Text Classification with Topic-based Word Embedding and Convolutional Neural Networks

Haotian Xu
Department of Computer Science
Wayne State University
Detroit, Michigan 48202
htxu@wayne.edu

Ming Dong*
Department of Computer Science
Wayne State University
Detroit, Michigan 48202
mdong@wayne.edu

Dongxiao Zhu
Department of Computer Science
Wayne State University
Detroit, Michigan 48202
dzhu@wayne.edu

Alexander Kotov
Department of Computer Science
Wayne State University
Detroit, Michigan 48202
kotov@wayne.edu

April Idalski Carcone
Pediatric Prevention Research Center
Wayne State University
Detroit, Michigan 48201
acarcone@med.wayne.edu

Sylvie Naar-King
Pediatric Prevention Research Center
Wayne State University
Detroit, Michigan 48201
snaarkin@med.wayne.edu

ABSTRACT

Recently, distributed word embeddings trained by neural language models are commonly used for text classification with Convolutional Neural Networks (CNNs). In this paper, we propose a novel neural language model, Topic-based Skip-gram, to learn topic-based word embeddings for biomedical literature indexing with CNNs. Topic-based Skip-gram leverages textual content with topic models, e.g., Latent Dirichlet Allocation (LDA), to capture precise topic-based word relationship and then integrate it into distributed word embedding learning. We then describe two multimodal CNN architectures, which are able to employ different kinds of word embeddings at the same time for text classification. Through extensive experiments conducted on several real-world datasets, we demonstrate that combination of our Topic-based Skip-gram and multimodal CNN architectures outperforms state-of-the-art methods in biomedical literature indexing, clinical note annotation and general textual benchmark dataset classification.

CCS Concepts

•**Computing methodologies** → **Natural language processing;**

Keywords

text classification; convolutional neural networks; word embeddings; medical subject headings

*Corresponding author

BCB '16, October 02-05, 2016, Seattle, WA, USA

DOI: http://dx.doi.org/10.1145/2975167.2975176

1. INTRODUCTION

As the amount of biomedical textual data in MEDLINE of the US National Library of Medicine (NLM) is growing exponentially, the indexing of biomedical articles is becoming a much more difficult task. Medical Text Indexer (MTI)[1] [1] has been assigned to this task as a support tool which produces (semi-)automated recommendation indexing based on predefined Medical Subject Headings (MESH)[2]. Meanwhile, biomedical literature indexing can also be viewed as a classification over textual data into a set of predefined classes. However, as discussed in [26, 29], traditional machine learning algorithms, including Naive Bayes, Support Vector Machine and Logistic Regression, cannot outperform MTI system without ensemble.

Recently, CNN models have achieved remarkably strong performance in natural language processing and become commonly used architectures for text classification [11, 12, 14, 31]. As input features of CNNs, various types of word vector representations have been proposed. Generally speaking, there are two model families to represent words with real-valued vectors: 1)matrix factorization methods, such as [7, 17] and 2)local window-based methods, such as [2, 6, 21]. Both families have their own pros and cons. Although matrix factorization methods do not require much domain expertise of word embedding and efficiently leverage statistical information of corpora, their main problem is that most frequent words (or characters) have a large negative impact on word similarity measure, which leads to poor performance on word analogy tasks. Local window-based methods perform better on analogy tasks, but they poorly utilize statistical information about corpus because these models are trained on separate local windows of content.

In the presented work, we propose a novel word embedding learning approach, which provides topic-based semantic word embeddings and two CNN architectures, which can utilize multiple word representations simultaneously for text classification. Specifically, our framework first leverages the whole text corpus with topic models to capture semantic re-

[1]http://ii.nlm.nih.gov/MTI/index.shtml
[2]https://www.nlm.nih.gov/pubs/factsheets/mesh.html

lationship between words and then take it as the input for word representation learning using Topic-based Skip-gram with a novel objective function. Then, these topic-based word representations are used together with other state-of-the-art word embeddings for text classification in multimodal CNN models. Specifically, the main contributions of this paper are summarized as follows:

- We develop a word embedding learning model, Topic-based Skip-gram, which captures word semantic relationship with topic models, e.g., LDA, and then integrate it into distributed word embedding learning with a novel objective function.
- We introduce two complementary multimodal CNN architectures that are able to simultaneously take multiple kinds of word embeddings as inputs for text classification.
- We combine the proposed topic-based word embedding and other state-of-the-art word embeddings as inputs to the proposed multimodal CNN architectures. Our experiments conducted on several real-world datasets show that combination of the proposed topic-based word representations and our multimodal CNNs outperforms state-of-the-art word representations in various text classification tasks, including indexing of biomedical articles.

The rest of this paper is organized as follows. In Section 2, we review related work in biomedical literature indexing and word embedding learning. The details of our word representation learning approach and multimodal CNN models are introduced in Section 3. In Section 4, we demonstrate that our topic-based word embedding produces competitive results with CNN architecture and outperforms state-of-the-art approaches with our multimodal CNN models in three case studies. At last, we conclude in Section 5.

2. RELATED WORK

2.1 Indexing of Biomedical Literature

Our work shares the high-level goal of biomedical literature indexing with many previous works, such as USI [9], MeSHLabeler [16], MeSH Now [18] and Atypon [22]. Several other works [26, 29] tried to improve the MTI system with automatic machine learning methods. Among them, Yepes et al. [29] pointed out that ensemble of classic machine learning methods can outperform indexing performance of MTI. Rios and Kavuluru [26] surpassed MTI performance by utilizing CNNs for sentence-level textual classification [12] with word embeddings trained by the Skip-gram model [21], which is more closely related to our work. However, these works focus on utilizing classic machine learning methods for biomedical literature indexing, while we propose a novel Topic-based Skip-gram for learning topic-based semantic word representations and obtain state-of-the-art classification performance with deep learning architectures.

2.2 Topic Models

Topic models are probabilistic generative models to discover main themes of documents. These models share the same assumptions: 1) they posit there are a set of latent topics, which are multinomial distributions over vocabulary; 2) each document is a mixture of these topics. Recently, topic models have become a popular tool for text classification [19, 24], image classification [8, 25], transfer learning [5, 27] and unsupervised analysis of textual data [3, 4]. As one of the most commonly used unsupervised topic models, Latent Dirichlet Allocation (LDA) [4] can extract semantic information from corpora. The basic assumption of LDA is that each document is a mixture of topic proportions and each topic is a distribution over fixed vocabulary. In this paper, we employ LDA to identify topic-based semantic relationships between words in each corpus.

