A Comprehensive Survey on Short Text Clustering

Nanda P Sadasivan
M.tech Student, Computer Science and Engineering
NSS College of Engineering
Palakkad,Kerala,India
nandasivan18@gmail.com

S Sindhu
Associate Professor,Computer Science and Engineering
NSS College of Engineering
Palakkad,Kerala,India
sindhu.nss@gmail.com

***Abstract*—The wide acceptance of social platforms has changed the way we access and share information. This led to the generation of huge amount of data everyday. Most of these data consists of short texts. Short texts are sentences which have less than 245 words. Social network chatting, microblogs and different types of mobile based text communications contribute to the generation of short text. Short text clustering is used in various applications like user profiling, recommendation etc. Hence there is a need for clustering the short text. But short text processing is very difficult due to the sparseness of its text representation.**

***Index Terms*—Short text clustering, Neural Networks, Deep learning.**

I. Introduction

The popularity of social media led to the generation of a large amount of short texts. These texts are unstructured text and cannot be simply processed and perceived by computer. Since the applications like user profiling, recommendation systems etc. need these short text texts, its very important to process it carefully. It would be very helpful when these texts are clustered. Clustering is an unsupervised technique in which the given data is grouped, in such a way that the objects in the same group(called a cluster) are similar to each other than those in other groups. Sparseness of the text representation is the most challenging problem when handling the short text,i.e., most of the words occur only once in each short text. As a result, the term frequency-inverse document frequency(*tf-idf*) measure cannot work well in the terms of short text representation.

Recently there has been an increase in interest towards clustering short text because it is used in many NLP applications. According to the application, a variety of short text could be defined mainly in terms of their length(e.g. sentence, paragraphs) and type(e.g.,scientific papers, newspapers). This is another challenge for clustering the short texts. In general finding a clustering method which is able to cluster short text is difficult. Many researchers work on the clustering of short text and this paper is a comprehensive study on those techniques.

Some researchers work on enriching and expanding the context of data from Wikipedia. Some others make sophisticated models to cluster the short texts. Dirichlet multinomial mixture model-based approach is one such approach for short text clustering. Another method used to cluster the short text is directly trained based on Bag-of-Words(BoW), which are shallow structures. Also various dimension reduction methods such as Latent Semantic Analysis(LSA), Laplacian Eigenmaps(LE), and Locality Preserving Indexing(LPI) are studied under this context. The genetic algorithm like Particle Swarm Optimization is also studied under the context of short text clustering.

The main area which is handled with the short text clustering is the deep learning techniques used in the clustering of short texts. The survey on short text clustering deals with the Deep Neural Networks such as Autoencoders, Word embeddings, Word2Vec, Recursive and Recurrent Neural Networks, Convolutional Neural Networks etc.

II. Short text clustering

The main problem which is to be dealt with short text clustering is the data sparsity. i.e., most of the words occur only once in the short text.

A. NLP Techniques

A number of methods are used to overcome the sparseness of short text representation. One way is to expand and enrich the context of data. Previously Wikipedia is used for this purpose [22]. When a short text is given, two query strings are created from the text. Then these query strings are used to retrieve the top matching Wikipedia articles from the Lucene index(http://lucene.apache.org/). Now the titles of these retrieved Wikipedia articles serves as the additional features of the text for clustering. These titles are referred as Wikipedia concepts. These additional features now reduce the sparseness of the short text and helps in the clustering process.

Another NLP method in short text clustering incorporates semantic knowledge from an ontology into text clustering [15]. An ontology can be used to greatly reduce the number of features needed for clustering. In this method nouns are used as cluster features. WordNet noun database is used to identify the nouns in the text. The most important part is to identify the synonymous and polysemous nouns. That is different nouns having same meaning and single noun having multiple

Identify applicable funding agency here. If none, delete this.

meanings. Polysemy and synonymy affect the similarity value computed between the documents. Hence these nouns are known to be the core semantic features. These features are not only useful for clustering, but they may represent the main theme of the document. Once the core features are identified spherical k-means is used to produce the base clusters.

