embeddings

First, we conduct several experiments in which the pre-trained word embeddings is fixed during training. We hope to answer two questions via these experiments: 1) whether the concept incorporation from different lexical resources provide additional information; 2) which incorporation method is better, direct or gated mapping. As shown in Table 2, concept information from GloVe clusters, WordNet and MetaMap helps propagate the general topic expression to classifiers. Also, gated mapping brings more benefits than direct mapping.

Model	Datasets		
	20News	IMDB	CTReports
NBOW-fixed	66.50	84.01	74.53
NBOW-DM-fixed-noAttention			
-GloVe	66.83	84.25	74.10
-WordNet	66.43	84.33	74.25
-GloVe+WordNet	66.60	84.35	74.40
-MetaMap	-	-	75.02
-All	-	-	75.25
NBOW-GM-fixed-noAttention			
-GloVe	67.10	84.35	74.18
-WordNet	67.53	84.53	74.35
-GloVe+WordNet	67.42	84.76	74.58
-MetaMap	-	-	75.53
-All	-	-	76.15
NBOW-DM-fixed-Attention			
-GloVe	66.43	84.75	74.37
-WordNet	**67.54**	84.85	74.35
-GloVe+WordNet	66.12	**85.02**	74.80
-MetaMap	-	-	75.50
-All	-	-	**76.24**
NBOW-GM-fixed-Attention			
-GloVe	68.13	86.15	75.20
-WordNet	68.20	86.26	75.30
-GloVe+WordNet	**68.55***	**86.85***	75.37
-MetaMap	-	-	76.82
-All	-	-	**77.10***

Table 2: Evaluation with fixed word embeddings during training.

Model	Datasets		
	20News	IMDB	CTReports
NBOW	67.62	84.32	76.20
DCNN	68.13	85.90	76.95
SVM			
-unigram	63.00	75.43	63.46
-unigram+Glove	64.20	75.53	63.81
-unigram+WordNet	64.13	75.62	63.58
-unigram+GloVe+WordNet	64.35	76.03	63.81
-unigram+MetaMap	-	-	64.52
-unigram+All	-	-	64.82
NBOW-DM-noAttention			
-GloVe	67.83	84.40	76.25
-WordNet	67.43	84.58	76.39
-GloVe+WordNet	67.60	84.83	76.45
-MetaMap	-	-	77.92
-All	-	-	78.14
NBOW-GM-noAttention			
-GloVe	68.15	85.12	77.13
-WordNet	68.65	85.65	77.15
-GloVe+WordNet	68.92	86.10	77.50
-MetaMap	-	-	78.20
-All	-	-	79.05
NBOW-DM-Attention			
-GloVe	**68.69**	84.62	77.50
-WordNet	67.53	84.80	76.30
-GloVe+WordNet	67.60	**85.16**	79.21
-MetaMap	-	-	78.02
-All	-	-	**80.43**
NBOW-GM-Attention			
-GloVe	69.62	86.85	78.52
-WordNet	68.50	89.43	77.43
-GloVe+WordNet	**69.82***	**90.10***	79.80
-MetaMap	-	-	80.26
-All	-	-	**82.56***

Table 3: Evaluation with updated word embeddings during training.

5.3 Comparison with the state-of-art with updated word embeddings

As we had wondered if update word embeddings during training would enhance the model performance, we re-ran the experiments with all the same settings except that the original word vectors could be updated. Most current neural models for text classification are variants of either recurrent or convolutional networks. Besides NBOW, we use another two strong baselines: the first one is DCNN (Kalchbrenner et al., 2014) which extends traditional CNN with dynamic k-max pooling, the second one is SVM with unigram features as well as additional concept annotations from the same five different sources. We also test our two model variants without the attention layer, in which the attention computation is replaced by an averaged summation.

As shown in Table 3, on 20 Newsgroup, our first model variant NBOW-DM-Attention achieves slightly better result on 20 Newsgroup with the incorporation of GloVe clusters. It is also noticed that the incorporation of WordNet categories hurt the model in some degree, we analyze that it is caused by the limited vocabulary size compared to that of GloVe, as well as the intermediate disambiguation step during concept annotation. Our second model variant NBOW-GM-Attention with GloVe amd WordNet concept embeddings achieves best results on 20 Newsgroup, compared with the baselines and the first model variant NBOW-DM-Attention. While on IMDB, NBOW-GM-Attention with concept incorporation from GloVe and WordNet achieves the best, even if NBOW-DM-Attention with the same setting does not beat DCNN. On CT Reports, both our two model variants achieve better accuracy with all the group information from GloVe, WordNet and MetaMap. Besides, it is noticed that the variants with attentions generally perform better than those with no attentions. Overall, the results show that NBOW-GM-Attention generally performs better than NBOW-DM-Attention, which indicates that the concept incorporation by gated mapping is more reliable than that of a direct concept embedding, and the incorporation of appropriate concept information with our second model variant makes a contribution to the classification tasks.