2.3 Word Embedding Learning Methods

Recently, Mikolov et al. introduced an algorithm for learning fixed length distributed representations of words in a vector space, the Skip-gram model [20], which is a single-layer neural network based on inner products between word vectors. As one of the local window-based methods, Skip-gram's objective is to learn word embeddings that can predict the textual content of a word given the word itself. Through experiments on word and phrase analogy tasks, this model demonstrated its capacity to capture linguistic relationships between word vectors. However, Skip-gram model suffers from the disadvantage that it does not utilize the co-occurrence statistics of the corpus. Instead, Skip-gram scans textual corpus with local context windows, which fails to make use of statistical information of the whole corpus. Pennington et al. [23] took the advantages of both global matrix factorization and local content window-based methods by training their model only on nonzero elements in the word co-occurrence matrix. Different from their approach, Topic-based Skip-gram leverages global statistical information of the whole corpus with LDA and learns the semantic information with local content windows.

2.4 CNNs for Text Classification

A number of CNN architectures have been developed for text classification [11, 12, 14, 31]. Kalchbrenner et al. [11] focused on sentence modeling with a CNN-based model for word-level input. Zhang and LeCun [31] concentrated on character-level input with a very deep CNN architecture which requires a large amount of training data and training time. Lai et al. [14] proposed a model which combines Recurrent Neural Networks (RNN) with CNN. Kim [12] proposed a two-layer CNN model for sentence-level text classification with single kind of word embeddings. This model is simple but very effective for text classification. Our multimodal approach is inspired by this model. In contrast to the architecture described by Kim, our multimodal approaches are able to simultaneously take multiple kinds of word representations as inputs.

3. OUR APPROACH

In this section, we first present technical details of Topic-based Skip-gram for learning topic-based semantic word embeddings and then introduce two multimodal CNN architectures which employ multiple kinds of word embeddings as inputs for text classification.

3.1 Topic-based Skip-gram

Topic-based Skip-gram identify semantic relationship between words from corpus using LDA and then integrate it into word representation learning with a novel objective

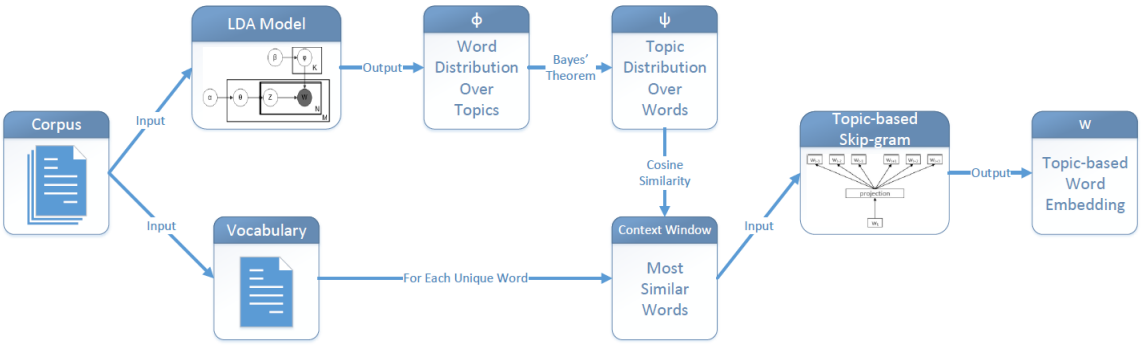

Figure 1: Workflow of Topic-based Skip-gram. Given the training corpus, we first get its vocabulary and train the LDA model on the corpus. As the output of LDA, word distribution over topics ϕ is then used for computation of topic distribution over words ψ. For each unique word in the vocabulary, we find out the most similar words for it based on the cosine similarity of ψ as the input to proposed Topic-based Skip-gram. At last, we get topic-based word embedding.

function. The workflow is shown in Fig. 1 and we will introduce the details in this subsection.

3.1.1 *Leveraging Topic-based Semantic Information with LDA*

LDA. The basic idea of LDA is that each document d is a distribution over K latent topics and each topic is a distribution over V unique words in the dictionary. Given a corpus of M documents and each document has N_m words, the generative process of LDA is as follows:

1. Choose $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$
   $\boldsymbol{\theta}$ denotes topic distribution over documents. Each document has its own $\boldsymbol{\theta}$, which needs to be estimated during the training stage. Each $\boldsymbol{\theta}$ is a vector of length K, where K is the number of topics and chosen manually at the beginning of training. $\boldsymbol{\alpha}$ is the hyperparameter of document-topic distribution.

2. Choose $\boldsymbol{\phi} \sim \text{Dirichlet}(\boldsymbol{\beta})$
   $\boldsymbol{\phi}$ is word distribution over topics, also known as topic in [4], which is a matrix of K rows and V columns. Element $\phi_{i,j}$ equals $p(w_j|z_i)$, which is the probability of generating word w_j given this word belonging to topic z_i. $\boldsymbol{\beta}$ is the hyperparameter of topic-word distribution.

3. For each of the N words w_n in each document d_m of the M documents in the corpus:

   (a) Choose a topic $z_n \sim \text{Multinomial}(\boldsymbol{\theta})$
   The topic indicator z_n is the topic k assigned to word w_n.

   (b) Generate a word $w_n \sim \text{Multinomial}(z_n, \boldsymbol{\beta})$
   Generate a word as w_n, which is the nth unique word in the dictionary, from Multinomial distribution $p(w_n|z_n, \boldsymbol{\beta})$.

Topic-based Semantic Information of Corpus. In this paper, we treat the topic distribution over words $\boldsymbol{\psi}$ as topic-based semantic information of corpus for learning word embeddings. $\boldsymbol{\psi}$ is a $V \times K$ matrix. Its element $\psi_{i,j}$ is equal to $p(z_i|w_j)$, which is the probability for word w_j to be assigned to topic z_i. It can be approximated with word distribution over topics $\boldsymbol{\phi}$ based on Bayes' theorem:

$$p(z_i|w_j) = \frac{p(w_j|z_i) \cdot p(z_i)}{p(w_j)}, \tag{1}$$

where $p(z_i)$ is the marginal probability of topic z_i and $p(w_j)$ denotes the marginal probability of word w_j in the dictionary. $p(z_i)$ and $p(w_j)$ can be calculated as follows:

$$p(z_i) = \frac{\sum_{m=1}^{M} z_i^m}{M}, \tag{2}$$

$$p(w_j) = \frac{\sum_{m=1}^{M} N_m^j}{\sum_{m=1}^{M} N_m}, \tag{3}$$

where z_i^m is the topic proportion of z_i in document d_m and N_m^j is the count of word w_j in the document d_m. The topic-based semantic information matrix $\boldsymbol{\psi}$ is then used as training data in the word embedding learning step.