B. Dimension reduction methods

The above mentioned methods need solid NlP knowledge and use high-dimensional representation which is a waste of memory and computation time. So another way used is to map the original features into reduced space. Latent Semantic Analysis(LSA), Laplacian Eigenmaps(LE), and Locality Preserving Indexing(LPI) are used for this purpose.

When using the *tf-idf* scheme a collection of documents can be represented as a $V \times$D matrix. Here the rows denote the terms or words and the columns contain the *tf-idf* value for the chosen terms. But this structure reveals little about the semantic structure of the documents. Latent Semantic Analysis(LSA) addressed the limitations of the *tf-idf* scheme by performing singular value decomposition(SVD) on the $V \times$D matrix [29]. LSA have been used in the documents which contains a few hundred words, i.e., short texts.

Since most of the NLP techniques for feature representation makes high-dimensional feature space, another way for reducing this is the method of Laplacian Eigenmaps [26]. In this approach a graph is build, which incorporates neighborhood information of the data set. Using the concept of Laplacian of the graph, a low-dimensional representation of the data set is computed. This representation preserves local neighborhood information in a certain sense.

Locality Preserving Indexing is a form of dimensional reduction which can be used in the information processing problem [25]. These are linear projective maps that arise by solving a variational problem that optimally preserves the neighborhood structure of the data set. The locality preserving quality of LPP is used in the information retrieval from text documents under a vector space model. In this method for retrieval purpose, one have to do a nearest neighbor search in the low dimensional space. Here in LPI the problem is to make a transformation matrix that maps the points from a high-dimensional space to the low-dimensional space.

Some method choses both feature enrichment and dimensional reduction. One such method enrich the text features by employing machine translation and reduce the original features simultaneously through matrix factorization techniques [13].

C. Model for clustering short text

Many sophisticated models are used to cluster short text. Dirichlet Multinomial Mixture(DMM) model is one among them [4]. DMM is probabilistic generative model for documents. It is based on two assumption. First assumption is that the documents are generated by a mixture model. A mixture model is a probabilistic model for representing the presence of subgroups within a group. The second assumption is that, there is a one-to-one correspondance between mixture components and clusters. When the documents are clustered, DMM first selects a cluster according to the weights of clusters. Then a document d is clustered by the selected cluster using probability distribution.

D. PSO techniques

This section presents the survey on PSO techniques for short text clustering. The Particle Swarm Optimization(PSO) is a genetic algorithm in which a system is initialized with a population of random solutions. On the other hand, It's in contrast to the genetic formula. Every possible solutions which are known to be particles, have irregular speed and these particles are flown into the hyperspace. The fitness of each individual particle is evaluated. Then the velocities are modified based on previous best and global best. The velocity is updated and the new position of the particle is computed until a stopping criterion is met.

Several PSO algorithms were used to cluster the short text. One such algorithm is CLUDIPSO [11]. CLUDIPSO(CLUstering with a DIscrete Particle Swarm OPtimization), is a discrete version of the PSO algorithm. In CLUDIPSO algorithm each particle represents a valid cluster. Each position in the particle corresponds to the documents to be clustered. The integer value stored i this position specifies the group that the document belongs to. The position and velocity of the particles are updated until the best cluster is obtained. The formulas for updating the position and velocity is somewhat similar to the PSO algorithm.

Another PSO approach is proposed for the clustering problem. This presents effective, robust, comparatively efficient easy-to-tune PSO algorithm and is applicable when the number of clusters is either known or unknown [18]. The particles continuously moves to search for better solutions, and movement depends on various topologies. Here it follows the so-called ”gbest neighborhood topology”. This method randomly generates initial particles. Then each object is assigned to the nearest cluster of the particle. Then the cluster centers of particle is updated and the fitness function for the particle is calculated. The particle memorizes the initial position as the best. Next the best particle of the swarm is found. Next is moving a particle. According to this topology, a particle move towards its best position and towards the best neighbor in the swarm. If the current position is better than the memorized position, then the particle memorizes current position as the best. Then at last each particle will be in their best position.

A study on short text clustering proposes a novel discrete PSO-based algorithm which incorporates the CLUDIPSO algorithm and another algorithm called

CLUCOPSO(CLUstering with a COntinuous PSO) [20]. This is a continuous PSO based algorithm similar to that of CLUDIPSO. It reduces high dimensionality of the search space. In this method unsupervised measures are used to evaluate the results of the clustering algorithm.

