PSEUDO-SUPERVISED APPROACH FOR TEXT CLUSTERING BASED ON CONSENSUS ANALYSIS

Peixin Chen, Wu Guo, Lirong Dai, Zhenhua Ling

National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China

ABSTRACT

In recent years, neural networks (NN) have achieved remarkable performance improvement in text classification due to their powerful ability to encode discriminative features by incorporating label information into model training. Inspired by the success of NN in text classification, we propose a pseudo-supervised neural network approach for text clustering. The neural network is trained in a supervised fashion with pseudo-labels, which are provided by the cluster labels of pre-clustering on unsupervised document representations. To enhance the quality of pseudo-labels, a consensus analysis is employed to select training samples for the neural network. The experimental results demonstrate that the proposed approach can improve the clustering performance significantly.

Index Terms— Text clustering, pseudo-supervised, consensus analysis

1. INTRODUCTION

Text clustering is a fundamental task in text mining and information retrieval that aims to group similar documents into clusters [1]. In general, the documents are first represented as fixed-dimensional feature vectors, and clustering algorithms, such as K-means and hierarchical clustering algorithms, are subsequently performed to partition the documents into groups. The most naïve but common approach for document representation is bag-of-words, such as the Term Frequency−Inverse Document Frequency (TF−IDF) [2]. Though simple and feasible, TF−IDF does not discover the latent semantic structure information and suffers from data sparsity caused by high dimensionality. To overcome the drawbacks of bag-of-words, researchers have developed a series of dimensionality reduction techniques that uses term co-occurrence statistics to capture the latent semantic structure of documents and represent them as low-dimensional vectors. The typical methods include Latent Semantic Analysis (LSA) [3], Probabilistic Latent Semantic Analysis (PLSA) [4] and Latent Dirichlet Allocation (LDA) [5]. An alternative approach is to use neural network-based topic models, such as the Replicated Softmax [6], Neural Autoregressive Density Estimators (DocNADE) [7] and the Over-Replicated Softmax [8]. Although the abovementioned approaches are aimed at capturing salient statistical patterns in the co-occurrence of words within documents, they do not take advantage of recent advances in distributed word representations that can capture semantically meaningful regularities between words [9][10]. To encode distributed semantic features of documents based on word embeddings, Le and Mikolov proposed Paragraph Vectors, which can predict the words in each document [11]. In [12], Moody proposed lda2vec that embeds both word vectors and document-level mixtures of topic vectors into the same space and trains them simultaneously.

All of the aforementioned models are unsupervised. Compared with the unsupervised models, the supervised ones usually generate more discriminative hidden topic features. The distributed representations based on Convolutional Neural Networks (CNN) are able to learn n-gram features through multiple filters [13][14], and the ones based on Recurrent Neural Networks (RNN) perform excellently at learning sequential information from word sequences [15][16]. To capture more contextual information than the conventional fixed-size filters in CNN, Lai et al. proposed the Recurrent Convolutional Network (RCNN) [17], which captures the contexts around each word with bidirectional recurrent structure and later constructs the representation of text with a convolutional architecture.

The CNN, RNN and RCNN mentioned above are primarily designed for specific tasks, such as text classification, where predefined labels are provided for modeling training. In contrast, text clustering, a typical unsupervised task, has no predefined labels available during training. To benefit from the discriminative ability of these supervised neural network models, we propose a pseudo-supervised training approach in this paper. In contrast to conventional supervised training, the supervision information provided for NN training is from pseudo-labels, which are generated by pre-clustering on unsupervised document representations. To suppress noise in the pseudo-labels, we propose a consensus analysis approach, which generates more appropriate pseudo-labels based on two pre-clusterings. The documents assigned to consistent cluster labels in the two pre-clusterings are selected as the training samples for NN, while the others are regarded as noise samples. The experiments are carried out on the Fisher corpus,

and the results have demonstrated the effectiveness of the proposed method.

2. BACKGROUND

Before going into the details of our framework, we provide some background on four popular unsupervised document modeling methods that are relevant to our work.

LSA: Let $\mathbf{X}$ be the term-document matrix where elemen-t (i, j) describes the weight of term i in document j. LSA employs singular value decomposition (SVD) on $\mathbf{X}$ to find the latent semantic structure:

$$\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{1}$$

where $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, ..., \sigma_l)$ are the singular values. S-electing the top *r* largest singular values, LSA uses the corresponding singular vectors from $\mathbf{U}$ to embed the document features into *r*-dimensional vectors.