3.1.2 *Learning Topic-based Word Embeddings*

Skip-gram. The training objective of Skip-gram [21] is to learn distributed word representations which aim at predicting the surrounding words in the documents. Given a training corpus of T words $w_1, w_2, w_3, \cdots, w_T$, the learning objective of the Skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leqslant j\leqslant c, j\neq 0} \log p(w_{t+j}|w_t), \tag{4}$$

where c is the size of training content. In other words, given a local window of size $2 \cdot c + 1$, the objective of Skip-gram model is to maximize prediction log probability of the $2 \cdot c$ words $w_{t-c}, w_{t-c+1}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+c-1}, w_{t+c}$ given the word w_t in the center.

Learning Semantic Word Embeddings. We propose a novel training objective for Topic-based Skip-gram that is to learn distributed word embeddings which are useful to predict words with similar topic-based semantic information. The basic assumption of Topic-based Skip-gram is that if topic distributions of two words $\boldsymbol{\psi_i}$ and $\boldsymbol{\psi_j}$ have a large cosine similarity between each other, then these two words share similar topic-based semantic information. Given a dictionary of N unique words $w_1, w_2, w_3, \cdots, w_N$ of a corpus,

the objective of Topic-based Skip-gram model is to maximize the average log probability

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{-c\leqslant j\leqslant c, j\neq 0}\log p(w_{n+j}|w_n). \quad (5)$$

In other words, given half window size c (s.t. window size is $2c+1$) and a word in the dictionary w_n, the training objective of Topic-based Skip-gram is to maximize prediction log probability of the top $2c$ words similar to w_n. The probability $p(w_{n+j}|w_n)$ is defined using $softmax$ function

$$p(w_{n+j}|w_n) = \frac{\exp(v_{w_{n+j}}^{\top} v_{w_n})}{\sum_{1\leqslant i\leqslant N, i\neq n}\exp(v_{w_i}^{\top} v_{w_n})}, \quad (6)$$

where v_{w_n} is the vector representation of word w_n. In practice, the cost of computing $\triangledown \log p(w_{n+j}|w_n) \propto N$, where N can be very large ($10^6 - 10^8$ unique words).

Optimization. Same with Skip-gram, we use Negative Sampling [21] to optimize the objective function of Topic-based Skip-gram. In Negative Sampling, $p(w_{n+j}|w_n)$ is replaced as

$$\log\sigma(w_{n+j}|w_n) + \sum_{i=i}^{k} E_{w_i\sim P_n(w)}[\log\sigma(-v_{w_i}^{\top} v_{w_n})]. \quad (7)$$

The idea is to distinguish target word w_{n+j} from k noise words which are drawn from noise distribution $P_n(w)$ using logistic regression by maximizing the probability of target word (first item) and minimizing the probability of noise words (second term). According to results reported in [21], we choose $k = 15$ and $P_n(w) \sim \frac{U(w)^{0.75}}{Z}$,where $U(w)$ is unigram distribution.

Time efficiency. Given a dataset of N unique words and L words in total, proposed Topic-based Skip-gram optimizes N word windows and Skip-gram optimizes L windows. Note that $N \ll L$ in most cases. Furthermore, Topic-based Skip-gram can also work with other semantic indexing models in addition to LDA, which may significantly expedite the training process.

We summarize the learning procedure for topic-based semantic word embedding in Algorithm 1.

Algorithm 1 Topic-based Skip-gram

Input: Raw training textual corpus $\boldsymbol{\mathcal{D}}$; Topic number K, Hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ for LDA, Half window size c
Output: Topic-based semantic word embedding $\mathcal{W}$
procedure GetWordEmbedding
  $\boldsymbol{\phi} = LDA(\boldsymbol{\mathcal{D}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, K)$ ▷ Train LDA model on the corpus $\boldsymbol{\mathcal{D}}$ and get word distribution over topics $\boldsymbol{\phi}$
  for Each topic z_i **do**
    Compute marginal probability of each topic $p(z_i)$ with Eq. (2)
  end for
  for Each word w_j **do**
    Compute marginal probability of each word $p(w_j)$ with Eq. (3)
  end for
  Compute topic distribution over words $\boldsymbol{\psi}$ based on Eq. (1), (2) and (3)
  for Each word w_j **do**
    Find $2c$ words with most similar topic distribution over words to w_j according to cosine similarity ▷ These $2c+1$ words are then used as an input window win_j for Topic-based Skip-gram
  end for
  $\mathcal{W}$=Topic-based-Skip-gram($\boldsymbol{win}$) ▷ Take all word windows $\boldsymbol{win}$ as input of Topic-based Skip-gram to learn topic-based word embedding $\mathcal{W}$ based on the objective function in Eq. (5)
end procedure

3.2 Multimodal CNN Architectures

In this part, we first introduce a single channel CNN model [12], which is used as baseline architecture in the experiments. Then we will describe the two proposed multimodal CNN architectures which can take multiple types of word embeddings with different length.

3.2.1 Baseline CNN

The baseline CNN has one input layer, one convolution layer, one sub-sampling layer and one fully connected layer. Although one output neuron with $sigmoid$ or $tanh$ function is sufficient for binary classification, we choose multiple neurons with $softmax$ function to make it easier to adopt CNN models for multi-class classification. The details of each layer are described as follows.

Input layer. Formally, we denote $\mathbf{x_i} \in \mathbb{R}^k$ as the k-dimensional word representation for the ith word in a sentence. A sentence of length n is denoted as

$$\mathbf{X_{1:n}} = \mathbf{x_1} \oplus \mathbf{x_2} \oplus \cdots \oplus \mathbf{x_n}, \quad (8)$$

where $\oplus$ is the concatenation operator. By this, each input sentence is represented as a $n \times k$ matrix. In practice, short sentences are padded with zeros to same length, such that, each matrix shares the same size.