A better quality document cluster is acquired through an intelligent hybrid PSO algorithm [12]. This approach uses a Fuzzy C-Means(FCM) and K-Means hybridised with Particle Swarm Optimization. Here the PSO algorithm is applied to the documents to get the cluster. At the initial stage cluster centroid vectors are choosen. For each particle each document vector is assigned to the document set closest to the centroid vector. Then the fitness value is calculated. Next solutions are generated by updating the velocity and particle position. This is repeated until a good solution is obtained.

III. Deep neural networks

Deep learning architecture such as Deep Neural Networks(DNN) can be applied in the field of Natural Language Processing. Many researchers have used deep learning to learn features.

One of the approaches uses DAE to learn text representation [24]. Autoencoder is a neural network architecture. It learns to encode a variable-length input sequence into a fixed-length vector representation and then it decode it into a variable length sequence, which is trained to resemble the initial input. The encoder reads each symbols of input sequence as a word vector and uses it to update the hidden state. When all the symbols of the input are passed, the hidden state of that sequence is formed. The decoder update a hidden state using the previously outputted word. The decoder continues to unfold the hidden state until it output the entire sequence.

To improve DNN performance on NLP tasks, the researchers propose to use external corpus to learn a distributed representation for each word, called word embedding. Word embeddings is a neural language model that could be trained over billions of words [19]. This neural language model is discriminative and non-probabilistic. This model is used to map words or phrases from the vocabulary to vectors of real numbers. Neural networks is used to generate this mapping.

The Skip-gram and continuous bag-of-word models of Word2Vec are efficient approach for the estimation of word representations in vector space [10]. Word2Vec is a group of models which tries to represent each word in a large text as a vector in the vector space. The Skip-gram method takes every word in the text corpora and it also take one-by-one the words that surround it within a 'window'. This is then feded to a neural network that after training will predict the probability for each word to actually appear in the window around the focus word. Continuous bag-of-words model is yet another model of Word2Vec. The input to this model are the preceding and following words of the current word and the output of the neural network will be the current word. Thus it is predicting the context when a word is given.

GloVe [5] is another model for word representation which captures the global corpus statistics. The primary source of information available to all unsupervised methods for learning word representations is the statistics of word occurrences in a corpus. But the word vectors might not represent the meanings generated from these statistics. So a new model for word representation GloVe, means Global Vectors is constructed, which directly captures the global corpus statistics. Here the relationship between two words can be examined by studying the ratio of their co-occurrence with various probe words. i.e., the appropriate way for word vector learning should be with the ratios of co-occurrence probabilities rather than the probabilities themselves.

The Word2Vec model is extended to represent the vectors of sentences. This is by predicting words in the sentence and is named as Paragraph vector(Para2vec) [6]. Paragraph Vector(Para2Vec), is an unsupervised framework which learns continuous distributed vector representations for short text pieces. In paragraph vector framework, every paragraph is mapped to a unique word vector. The obtained paragraph vector and word vector are averaged or concatenated to predict the the next word in a context.

RecNN and RNN are neural networks which utilize word embeddings to capture meaningful syntactic and semantic regularities. Recursive Neural Network(RecNN) [9], [16] compute compositional vector representations for phrases of variable length and syntactic type. These representations are used as features to classify each phrase. In Recurrent Neural Network(RNN) [17], the vector of a sentence is formed by concatenating the word vectors that represent current word while using 1 of N coding. The training of the network is by using the standard backpropagation and the network contains input, hidden and output layers.

Two sophisticted recurrent hidden units of RNN are Long Short-Term Memory(LSTM) and Gated Recurrent Unit(GRU). LSTM is a simple recurrent neural network which are the building component of recurrent neural network [28]. The main components of LSTM block are: a cell, an input gate, an output gate, and a forget gate. The word memory in LSTM stands for remembering the values over arbitrary time intervals. The cell is responsible for remembering these values. All the three gates are artificial neuron, as in the case of a feedforward neural network. Each of these gates uses activation function to compute the activation for weighted sum. The term LSTM refers to the fact that short-term memory which can last for long period of time. Gated Recurrent Units(GRUs) [7] are a gating mechanism in recurrent neural networks. Their modelling is similar to that of long short-term memory but have fewer parameters than

LSTM, because they lack an output gate.