LDA: LDA defines a generative data model that represents each document *m* as a multinomial distribution θ_m over latent topics, where each topic *k* is a multinomial distribution φ_k over words. Both multinomial distributions satisfy Dirichlet prior distributions. Following the procedure of Gibbs sampling [18], the value of θ_m that is the low-dimensional representation of document *m* can be inferred.

DocNADE: For a sequence of words, DocNADE models the joint probability of words through the probability chain rule:

$$p(\mathbf{v}) = \prod_{i=1}^{n} p(v_i|\mathbf{v}_{<i}) \tag{2}$$

where $\mathbf{v}$= [v_1,v_2,...,v_n] are the observation sequences and $\mathbf{v}_{<i}$ is the subvector [v_1,v_2,...,v_{i-1}]. DocNADE assumes that each conditional probability $p(v_i|\mathbf{v}_{<i})$ can be modeled by a feedforward neural network, where the probability of the i^{th} word v_i is based on a position dependent hidden layer $\mathbf{h}_i(\mathbf{v}_{<i})$ that extracts a representation out of all previous words $\mathbf{v}_{<i}$. Once the model is trained, a latent representation can be extracted from a new document $\mathbf{v}^*$ by computing the value of its hidden layer $\mathbf{h}_i(\mathbf{v}^*)$.

Lda2vec: The lda2vec model stems from Skip-gram [9], which uses each pivot word to predict target words within a certain range before and after the pivot word. For each pair of pivot word $\mathbf{w}_j$ and target word $\mathbf{w}_i$, Skip-gram samples *n* negative samples and calculates the likelihood as follows:

$$L_{ij} = \log \sigma(\mathbf{w}_j \cdot \mathbf{w}_i) + \sum_{l=1}^{n} \log \sigma(-\mathbf{w}_j \cdot \mathbf{w}_l) \tag{3}$$

where σ refers to the sigmoid function; in lda2vec, the pivot word $\mathbf{w}_j$ in L_{ij} is replaced by a context vector $\mathbf{c}_j$, the sum of $\mathbf{w}_j$ and document vector $\mathbf{d}_j$. The document vector $\mathbf{d}_j$, shared by all pivot-target pairs in a document, is a mixture of a set of latent topic vectors $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_k$. In addition, the document weights over latent topics are optimized by a Dirichlet likelihood.

3. PSEUDO-SUPERVISED APPROACH

The overall architecture of the proposed approach is depicted in Fig.1. Given the raw document collection $\mathbb{D}$, the objective is to group these documents into clusters $\mathbb{C}_{\text{final}}$. Overall, the proposed framework mainly consist of three parts: the generation of consensus samples, pseudo-supervised training of RCNN with consensus samples, and clustering based on distributed semantic features.

3.1. Consensus Samples Generation

The generation of consensus samples involves two main steps. The first step is to align the cluster labels of two pre-clusterings. Secondly, the documents with consistent cluster labels in the two pre-clusterings are chosen as consensus samples, while the others are regarded as noise samples. In the first step, two pre-clusterings $\mathbb{C}_1$ and $\mathbb{C}_2$ are obtained by performing the clustering algorithm on feature sets $\mathbb{F}_1$ and $\mathbb{F}_2$, respectively, which are learned by two different unsupervised models chosen from among those described in section 2. The number of clusters *r* is a preset value. Next, a mapping function is employed to map each cluster label in $\mathbb{C}_2$ to the best-matched cluster label in $\mathbb{C}_1$. This computation is equivalent to the problem of finding the maximum weighted bipartite matching [19] in a bipartite graph $G = (U, V, E)$, where the vertex sets U and V contain the cluster labels of $\mathbb{C}_1$ and $\mathbb{C}_2$, respectively, each edge in E connects a vertex in V to one in U, and the weight of each edge is the number of documents shared by both vertexes. The best mapping can be achieved using the Kuhn-Munkres algorithm [20]. Since the number of clusters provided by $\mathbb{C}_1$ and $\mathbb{C}_2$ are equal, the Kuhn-Munkres algorithm will match each cluster in $\mathbb{C}_2$ to a unique cluster in $\mathbb{C}_1$.