Convolution layer. A convolution filter $\mathbf{w} \in \mathbb{R}^{h\times k}$, which is applied to a window of h words of k-dimensional embeddings, produces a new feature. For instance, given a window of words $\mathbf{X_{i:i+h-1}}$ and a bias term $b \in \mathbb{R}$, a new feature c_i is generated by

$$c_i = f(\mathbf{w}\cdot\mathbf{X_{i:i+h-1}} + b), \quad (9)$$

where f is a non-linear function. In our case, we apply the element-wise function Rectified Linear Unit (ReLU) to the input matrices:

$$ReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Each filter produces a feature map $\mathbf{c} = [c_1, c_2, \cdots, c_{n-h+1}]$ from every possible window $\{\mathbf{X_{1:h}}, \mathbf{X_{2:h+1}}, \cdots, \mathbf{n-h+1:n}\}$ of a sentence of length n. In [12], multiple layers of various sizes are applied in the convolution layer, and multiple feature maps are generated.

Sub-sampling layer. There are several sub-sampling methods, such as average pooling, median pooling and max pooling. In this case, we apply max pooling over each feature map produced by the convolution layer and take the maximum element $\hat{c} = max\{\mathbf{c}\}$. Let's denote features generated by this max pooling layer as

$$\hat{\mathbf{c}} = \hat{c}_1 \oplus \hat{c}_2 \oplus \cdots \oplus \hat{c}_m, \quad (11)$$

where m is the number of feature maps.

Fully connected layer. Given $\hat{\mathbf{c}}$ as the input, the fully connected layer produces

$$P(Y = i|\hat{\mathbf{c}}, \boldsymbol{\theta}) = softmax_i(\mathbf{W} \cdot (\hat{\mathbf{c}} \circ \mathbf{r}) + b), \quad (12)$$

where Y is the prediction, $\boldsymbol{\theta}$ denotes parameters $\{W, b\}$, $\mathbf{W}$ denotes weights, $\circ$ denotes the element-wise multiplication operator and $\mathbf{r} \in \mathbb{R}^m$ is a dropout mask vector of Bernoulli variables with probability p of being zero. During the back propagation stage, only unmarked elements in $\hat{\mathbf{c}}$ are involved in the computation. l_2-norm [10] is also applied to weight matrices W. If $\|W\|_2 > s$ after gradient descent step, we rescale W, such that $\|W\|_2 = s$. Here, s is a manually defined parameter. By applying dropout and l_2-norm, we prevent the overfitting problem.

Optimization. A reasonable training objective is to minimize categorical (or binary) cross-entropy loss. The average loss for each sample is

$$\begin{aligned} Q(\boldsymbol{\theta}) &= \frac{1}{|\boldsymbol{\mathcal{D}}|}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\mathcal{D}}) \\ &= -\frac{1}{|\boldsymbol{\mathcal{D}}|}\sum_{i=1}^{|\boldsymbol{\mathcal{D}}|} \log P(Y = y^i|x^i, \boldsymbol{\theta}), \end{aligned} \quad (13)$$

where x^i is the ith sample in the dataset and y^i is the prediction for it. In this paper, we update the parameters $\boldsymbol{\theta}$ by Adadelta [30], which is an adaptive learning rate approach for Stochastic Gradient Descent.

3.2.2 Multi-channel CNN (CNN-channel)

As shown in the top panel of Fig. 2, CNN-channel model combines two baseline CNN models. More formally, we denote two kinds of word embeddings $\mathbf{x_i^1} \in \mathbb{R}^{k_1}$ and $\mathbf{x_i^2} \in \mathbb{R}^{k_2}$ as k_1- and k_2-dimensional word representations for the ith word in a sentence. So, a sentence of length n can be represented in two ways

$$\mathbf{X_{1:n}^1} = \mathbf{x_1^1} \oplus \mathbf{x_2^1} \oplus \cdots \oplus \mathbf{x_n^1} \quad (14)$$

and

$$\mathbf{X_{1:n}^2} = \mathbf{x_1^2} \oplus \mathbf{x_2^2} \oplus \cdots \oplus \mathbf{x_n^2}, \quad (15)$$

where $\mathbf{X_{1:n}^1}$ is used as the input matrix for the 'top channel' of CNN-channel and $\mathbf{X_{1:n}^2}$ is the input for the 'bottom channel' of CNN-channel. Similarly, after applying convolution and max-pooling layers, $\hat{\mathbf{c}}^\mathbf{1}$ and $\hat{\mathbf{c}}^\mathbf{2}$ are generated. In CNN-channel, they are merged in an element-wise addition fashion

$$\hat{\mathbf{c}} = \hat{\mathbf{c}}^\mathbf{1} + \hat{\mathbf{c}}^\mathbf{2} \quad (16)$$

Here, + denotes element-wise addition. Then we apply the fully connected layer with dropout and softmax output and l_2 regularization as in the baseline CNN model.

3.2.3 Concatenation CNN (CNN-concat)

As shown in the bottom panel of Fig. 2, CNN-concat is also built on top of the baseline CNN model. Different from CNN-channel, $\hat{\mathbf{c}}^\mathbf{1}$ and $\hat{\mathbf{c}}^\mathbf{2}$ are merged by concatenation

$$\hat{\mathbf{c}} = \hat{\mathbf{c}}^\mathbf{1} \oplus \hat{\mathbf{c}}^\mathbf{2} \quad (17)$$

Then $\hat{\mathbf{c}}$ is taken as the input of fully connected layer as in the baseline CNN model. Although CNN-channel and CNN-concat models can be expended to utilize as many types of word embeddings as needed, we only employ two kinds of word representations in our experiments.

3.2.4 Deep Understanding of Multimodal CNNs

Multimodal CNNs vs. original CNN model. Original CNN architecture, which was proposed in [12], can only take one kind of word embedding as input. Meanwhile, our proposed multimodal CNNs are able to simultaneously take multiple types of word embeddings as inputs, which means that multimodal CNNs have stronger learning ability than the original CNN model. Specifically, by combing the topic-based word embedding and local window-based word embeddings, the multimodal CNNs are able to utilize both topic-based semantic relationship and local content information and outperform the original CNN model.

CNN-channel vs. CNN-concat. CNN-channel combines the two kinds of word representations by element-wise addition, commonly used for multi-channel image classification. On the other hand, CNN-concat concatenates two parts together, which introduces more parameters to fit. In other words, CNN-concat has stronger learning ability but needs more training data to preserve from overfitting than CNN-channel. If the dataset contains enough amount of positive samples for binary classification task or is balanced for multi-class classification problem, CNN-concat is a better choice than CNN-channel.