Convolutional Neural Networks(CNNs) [1] are another type of Deep Neural Networks which are used to learn non-biased implicit features which has been successfully exploited for many supervised NLP learning tasks. In Convolutional Neural Network model, a raw text vector is feed as the input. The CNN consists of several hidden layers. Each layer has functions to do with the input such as wide one-dimensional convolution, folding and dynamic k-max pooling. The output of these layers are the deep feature representation of the text. Many CNN based variants such as Dynamic Convolution Neural Network(DCNN), Gated Recursive Convolutional Neural Network(grConv) and Self-Adaptive Hierarchical Sentence model(AdaSent) are the different Deep Neural Network techniques used in the area of feature learning.

In Dynamic Convolution Neural Network(DCNN) [8] the matrix representation of the sentence in fed into convolutional layer with multiple filter widths and feature maps. The output the convolutional layer passes to the pooling layer and fully connected layer to obtain the text representation.

A good approach to deal with the variable-length sequence is Gated Recursive Convolutional Neural Network(grConv) [31]. This is a binary convolutional neural network whose weights are recursively applied to the input sequence until it outputs a single fixed-length vector. Instead of maintaining a fixed length continuous vectorial representation, Self-Adaptive Hierarchical Sentence model(AdaSent) [32] model forms a multi-scale hierarchical representation. AdaSent is inspired from the gated recursive convolutional neural network (grConv) in the sense that the information flow forms a pyramid with a directed acyclic graph structure where local words are gradually composed to form intermediate representations of phrases.

Inferences

Different techniques used for clustering the short text are discussed here. The approaches like enriching the content of short text using Wikipedia and incorporating semantic knowledge from an ontology into text clustering needs solid knowledge in natural language processing. Also these methods use high-dimensional representation which may result in a waste of both memory and computation time. Yet another way is to bring sophisticated models for short text clustering. But how to design an effective model is an open question because these models use Bag-of-Words, which are shallow structures and can not preserve the accurate semantic similarity. Further studies employed PSO techniques and deep learning methods in clustering short text. With the help of word embeddings, neural networks show great performance. But RecNN shows high time complexity to construct the textual tree. RNN uses the last word to represent the text representation. Hence the later words are more dominant than earlier words. Recently CNN is the most popular neural network model for applying convolutional filters to capture local features, and achieved a better performance in the case of short text clustering.

References

[1] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, J. Zhao, B. Xu, ”Self-taught convolutional neural networks for short text clustering, Neural Networks” 88 (2017) 2231. doi:10.1016/j.neunet.2016.12.008.

[2] S. Lai, L. Xu, K. Liu, J. Zhao, ”Recurrent convolutional neural networks for text classification”, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[3] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, H. Hao, ”Short text clustering via convolutional neural networks”, in: Proceedings of NAACLHLT (workshop), 2015, pp. 6269.

[4] J. Yin, J. Wang, ”A dirichlet multinomial mixture model-based approach for short text clustering”, in: Proceedings of the 20th ACM SIGKDD inter490 national conference on Knowledge discovery and data mining, ACM, 2014, pp. 233242.

[5] J. Pennington, R. Socher, C. D. Manning, Glove: ”Global vectors for word representation”, Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12.

[6] Q. Le, T. Mikolov,” Distributed representations of sentences and documents”, in: Proceedings of the 31st International Conference on Machine Learning 555 (ICML-14), 2014, pp. 11881196.

[7] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, ”Learning phrase representations using rnn encoder decoder for statistical machine translation”, Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014).

[8] Y. Kim, ”Convolutional neural networks for sentence classification”, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014.

[9] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, ”Recursive deep models for semantic compositionality over a sentiment treebank”, in: Proceedings of the conference on empirical methods in natural language processing (EMNLP), Vol. 1631, Citeseer, 2013, p. 1642.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, ”Distributed representations of words and phrases and their compositionality”, in: Advances in Neural Information Processing Systems, 2013, pp. 31113119.

[11] Leticia Cagnina, Marcelo Errecalde, Diego Ingaramo and Paolo Rosso, An efficient particle swarm optimization approach to cluster short texts, http://dx.doi.org/10.1016/j.ins. 2013.12.010.

[12] Stuti Karol, Veenu Mangat Evaluation of a Text Document Clustering Approach based on Particle Swarm Optimization IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.7, July 2013

[13] J. Tang, X. Wang, H. Gao, X. Hu, H. Liu, ”Enriching short text representation in microblog for clustering”, Frontiers of Computer Science 6 (1) (2012) 88101.

[14] E. H. Huang, R. Socher, C. D. Manning, A. Y. Ng, ”Improving word representations via global context and multiple word prototypes”, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics,2012, pp. 873882.

[15] S. Fodeh, B. Punch, P.-N. Tan, ”On ontology-driven document clustering using core semantic features”, Knowledge and information systems 28 (2) (2011) 395421.

[16] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, C. D. Manning, ”Semisupervised recursive autoencoders for predicting sentiment distributions”, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 151161.

[17] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, S. Khudanpur, ”Extensions of recurrent neural network language model”, in: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, IEEE, 2011, pp. 55285531.

[18] Tunchan Cura, A particle swarm optimization approach to clustering, Expert System with Application 39 (2012) 1582-1588.

[19] J. Turian, L. Ratinov, Y. Bengio, ”Word representations: a simple and general method for semi-supervised learning”, in: Proceedings of the 48th annual meeting of the association for computational linguistics, Association for Computational Linguistics, 2010, pp. 384394.

[20] Diego INGRAMO, Marcelo ERRECALDE and Leticia CAGNINA Paolo ROSSO, ”Particle Swarm Optimization for clustering short-text corpora”, http://users.dsic.upv.es prosso resources Ingaramo.pdf 2009.

[21] X.-H. Phan, L.-M. Nguyen, S. Horiguchi, ”Learning to classify short and sparse text web with hidden topics from large-scale data collections”, in: Proceedings of the 17th international conference on World Wide Web, ACM, 2008, pp. 91100.

[22] S. Banerjee, K. Ramanathan, A. Gupta, ”Clustering short texts using wikipedia”, in: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2007, pp. 787788.

[23] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, ”Self-taught learning:transfer learning from unlabeled data”, in: Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 759766.

[24] G. E. Hinton, R. R. Salakhutdinov, ”Reducing the dimensionality of data with neural networks”, Science 313 (5786) (2006) 504507.

[25] X. He, P. Niyogi, ”Locality preserving projections”, in: Neural Information Processing Systems, Vol. 16, MIT, 2004, pp. 153160.

[26] A. Y. Ng, M. I. Jordan, Y. Weiss, et al., ”On spectral clustering: Analysis and an algorithm”, Advances in neural information processing systems 2(2002) 849856.

[27] M. Belkin, P. Niyogi, ”Laplacian eigenmaps and spectral techniques for embedding and clustering”, in: Advances in Neural Information Processing Systems, Vol. 14, 2001, pp. 585591.

[28] S. Hochreiter, J. Schmidhuber, ”Long short-term memory”, Neural computation 9 (8) (1997) 17351780.

[29] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, R. A. Harshman, ”Indexing by latent semantic analysis”, JAsIs 41 (6) (1990) 391 407.

[30] L. Shang, Z. Lu, H. Li, ”Neural responding machine for short-text conver sation”, arXiv preprint arXiv:1503.02364.

[31] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, ”On the properties of neural machine translation: Encoder-decoder approaches”, arXiv preprint arXiv:1409.1259.

[32] H. Zhao, Z. Lu, P. Poupart, ”Self-adaptive hierarchical sentence model”,arXiv preprint arXiv:1504.05070.

[33] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, Y. Bengio, Renet: ”A recurrent neural network based alternative to convolutional networks”, arXiv preprint arXiv:1505.00393.

[34] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

[35] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, R. Ward, ”Deep sentence embedding using the long short term memory network: Analysis and application to information retrieval”, arXiv preprint arXiv:1502.06922.