5.4 Error analysis and improvement

CT reports, which have technical content and highly specialized conventions, are arguably the most distant genre from news and movie reviews among those we consider. Therefore, we manually check the false predictions returned by our best model above. It turns out the classifier cannot capture two kinds of patterns: In the first, there is some context information provided in the report which contains comparison with a previous patient record, e.g. in the sentence "hypodense liver lesion in segment has significantly decreased in size from 12mm to 7mm", the diagnosis of whether the patient is infected or not relies on the magnitude of "decrease", which is highly professional. Second, human label noise occurs in some cases when doctors will not make immediate decisions, for instance "suspicious for infection" and "likely to be infected" happen in both positive and negative reports.

In order to see whether modeling context information can help or not, we conduct two transformation for $\boldsymbol{h}_i$ to get a new $\tilde{\boldsymbol{h}}_i$, one is convolution-based (CNN-GM): $\tilde{\boldsymbol{h}}_i = \tanh(\boldsymbol{W}_c \cdot (\boldsymbol{h}_{i-1} \oplus \boldsymbol{h}_i \oplus \boldsymbol{h}_{i+1}) + \boldsymbol{b}_r)$, the other is recurrence-based (RNN-GM): $\tilde{\boldsymbol{h}}_i = \tanh(\boldsymbol{W}_h \cdot \boldsymbol{h}_i + \boldsymbol{W}_r \cdot \tilde{\boldsymbol{h}}_{i-1} + \boldsymbol{b}_r)$. Thus, in the above NBOW-GM settings, $\tilde{\boldsymbol{h}}_i = \boldsymbol{h}_i$. We use the three corresponding gated mapping variant with the best settings, and compare the number of parameters and the average running time per epoch. Table 4 shows that RNN-GM generally performs best at the cost of more parameters and training time per epoch. In contrast, CNN-GM is a trade-off between model complexity and performance. All timing experiments are specific for CT reports and performed on a single core of an Intel I5 processor with 8GB of RAM.

Model	20News	IMDB	CTReports	Parameters	Time(s)
NBOW-GM	69.82	90.10	82.56	3480.40k	15s
CNN-GM	71.58	**91.23**	86.13	3488.05k	21s
RNN-GM	**72.00**	91.05	**86.96**	3500.50k	30s

Table 4: Evaluation of gated mapping with convolution or recurrence transformation.

6 Conclusions and future work

In this paper, we propose two different methods for incorporating concept information from external knowledge bases into a neural bag of words model: the neural bag of words with either direct mapping (NBOW-DM) or gated mapping (NBOW-GM), which leverages both the word and concept representation through multiple hidden layers before classification. The model with gated mapping does better than direct mapping, and performs competitively with more complicated neural models as well as a traditional statistic model on different text classification tasks, and achieves good results on a practical biomedical text classification task. Moreover, our two model variants are also time efficient. They generally require less training time than their counterparts, which allow them to be used for datasets where few annotation is available or manual annotation is expensive.

For future work, we will consider using some global semantic information such as Rhetorical Structure Theory (RST), which is a theory of discourse that has enjoyed popularity in NLP. RST posits that a document can be represented by a tree whose leaves are elementary discourse units. We seek to develop approaches to combine local linguistic and global semantic knowledge into our model.

On the other hand, our proposed method takes the information from outsourced knowledge bases into account and ignores the information of unlabelled data. We will considering using deep reinforcement learning to learn how to select the query unlabelled data points in a sequential manner, formulated as a Markov decision process. With more labels as well as information from some prior knowledge bases, our model can be developed for large scale text processing and analysis.

References

Alan R Aronson. 2001. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, page 17.

Robert H Baud, Anne-Marie Rassinoux, Christian Lovis, Judith Wagner, Vincent Griesser, Pierre-Andre Michel, and Jean-Raoul Scherrer. 1996. Knowledge sources for natural language processing. In *Proceedings of the AMIA Annual Fall Symposium*. American Medical Informatics Association, page 70.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32(suppl_1):D267–D270.

James J Cimino. 2000. From data to knowledge through concept-oriented terminologies: experience with the medical entities dictionary. *Journal of the American Medical Informatics Association* 7(3):288–297.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(Jul):2121–2159.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Andreas Hotho, Steffen Staab, and Gerd Stumme. 2003. Ontologies improve text document clustering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, pages 541–544.

Mohit Iyyer, Varun Manjunatha, Jordan L Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *ACL (1)*. pages 1681–1691.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* .

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*. volume 10, pages 331–339.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1188–1196.

Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. *Pattern Recognition* 36(2):451–461.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 142–150. http://www.aclweb.org/anthology/P11-1015.

David Martinez, Michelle R Ananda-Rajah, Hanna Suominen, Monica A Slavin, Karin A Thursky, and Lawrence Cavedon. 2015. Automatic detection of patients with invasive fungal disease from free-text computed tomography (CT) scans. *Journal of biomedical informatics* 53:251–260.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

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*. pages 3111–3119.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38(11):39–41.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing(EMNLP)*. volume 14, pages 1532–1543.

Alan L Rector. 1995. Coordinating taxonomies: Key to re-usable concept representations. In *Conference on Artificial Intelligence in Medicine in Europe*. Springer, pages 15–28.

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013a. Parsing with compositional vector grammars. In *ACL (1)*. pages 455–465.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pages 151–161.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts, et al. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. volume 1631, page 1642.

Kent A Spackman, Keith E Campbell, and Roger A Côté. 1997. SNOMED RT: a reference terminology for health care. In *Proceedings of the AMIA annual fall symposium*. American Medical Informatics Association, page 640.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .