A Review of Recent Research Papers on Text Clustering and Categorization

Pranav Panchumarthi
Department of Computer Science
Birla Institute of Technology and Science Pilani
Pilani, India
f20170153@pilani.bits-pilani.ac.in

***Abstract*—This paper gives a brief review of recent research papers in the field of text clustering and categorization, particularly putting a focus on short text. In addition, it provides certain limitations of the reviewed research papers as well as possible solutions or suggestions to further improve the work already been done. The first paper gives a method for improving text clustering, particularly for short text, by including representation learning and training it along with the iterative k-means algorithm in order to create a mix of labeled and unlabeled data for improved performance. The second paper attempts to model short texts using semantic clustering and convolutional neural networks.**

***Keywords*— representation, convolution, max-pooling, clustering, word vector.**

Introduction

One of the most fundamental problems existing in the field of information retrieval is text mining. Conventional text mining methods use text represented as a bag-of-words or term frequency-inverse document frequency vector (TF-IDF) vector. For text clustering, such a representation is then used as input to the k-means algorithm in order to partition the set of texts in different groups or clusters. Under normal circumstances, such a method for clustering or modeling has been proven to be of good accuracy and evaluation of such models as given noteworthy results. However, with the rise of social media in today's generation, queries involving short texts have become more popular than ever, and more importantly, pose an issue as an edge case in the field of information retrieval. One of the issues raised by short texts is the number of unique words in the short text being small. This results in vagueness in the semantics and context of usage of the text, as well as lexical sparsity and ambiguity. The research papers reviewed in this paper are attempts at implementing methods to deal with this issue and in turn provide a method to accurately classify or group short texts of similar nature and meaning.

Paper 1

The paper “Semi-supervised Clustering for Short Text via Deep Representation Learning” extends the classical unsupervised k-means clustering algorithm and combines it with an initial representation of short texts using neural networks to combine both labeled and unlabeled data for training, hence mixing unsupervised and supervised learning in order to provide a unified framework for the short text clustering.

The paper gives a brief explanation of the representation learning implemented. Each word is represented by a dense vector *w* and a combination of such words to form a short text is represented as a matrix *s*. Two different neural networks were used to create the representation of the short text, namely convolutional neural networks (CNN) and long short-term memory (LSTM).

The CNN model applied two sequential operations on a given short text matrix, namely convolution to generate a series of features, and max-pooling, to create a vector of such features. A fully connected layer is then applied to transform the final vector to be used as the representation of the short text to a fixed size.

The LSTM model takes each word vector as input in a sequence and the mean of the hidden states over the entire sentence is taken as the final vector to be used for representation.

The paper then gives a brief review of the k-means algorithm that is conventionally used for clustering and in turn shows how the semi-supervised method aims to include it along with the representation learning. Rather than training the representation model separately, it is included in the k-means algorithm. In this manner, both labeled and unlabeled data are used for both representation learning and text clustering.

This results in the general equation pertaining to the k-means algorithm

$$J_{unsup} = \sum_{n=1}^{N} \sum_{k=1}^{K} (r_{nk} || x_n - \mu_k ||)^2$$

Where x_n is the data point representation of the nth short text, N is the total number of short texts, K is the number of clusters, r_{nk} is a binary value exhibiting whether the data point x_n belongs to cluster *k*, and μ_k is the center of the *k*th cluster, transforming into

$$J_{semi} = \alpha \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left|\left| f(s_n) - \mu_k \right|\right|^2$$

$$+ (1\text{-} \alpha) \sum_{n=1}^{L} \{ \left|\left| f(s_n) - \mu_{gn} \right|\right|^2 +$$

$$\sum_{j \neq g_n} [l + \left|\left| f(s_n) - \mu_{gn} \right|\right|^2 - \left|\left| f(s_n) - \mu_j \right|\right|^2] \}$$

The first term in the equation is the same as that present in the k-means algorithm. The second term is introduced to encourage and add weight to labeled data as an addition to the conventional k-means algorithm. Both algorithms aim to minimize the left-hand side. For the above-modified equation, we assume that the labeled data set is given by $\{(s_1, y_1), (s_2, y_2), \ldots, (s_L, y_L)\}$ and the unlabeled data set is $\{(s_{L+1}, s_{L+2}, \ldots, s_{L+N}\}$, where y_i is the given label for the short text s_i.

Previous work related to this paper: The initial direction taken by this paper to encode texts into distributed vectors with neural networks was introduced by Hinton and Salakhutdinov in 2006 and further elaborated by Xu in 2015.

Semi-supervised approaches to resolve the issue of clustering based on different intentions, that is, semantics were first developed by Bilenko in 2004, Davidson and Basu in 2007, and Bair in 2013. The prominence and uniqueness of this paper lies in its attempt to propose a unified framework for short text clustering using both approaches integrated together.

Overview of the procedure:

The sequence of steps followed by the new algorithm introduced by the paper are as follows:

1. The training procedure begins with the random initialization of centroids and neural networks.
2. Each short text is assigned to the nearest centroid based on its representation as provided and calculated by the neural networks.
3. The cluster centroids are re-evaluated based on the individual assignments of each short text to the respective cluster in the previous step.
4. Steps 2-4 are repeated until convergence.