In the second step, the documents with consistent cluster labels in $\mathbb{C}_1$ and $\mathbb{C}_2$ are selected. Given a document d_i, and letting α_i and β_i be the cluster labels provided by $\mathbb{C}_1$ and $\mathbb{C}_2$,

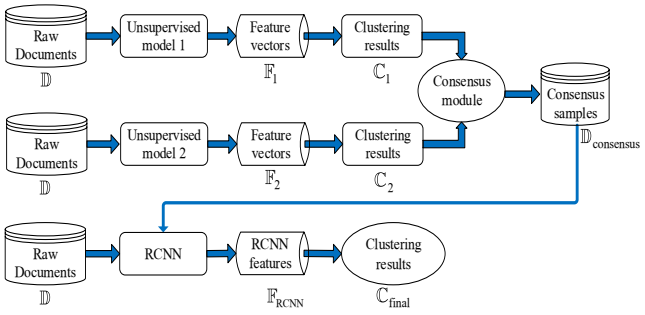


Fig. 1. The pseudo-supervised approach for text clustering.

respectively, we generate the consensus samples as follows:

$$d_i \begin{cases} \in \mathbb{D}_{\text{consensus}}, & \text{if } \alpha_i = map(\beta_i) \\ \notin \mathbb{D}_{\text{consensus}}, & \text{if } \alpha_i \neq map(\beta_i) \end{cases} \quad (4)$$

where $map(\beta_i)$ is the mapping function described above. In this manner, we can obtain the document set $\mathbb{D}_{\text{consensus}}$ whose cluster labels have less noise and high reliability. The documents with inconsistent cluster labels in $\mathbb{C}_1$ and $\mathbb{C}_2$ are excluded from $\mathbb{D}_{\text{consensus}}$.

3.2. RCNN Training

The RCNN is trained on the consensus samples $\mathbb{D}_{\text{consensus}}$, using the cluster labels as pseudo-labels for training. Based on the work of Lai et al. and Wen et al. [21], the RCNN in our approach mainly consists of five parts: embedding layer, recurrent layer, convolutional layer, pooling layer and regression layer. In the embedding layer, the input words are mapped to a matrix $\mathbf{x} \in \mathbb{R}^{d \times n}$, whose i^{th} column $\mathbf{x}_i \in \mathbb{R}^d$ corresponds to the embedding of the i^{th} word. In the recurrent layer, the word embeddings are fed into the bidirectional LSTM [22] architecture to generate the hidden states. We concatenate word embedding $\mathbf{x}_t$ with LSTM hidden state $(\mathbf{h}_t^{\leftarrow}, \mathbf{h}_t^{\rightarrow})$ at time step t to obtain the contextual representation of the t^{th} word:

$$\tilde{\mathbf{x}}_t = \mathbf{h}_t^{\leftarrow} \oplus \mathbf{x}_t \oplus \mathbf{h}_t^{\rightarrow} \quad (5)$$

where $\oplus$ is the concatenation operation. In the convolutional layer, the convolution operation is employed to generate the representations $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{n-w+1}$ as follows:

$$\mathbf{p}_i = \text{ReLU}(\mathbf{W} * \tilde{\mathbf{x}}_{i:i+w-1} + \mathbf{b}) \quad (6)$$

where $*$ refers to the convolution operation, $\mathbf{W}$ is the convolution weights, $\mathbf{b}$ is the bias, and w is the filter width. ReLU [23] is chosen as the activation function. The filter width w is fixed at 1 in this study, since the bidirectional LSTM has already obtained the context information around each word. To preserve more important information, we adopt a *k*-max pooling operation in the pooling layer, which extracts the top k maximum features instead of only one from each feature map. As a result, we can obtain a feature vector $\mathbf{s} \in \mathbb{R}^l$, where $l = k \times d_1$ (d_1 denotes the dimensionality of $\mathbf{p}_i$):

$$\mathbf{s} = k\text{-max}(\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{n-w+1}) \quad (7)$$

The last layer of RCNN is a logistic regression function:

$$\mathbf{y} = \text{softmax}(\mathbf{W}_o \mathbf{s} + \mathbf{b}_o) \quad (8)$$

where $\mathbf{W}_o$ is the weight matrix and $\mathbf{b}_o$ is the bias. The output $\mathbf{y} \in \mathbb{R}^r$ is the probability distribution over the *r* clusters.

3.3. Clustering

With the given document collection $\mathbb{D}$, we utilize the trained RCNN to obtain the semantic representations $\mathbf{s}$, which are the output features of the pooling layer. Subsequently, a traditional clustering algorithm is performed on $\mathbf{s}$ to categorize the documents into *r* clusters. We adopt the agglomerative hierarchical clustering (AHC) algorithm [24]. The AHC algorithm can obtain slightly better performance than K-means algorithm on this dataset. Besides, AHC is not affected by initial cluster centers as K-means is, which makes a fairer comparison between different systems. The linkage criterion adopted in AHC is Ward, which minimizes the variance of the clusters being merged.