4. EXPERIMENTS

We evaluate our framework by three tasks: 1)indexing of biomedical articles; 2)annotation of clinical text fragments with behavior codes; and 3)classification of benchmark newsgroups. Baselines and state-of-the-art algorithms are compared with our method in these experiments. In our experiments, we used the same code[3] and parameter settings as in [26] for the baseline CNN model. We make implementation of proposed multimodal CNNs publicly available[4].

4.1 Datasets

4.1.1 Indexing of Biomedical Articles

MEDLINE citations.A public dataset[5] of MEDLINE citations from November 2012 to February 2013 is used in this paper. The dataset contains 143,853 citations in total, from which 94,942 citations were selected for training and 48,911 were selected for testing. As in [26], we categorize 29 MeSH terms into three groups according to MTI's performance: check tags, low precision terms and low recall terms. The check tags group is a common set of top 12 MeSH headings routinely considered for almost all articles (e.g. Humans, Female and Male), the low precision group contains 10 MeSH headings with the lowest precision performance using MTI and the low recall group contains 7 MeSH headings with the lowest recall performance using MTI. We build CNN models as binary classifiers for each MeSH to classify if a document belongs to this MeSH term. Note that although only 29 terms are used in this experiment, our framework works for arbitrary number of MeSH terms.

4.1.2 Annotation of Clinical Text Fragments with Behavior Codes

[3]https://github.com/yoonkim/CNN_sentence
[4]https://github.com/HaotianMXu/Multimodal-CNNs
[5]http://ii.nlm.nih.gov/MTI_ML/index.shtml

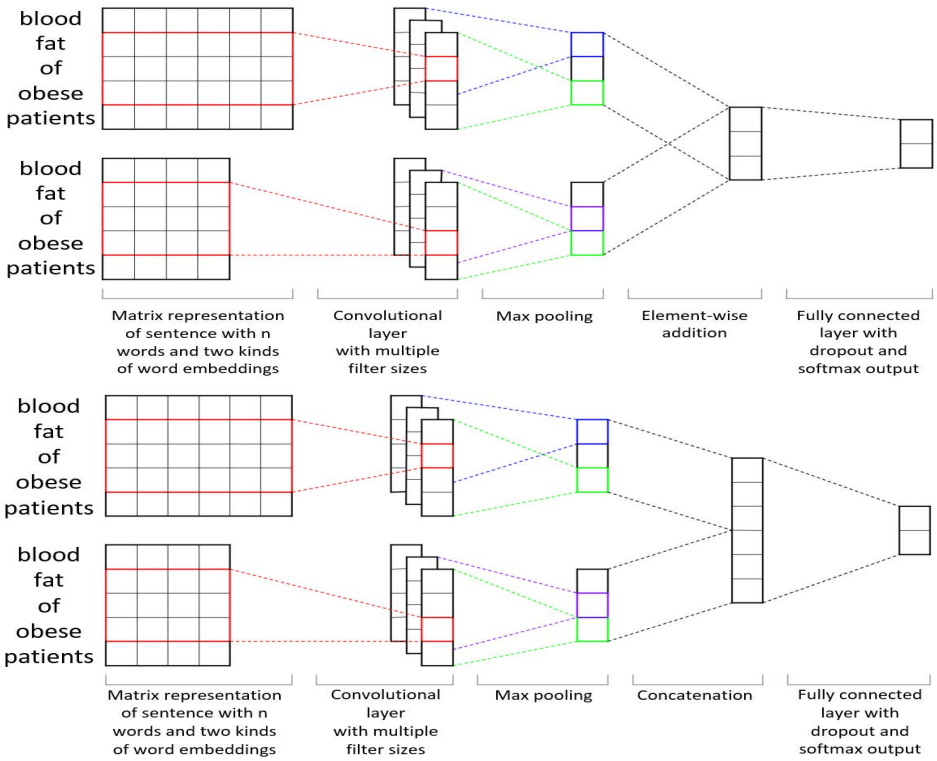

Figure 2: Top panel: architecture of CNN-channel. Bottom panel: architecture of CNN-concat.

Table 1: Description of five behavior code annotation.

Behavior	Definition	Sample Quote
Positive Commitment Language	Statement describing intentions, plans for, and action steps toward changing the current behavior pattern	Well, I've been trying to lose weight, but it really never goes anywhere.
Negative Commitment Language	Statement describing intentions, plans for, and action steps toward maintaining the current behavior pattern	I eat a lot of junk food, like cake and cookies, stuff like that.
Positive Change Talk	Statement describing the desire, ability, reason, or need for changing the current behavior pattern	Hmmm, I guess I need to lose some weight.
Negative Change Talk	Statement describing the desire, ability, reason, or need for maintaining the current behavior pattern	I just don't feel like I want to eat before I'm just not hungry at all.
Ambivalence	Statements that combine positive and negative commitment language and/or change talk	Fried foods may taste good, but it's not good for your health.

Clinical interview fragments. As discussed in [13], behavior code annotation can be treated as a classification problem which assigns a code to each utterance. We use a collection of motivational interviewing-based weight loss sessions, which consists of 11,353 utterances that were manually annotated by two human coders as a golden standard. On top of this dataset, we conduct three behavior code annotation tasks: A) Positive, Negative and Ambivalence; B) Commitment Language, Change Talk and Ambivalence; C) Positive Commitment Language, Negative Commitment Language, Positive Change Talk, Negative Change Talk and Ambivalence. The description of behavior code is listed in Table 1.

4.1.3 *Classification of News Groups*

20 Newsgroups. This publicly available[6] dataset[15] has been widely used to evaluate text classification algorithms. The 20 Newsgroups dataset is a collection of approximately 20,000 newsgroup documents across six categories, i.e., computers, recreation, science, politics, religion and forsale. In this paper, we use four most common classes, which are computers, recreation, science and politics, as a four-class classification task to evaluate our framework.

[6]http://qwone.com/~jason/20Newsgroups/

4.2 Methods Compared

4.2.1 *Baseline Approaches*

The following non-CNN models are used as our baseline:

- **MTI.** Medical Text Indexer, which is commonly used in biomedical literature indexing. We only compare our method with MTI in the indexing task of biomedical articles.
- **Prior-best.** Prior-best is the best-performed method

in the experiments of several classic machine learning methods, including Naive Bayes(NB), Logistic Regression(LR) and Support Vector Machine(SVM). For indexing of biomedical articles, Support Vector Machine with Huber Loss (SVM HL) [28] is also compared.