4. EXPERIMENTS

4.1. Dataset

The experiments were conducted on the Fisher English corpus, which is released by LDC (Linguistic Data Consortium) [25]. The Fisher English corpus consists of 11699 recorded telephone conversations, with corresponding text transcriptions. This corpus contains 40 topics in total, and each conversation is assigned to a specific topic. We employed the text transcriptions as the experimental documents for text clustering, with 117∼560 documents for each topic. The text preprocessing included tokenization and stop-words removal. We also removed the words whose document frequency (DF) is below 5. The average length of the documents is approximately 400 words after text preprocessing.

4.2. Hyperparameter Settings

In this section, we describe the system configurations, namely, the configurations of the baseline systems described in section 2 and the proposed pseudo-supervised approach.

In LSA, we retain the top 100 singular values to form the new subspace. In LDA, we set the number of latent topics to 200. In DocNADE, the hidden layer size is set to 200, and the sigmoid is chosen as the activation function. In lda2vec, the dimensionality of embedding vectors is set to 300, and the number of latent topics is set to 200.

The RCNN model is implemented by Tensorflow [26], and trained by Adam [27] with a learning rate 0.001. We initialize the word vectors with 300-dimensional pre-trained word2vec vectors. All the word vectors are fine-tuned along with other model parameters during training. The dimensionality of LSTM hidden state $\mathbf{h}_t^{\leftarrow}$ and $\mathbf{h}_t^{\rightarrow}$ are both set to 256. In the convolutional layer, 256 filters are used. The value of *k* for *k*-max pooling is 3. To prevent co-adaptation, a dropout rate of 0.5 is employed on the input of LSTM and before the logistic regression layer.

4.3. Results and analysis

The accuracy (ACC) and the normalized mutual information (NMI) [28] metrics are used to measure the clustering performance, which compare the clustering results with the topic

labels provided by LDC. The evaluations were conducted for the cluster numbers ranging from 30 to 50 with a step size of 10. Table 1 lists the clustering results on five unsupervised document features respectively, showing that DocNADE performs better than other document modeling methods in most cases. By contrast, the relatively poor performance of TF-IDF demonstrates the shortcomings of bag-of-words.

In Table 2, we report the clustering results of several different pseudo-supervised frameworks. The first four frameworks are based on consensus analysis, where two different unsupervised models are employed to obtain the consensus samples. The last two frameworks employ pseudo-labels for RCNN that are generated by pre-clustering on a single unsupervised model without consensus analysis. Table 2 shows that the pseudo-supervised RCNN based on consensus analysis yields noticeably better performance than the baseline systems with respect to both ACC and NMI. In contrast, the last two frameworks bring about no improvement over baseline models, despite more training samples for RCNN than the first four frameworks. To more clearly compare pseudo-supervised approaches to baselines, we randomly chose five topics and report the two-dimensional t-SNE [29] embedding of document features in Fig.2. This figure shows that the consensus-based pseudo-supervised semantic features have more clear-cut margins among different topics and a higher degree of intra-cluster similarity than unsupervised features.

The results indicate that the proposed pseudo-supervised approach based on consensus analysis is an effective approach to obtain semantic features for text clustering. The performance improvement benefits from the distinguishing ability of RCNN, as well as the pseudo-labels provided by the consensus samples. Compared with the pseudo-labels provided by an individual unsupervised model, the pseudo-labels provided by the consensus analysis are more stable and reliable in quality, since the documents with noisy cluster labels are excluded from training samples. Furthermore, the RCNN, which is a kind of discriminative model, can learn more distinguishing features of the clusters, since it incorporates the cluster label information into the training objective. As a result, the documents potentially belonging to the same category will be closer to each other in the semantic space constructed by the pseudo-supervised RCNN.

5. CONCLUSIONS

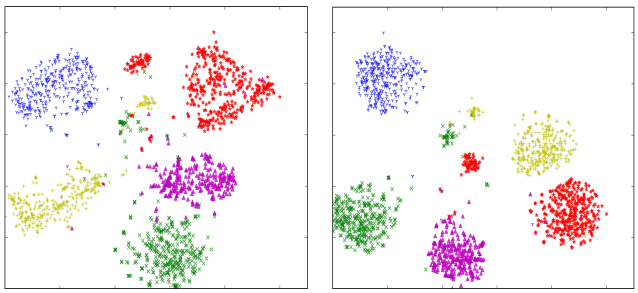
This paper introduces a pseudo-supervised approach for text clustering based on the distributed semantic features learned by the RCNN, which is trained with pseudo-labels. Experiments conducted on the Fisher English corpus demonstrate the effectiveness of this approach, which outperforms state-of-the-art systems for a variety of cluster numbers. Our approach is a flexible framework in which different unsupervised models can be employed; furthermore, the RCNN can be replaced by other neural network architectures as needed.