4.2.2 CNN-based Methods

In our experiments, we compared Topic-based Skip-gram with several baseline and state-of-the-art distributed word embedding learning methods, including:

- **CNN-rand.** Each word embedding is initialized with values drawn from continuous uniform distribution $U \sim [-0.25, 0.25]$. CNN-rand is used as a baseline of CNN-based methods.
- **CNN-gn.** These word vectors were trained by Mikolov et al. [21] on Google News and are publicly available[7]. It is also known as word2vec.
- **CNN-glove.** The word embeddings used in this paper were trained by Pennington et al. [23] [8].
- **CNN-local.** The word representations are trained by Skip-gram on the datasets to classify. The implementation of Skip-gram is publicly available[9].
- **CNN-topic.** These word embeddings are learned by our Topic-based Skip-gram on the datasets to categorize.

These kinds of word embeddings are compared under the baseline CNN architecture. Our two multimodal CNN architectures are also compared in this paper:

- **CNN-channel.** We utilize two kinds of word embeddings for CNN-channel, CNN-local and CNN-topic.
- **CNN-concat.** CNN-local and CNN-topic are employed for CNN-concat.

We also tried to combine CNN-gn and CNN-glove with CNN-topic for multimodal CNN models, but their classification performance is not as good as combination of CNN-local and CNN-topic. The reason is that there are quite a few appeared words not in the CNN-gn and CNN-glove vocabulary, and embeddings for these words need to be randomly initialized. For example, more than 60% of words in the vocabulary of MEDLINE citations are not in the pre-trained CNN-glove vocabulary and need to be randomly initialized. This significantly and negatively impacts the performance of CNN-gn and CNN-glove.

4.3 Metrics

In this paper, we use F_1 score to evaluate the performance of binary classifiers and macro-averaged F_1 score for multi-class classifiers.

4.3.1 F_1 score

F_1 score is a measure of binary classification accuracy, which is the harmonic mean of precision and recall of classification results:

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{18}$$

where precision is ratio of instances which are classified as positive are correct and recall is the ratio of positive instances that are correctly classified.

4.3.2 Macro-averaged F_1 score

For multi-class classifiers, we employ macro-averaged F_1 score to evaluate their performance, which is an arithmetic average of F_1 score for each class:

$$\text{Macro-averaged } F_1 = \frac{1}{n}\sum_{i=1}^{n} F_1^i, \tag{19}$$

where n is total number of classes and F_1^i is F_1 score for ith class.

4.4 Experimental Results

In this section, we report the experimental results of baselines, state-of-the-art methods and our topic-based word embedding and multimodal CNN models. Best results are marked in bold.

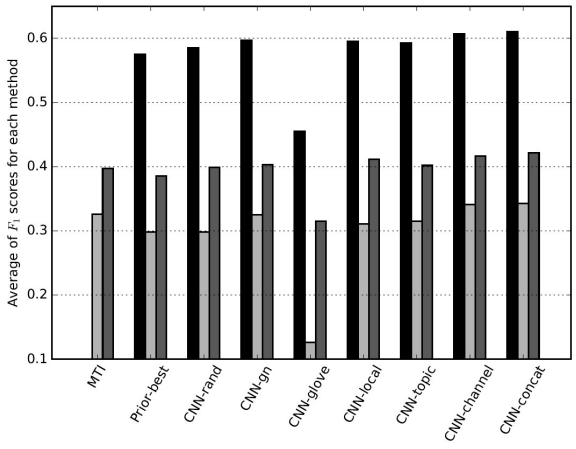

Figure 3: Macro-averaged F_1 scores of each method from the three groups. For each cluster: the black bar on the left represents performance for the check tags group; the light gray bar in the middle represents performance for the low precision group; and the dark gray bar on the right represents performance for the low recall group. Since we do not have MTI classification performance for the check tags group, its value is left blank.

4.4.1 Results of Indexing of Biomedical Articles

F_1 scores of each method over the check tags group, the low precision group and the low recall group are listed in Table 2, 3 and 4, respectively. The Positive column shows the number of positive samples for each MeSH. The results of MTI and Prior-best were reported in [26]. Although no single method outperforms all of the other approaches, the following observations can be made.

First, CNN-channel and CNN-concat give the best performance for more than 82.7% selected MeSH terms. Only for four MeSH terms: Brain, Molecular Sequence Data, Risk Assessment and Treatment Outcome, MTI system demonstrates better results than the proposed multimodal CNNs.

Second, our multimodal CNN architectures outperform baseline CNN models with a single type of word embedding.

[7] https://code.google.com/archive/p/word2vec/
[8] http://nlp.stanford.edu/projects/glove/
[9] http://word2vec.googlecode.com/svn/trunk/

Table 2: F_1 scores for check tags group.

MeSH Term	Positive	Prior-best	CNN-rand	CNN-gn	CNN-glove	CNN-local	CNN-topic	CNN-channel	CNN-concat
Adolescent	3824	0.4144	0.4321	0.4311	0.2677	0.4382	0.4104	0.4321	**0.4437**
Adult	8792	0.5700	0.6095	0.6192	0.5389	0.6159	0.6121	**0.6354**	0.6278
Aged	6151	0.5614	0.5695	0.5705	0.4378	0.5568	0.5645	**0.5841**	0.5737
Aged, 80 and over	2328	0.3227	0.321	0.3406	0.0642	0.3231	0.3316	0.3428	**0.3639**
Child, Preschool	1573	0.4954	0.4998	0.5126	0.4270	0.4944	0.4909	**0.5363**	0.5289
Female	16483	0.7517	0.7644	0.7761	0.7169	0.7761	0.7784	0.7810	**0.7840**
Humans	35967	0.9269	0.9307	0.9360	0.9113	0.9365	0.9351	**0.9366**	0.9361
Infant	1281	0.4441	0.4642	0.5032	0.1296	0.4923	0.4957	**0.5262**	0.5206
Male	15530	0.7294	0.7469	0.7477	0.6822	**0.7631**	0.7561	0.7543	0.7545
Middle Aged	8392	0.6377	0.6558	0.6665	0.6076	0.6692	0.6784	**0.6803**	0.6759
Swine	285	0.7071	0.7190	0.7332	0.6252	0.7406	0.7444	**0.7539**	0.7496
Young Adult	3807	0.3371	0.3125	0.3238	0.0499	0.3389	0.3128	0.3229	**0.3652**

Table 3: F_1 scores for low precision MeSH group.

MeSH Term	Positive	MTI	Prior-best	CNN-rand	CNN-gn	CNN-glove	CNN-local	CNN-topic	CNN-channel	CNN-concat
Age Factors	889	0.0844	0.1450	0.2150	0.2212	0.0001	0.2142	0.2233	0.2206	**0.2429**
Brain	823	**0.5201**	0.4182	0.4300	0.4596	0.1902	0.4226	0.4571	0.4697	0.4821
Cell Line	781	0.2876	0.2265	0.2277	0.2139	0.0721	0.3009	0.2389	0.2704	**0.3212**
Cells, Cultured	1079	0.3046	0.2784	0.2457	0.2936	0.0841	0.2807	0.2723	**0.3350**	0.2739
Models, Molecular	851	0.4292	0.3734	0.3769	0.4283	0.2282	0.3893	0.4138	0.4209	**0.4307**
Molecular Sequence Data	1527	**0.5495**	0.4094	0.3863	0.4035	0.2141	0.4140	0.3532	0.4211	0.4024
RNA, Messenger	628	0.4477	0.4385	0.4421	0.4397	0.3110	0.3918	0.4374	**0.4576**	0.4486
Severity of Illness Index	751	0.1824	0.1924	0.1598	0.2106	0.0372	0.1588	0.2106	0.1927	**0.2237**
Time Factors	2153	0.098	0.1393	0.091	0.1188	0.0221	0.1123	0.1179	**0.1401**	0.1364
United States	2658	0.3585	0.3655	0.4128	0.4599	0.1081	0.4213	0.4292	**0.4791**	0.4653

Table 4: F_1 scores for low recall MeSH group.

MeSH Term	Positive	MTI	Prior-best	CNN-rand	CNN-gn	CNN-glove	CNN-local	CNN-topic	CNN-channel	CNN-concat
Child	2780	0.5863	0.5723	0.6015	0.6099	0.5488	0.6102	0.6040	0.6180	**0.6192**
Follow-Up Studies	1470	0.0407	0.2300	0.2189	0.2368	0.1187	0.2247	0.2284	**0.2514**	0.2264
Reproducibility of Results	1206	0.3191	0.3138	0.2963	0.3220	0.1921	0.3261	0.3110	0.3147	**0.3274**
Retrospective Studies	2183	0.6608	0.6580	0.6647	0.6578	0.6346	0.6585	0.6617	**0.6754**	0.6589
Risk Assessment	1014	**0.2556**	0.1610	0.2063	0.1854	0.1411	0.2145	0.1979	0.2100	0.2298
Risk Factors	2365	0.4989	0.3778	0.4438	0.4510	0.3446	0.4711	0.4514	0.4654	**0.5003**
Treatment Outcome	2999	**0.4202**	0.3859	0.3635	0.3590	0.2274	0.3752	0.3592	0.3831	0.3876

This is mainly because multimodal CNNs utilize topic-based semantic word embedding as well as local content-based embedding. According to results shown in Table 2, 3 and 4, introducing topic-based semantic information improves indexing results.

Third, CNN-concat gives better results than CNN-channel for 15 terms among 29 terms and CNN-concat performs better than CNN-channel for more balanced MeSH terms. Considering there are 94,942 training samples in total, most MeSH terms are highly imbalanced. Among the 13 more balanced terms (Positive samples : Negative samples $> 0.025 : 1$), CNN-concat performs better than CNN-channel for eight MeSH terms and the average F_1 score of CNN-concat is 0.0063 higher than CNN-channel for the 13 terms.

Fourth, baseline CNN model with our proposed topic-based word embedding produces competitive results with CNN-gn and CNN-local. Word vectors used in CNN-gn and CNN-local are both trained with Skip-gram, which is the state-of-the-art word representation learning approach.

Fifth, CNN-glove demonstrates poor performance. The reason is that more than 60% of unique words in MEDLINE are not in the CNN-glove vocabulary and need to be randomly initialized. CNN-glove is pre-trained on Wikipedia2014 and Gigaword5 which do not contain many technical terms in biomedical domain. We can see that CNN-glove gives better performance on clinical text fragments and newsgroups datasets because more unique words are contained in the pre-trained vocabulary.

Sixth, the Prior-best columns refer to the best F_1 scores for traditional machine learning algorithms which give worse

performance than CNN-based models. It indicates that CNN-based approaches are more effective for indexing problems than NB, LR, SVM and SVM HL.

Finally, we summarize average of F_1 scores for each method in all of the three MeSH term groups in Fig. 3. Although there is no model outperforming all of the other models, CNN-concat demonstrates the best overall performance and CNN-channel gives very competitive average F_1 scores. Further, word embedding learned by our proposed Topic-based Skip-gram produces state-of-the-art results with baseline CNN model.

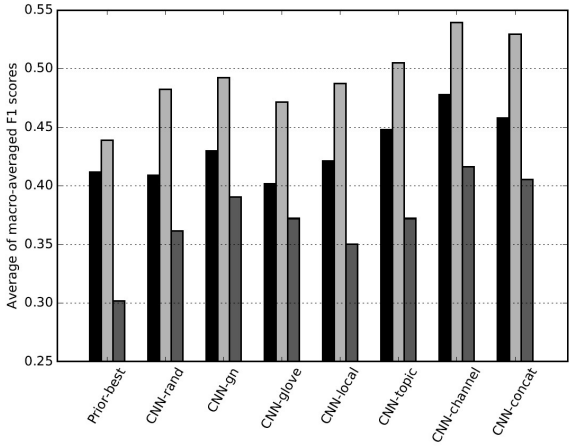

Figure 4: Macro-averaged F_1 scores for clinical text fragments. For each cluster: the black bar on the left represents the performance of annotation task A, the light gray bar in the middle represents the performance of annotation task B, and the dark gray bar on the right represents the performance of annotation task C.