Table 1. ACC(%)/NMI(%) of baseline systems

	30	40	50
LSA	69.08/78.29	80.43/81.29	76.39/80.14
LDA	72.34/78.95	**81.42**/81.97	74.85/80.13
DocNADE	**74.20/80.93**	80.70/**82.47**	**77.28/81.40**
lda2vec	72.74/79.77	80.37/80.91	74.95/79.73
TF-IDF	64.52/71.62	73.05/74.34	69.53/74.23

Table 2. ACC(%)/NMI(%) of pseudo-supervised frameworks

	30	40	50
LSA-LDA	76.39/84.14	89.19/89.65	84.80/**89.09**
LSA-lda2vec	76.61/85.12	**90.75/91.17**	83.20/88.81
LDA-DocNADE	76.66/84.37	89.64/90.09	**84.84**/88.18
DocNADE-lda2vec	**77.50/86.06**	87.43/89.36	84.24/88.39
LDA	72.33/78.96	81.76/82.38	75.18/80.48
DocNADE	74.28/81.07	80.81/82.65	77.41/81.73



(a) (b)

Fig. 2. The two-dimensional t-SNE embedding of document features under cluster number 40, where (a) shows the features inferred by LDA, (b) shows the pseudo-supervised features based on LSA-LDA consensus analysis.

Future research will address the selection of more effective training samples and the improvement of pseudo-labels.

6. ACKNOWLEDGEMENTS

This work was partially funded by the National Key Research and Development Program of China (Grant No. 2016YFB1001303). Besides, we would like to thank the anonymous reviewers for their valuable comments and suggestions.

7. REFERENCES

[1] W Bruce Croft, Donald Metzler, and Trevor Strohman, *Search engines: Information retrieval in practice*, vol. 283, Addison-Wesley Reading, 2010.

[2] Gerard Salton and Christopher Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[3] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391, 1990.

[4] Thomas Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, no. 1, pp. 177–196, 2001.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[6] Geoffrey E Hinton and Ruslan R Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in neural information processing systems*, 2009, pp. 1607–1614.

[7] Hugo Larochelle and Stanislas Lauly, "A neural autoregressive topic model," in *Advances in Neural Information Processing Systems*, 2012, pp. 2708–2716.

[8] Nitish Srivastava, Ruslan R Salakhutdinov, and Geoffrey E Hinton, "Modeling documents with deep boltzmann machines," *Computer Science*, 2013.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[11] Quoc Le and Tomas Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014.

[12] Christopher E Moody, "Mixing dirichlet topic models and word embeddings to make lda2vec," *arXiv preprint arXiv:1605.02019*, 2016.

[13] Yoon Kim, "Convolutional neural networks for sentence classification," *EMNLP*, 2014.

[14] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom, "A convolutional neural network for modelling sentences," *ACL*, 2014.

[15] Duyu Tang, Bing Qin, and Ting Liu, "Document modeling with gated recurrent neural network for sentiment classification.," in *EMNLP*, 2015, pp. 1422–1432.

[16] Ji Young Lee and Franck Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *NAACL*, 2016.

[17] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao, "Recurrent convolutional neural networks for text classification.," in *AAAI*, 2015, vol. 333, pp. 2267–2273.

[18] Gregor Heinrich, "Parameter estimation for text analysis," *Technical Report*, 2005.

[19] Douglas Brent West et al., *Introduction to graph theory*, vol. 2, Prentice hall Upper Saddle River, 2001.

[20] László Lovász and Michael D Plummer, *Matching theory*, vol. 367, American Mathematical Soc., 2009.

[21] Ying Wen, Weinan Zhang, Rui Luo, and Jun Wang, "Learning text representation using recurrent convolutional neural network with highway layers," *arXiv preprint arXiv:1606.06905*, 2016.

[22] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, 1997.

[23] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[24] Lior Rokach and Oded Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, pp. 321–352. Springer, 2005.

[25] Christopher Cieri, David Miller, and Kevin Walker, "The fisher corpus: a resource for the next generations of speech-to-text.," in *LREC*, 2004, vol. 4, pp. 69–71.

[26] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[27] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao, "Self-taught convolutional neural networks for short text clustering," *Neural Networks*, vol. 88, pp. 22–31, 2017.

[29] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.