4.4.2 *Results of Behavior Code Annotation of Clinical Text Fragments*

Three cases of multi-class behavior code annotation are conducted for this task: case 1, annotation over positive, negative and ambivalence, with sample ratio 1 : 0.014 : 0.150; case 2, annotation over commitment language, change talk and ambivalence, with sample ratio 0.527 : 1 : 0.094; case 3, annotation over positive commitment language, negative commitment language, positive change talk, negative change talk and ambivalence, with sample ratio 0.067 : 0.573 : 0.214 : 1 : 0.114. Clearly, all of these three data splits are highly imbalanced. For each case, we conduct 5-fold cross validation and report average macro-averaged F_1 scores for all methods over five folds. As shown in Fig. 4, CNN-channel gives the best F_1 scores among all the compared methods in the three cases and CNN-concat produces comparable results, which shows that CNN-channel performs better than CNN-concat for classification on highly imbalanced datasets. For word embeddings with baseline CNN models, CNN-topic, which is trained with proposed Topic-based Skip-gram, demonstrates better performance than other state-of-the-art word embeddings. Prior-best, which includes NB, LR and SVM in this task, is less effective than all CNN-based models.

4.4.3 *Results of Classification of Newsgroups*

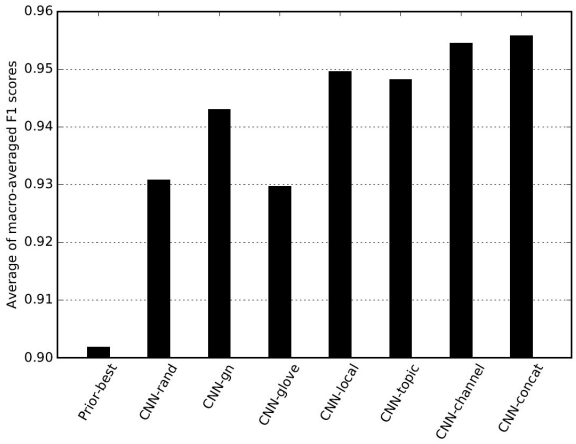

Figure 5: Macro-averaged F_1 scores for news groups.

This task is a 4-class classification problem over computers, recreation, science and politics. The sample ratio of the four categories is 1 : 0.876 : 0.811 : 0.668, which means that this dataset is nearly balanced. 5-fold cross validation is applied to the whole dataset and the average macro-averaged F_1 scores over the five folds are reported in Fig. 5. First, CNN-channel and CNN-concat outperform other baselines and state-of-the-art methods. Second, CNN-concat demonstrates better performance than CNN-channel on this balanced dataset. Third, CNN-topic with baseline CNN model produces a comparable F_1 score with other state-of-the-art word embeddings. Furthermore, CNN-based models significantly outperform non-CNN models (NB, LR and SVM).

5. CONCLUSION

In this paper, we proposed a novel framework,Topic-based Skip-gram, for learning topic-based semantic word embeddings for text classification with CNNs and achieved highly competitive results with word embeddings learned by Skip-gram. While Skip-gram focuses on context information from local word windows, the proposed Topic-based Skip-gram leverages semantic information from documents.

We also described two multimodal CNN architectures, CNN-channel and CNN-concat, which can ensemble different kinds of word embeddings. CNN-channel has a better imbalanced data resistance than CNN-concat, while CNN-concat has stronger learning ability and performs better on more balanced datasets.

Through experiments on indexing biomedical literature, annotation of clinical text fragments with behavior codes and text classification of a textual benchmark, we showed that our topic-based semantic word embeddings with multimodal CNNs outperform state-of-the-art word representations in text classification.

6. REFERENCES

[1] A. R. Aronson, J. G. Mork, C. W. Gay, S. M. Humphrey, and W. J. Rogers. The NLM indexing initiative's medical text indexer. *Medinfo*, 11(Pt 1):268–72, 2004.

[2] Y. Bengio and R. Ducharme. A neural probabilistic

language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

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

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] C. Chen, W. Buntine, N. Ding, L. Xie, and L. Du. Differential topic models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):230–242, 2015.

[6] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[8] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[9] N. Fiorini, S. Ranwez, S. Harispe, J. Montmain, and V. Ranwez. USI at BioASQ 2015: a semantic similarity-based approach for semantic indexing. In *Working Notes for the Conference and Labs of the Evaluation Forum (CLEF), Toulouse, France*, 2015.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[11] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[12] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[13] A. Kotov, M. Hasan, A. Carcone, M. Dong, S. Naar-King, and K. BroganHartlieb. Interpretable probabilistic latent variable models for automatic annotation of clinical text. In *AMIA Annual Symposium Proceedings*, volume 2015, page 785. American Medical Informatics Association, 2015.

[14] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273, 2015.

[15] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, pages 331–339, 1995.

[16] K. Liu, S. Peng, J. Wu, C. Zhai, H. Mamitsuka, and S. Zhu. MeSHLabeler: improving the accuracy of large-scale MeSH indexing by integrating diverse evidence. *Bioinformatics*, 31(12):i339–i347, 2015.

[17] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

[18] Y. Mao, C.-H. Wei, and Z. Lu. Ncbi at the 2014 bioasq challenge task: Large-scale biomedical semantic indexing and question answering. In *CLEF (Working Notes)*, pages 1319–1327, 2014.

[19] J. D. Mcauliffe and D. M. Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013.

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

[22] Y. Papanikolaou, G. Tsoumakas, M. Laliotis, N. Markantonatos, and I. Vlahavas. AUTH-Atypon at BioASQ 3: Large-scale semantic indexing in biomedicine. In *Working Notes for the Conference and Labs of the Evaluation Forum (CLEF), Toulouse, France*, 2015.

[23] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[24] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[25] N. Rasiwasia and N. Vasconcelos. Latent dirichlet allocation models for image classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2665–2679, 2013.

[26] A. Rios and R. Kavuluru. Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '15, pages 258–267, New York, NY, USA, 2015. ACM.

[27] G. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 627–634. ACM, 2008.

[28] L. Yeganova, D. C. Comeau, W. Kim, and W. J. Wilbur. Text mining techniques for leveraging positively labeled data. In *Proceedings of BioNLP 2011 Workshop*, pages 155–163. Association for Computational Linguistics, 2011.

[29] A. J. J. Yepes, J. G. Mork, D. Demner-Fushman, and A. R. Aronson. Comparison and combination of several mesh indexing approaches. In *AMIA annual symposium proceedings*, volume 2013, page 709. American Medical Informatics Association, 2013.

[30] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[31] X. Zhang and Y. LeCun. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015.