This method of semi-supervised clustering was evaluated on four short text datasets, namely *question_type, ag_news, dbpedia, and yahoo_answer,* and a comparison was provided with various other current models such as bow, tf-idf, metric-learn-bow, metric-learn-idf, and metric-learn-ave-vec. The semi-supervised clustering using CNN showed the greatest AMI(Adjusted Mutual Information(Vinh et al., 2009)) and ACC(accuracy(Amig´o et al., 2009)) among most other models, which were the factors used for evaluation.

Paper 2

The paper "Semantic Clustering and Convolutional Neural Network for Short Text Categorization" brings forward a method to model short texts based on semantic clustering and convolutional neural networks(CNN).

The motivation of this paper stems from a similar cause as the previous paper reviewed, that is, the lack of adequate results in categorization when using conventional methods to deal with short texts due to sparsity and ambiguity due to a low number of words. This paper hence attempts to introduce extra knowledge by pre-trained word embeddings and exploit the contextual and semantic information of short texts to improve their representation.

In order to do this, the paper takes advantage of the fact that neighbors of any given word are semantically related in an embedding space (Mikolov et al., 2013b). Hence clustering methods may be employed to discover semantic cliques, or clusters of semantically related words. This is done by using the fast clustering algorithm(Rodriguez and Laio, 2014), which is based on searching density peaks. Two quantities are computed for a data point representing a short text, namely local density $\rho_i = \sum_j \chi(d_{ij} - d_c)$.

Here d_{ij} is the distance between data points specified by i and j, and d_c is a threshold distance. Additionally, distance δ_i from points of higher density is measured by the equation:

$$\delta_i = min_{j:\rho_j>\rho_i}(d_{ij}), \text{ if } \rho_i < \rho_{max}$$

$$= max_j(d_{ij}), \text{ otherwise.}$$

According to the definitions given, word embeddings with large ρ and δ simultaneously are chosen as cluster centers, and are consecutively labeled using the corresponding words.

Previous work related to this paper: Conventional methods used for text categorization fail in the case of short text due to the sparsity in their representation(Sriram et al., 2010). The concept that neural networks such as CNN may be used for language modeling and word embeddings can be learned simultaneously was proposed by Mnih and The in 2012. Mikolov et al. (2013b) introduced the continuous Skip-gram model that is an efficient method for learning high-quality word embeddings from large-scale unstructured text data.

Overview of the procedure:

The proposed architecture is as follows:

1. For a short text consisting of words $\{w_1, w_2, \dots, w_n\}$, it may be represented in the form of a combination of vectors, or a matrix. This projected matrix is obtained from the lookup table initialized by the pre-existing word embeddings.
2. The next step consists of identifying multi-scale semantic units. This is done by performing semantic composition over *n-gram* embeddings. The obtained meaningful semantic units are then individually used to calculate the Euclidean distance between them and the previously obtained semantic cliques or semantic clusters. This is compared with a well-defined threshold and those semantic units whose distance is less than the threshold are passed on to constitute semantic matrices.
3. The sequential operations convolution and max-pooling are then executed on the semantic matrix.
4. A softmax classifier is finally used to predict the probability distribution of the various short texts over a wide array of categories.

The experiments conducted as per the paper were done on two datasets, namely Google Snippets and TREC questions dataset. Three pre-trained word embeddings, namely Senna, GloVe, and Word2Vec were used for initializing the lookup table. The classification accuracy of the experimental model was compared with the accuracy of other current state of the art models on the same datasets. Some of the models used in the comparison include Dynamic Convolutional Neural Networks (Kalchbrenner et al,2014), SVMs Parser (Silva et al., 2011), and CNN-TwoChannel (Kim et al., 2014). In the TREC dataset, the model developed by this paper, that is, semantic-CNN, obtained an accuracy of 97.2 using the GloVe word embeddings, while in the Google Snippets dataset, semantic-CNN obtained an accuracy of 85.1 using the Word2Vec word embeddings. In both datasets, semantic-CNN exceeded the other current models in this aspect.

Limitations of Paper 1

One of the drawbacks of the first paper was related to the efficiency and speed of the clustering algorithm. The experiment was able to run the algorithm completely only on the *question_type* dataset and was unable to do so with the other three datasets due to their large size. Instead, 10% of each of the datasets were sampled as labeled data and were used to run the model on the remaining portion of the dataset for evaluation. This is a common practice and was already introduced by Xu et al., 2015, but still results in a lower performance than the maximum possible.

Many of the limitations of the experiment process of the paper lie in the setting and choice of the hyperparameters and variables of the process. Such variables play a major role in the accuracy of the model but only a few have been chosen for optimization without loss of performance, such as the word vector length being set to 300. The word vectors were pre-trained on the Word2Vec toolkit for representation but it is not known that this is the best toolkit to use. In the case of the second paper, the model was trained using three different word embeddings, namely Senna, GloVe, and Word2Vec, therefore providing more information on a comparison and the extent of importance the different toolkits hold in terms of the final accuracy based on training on one such toolkit. Similarly, 100 was found to be the best output dimension in text representation models. This may change on training with a different toolkit and hence may affect the results further.

The parameter α is directly proportional to the importance and weight unlabeled data holds. The optimal value of α was found by varying it from values starting from 0.00001 to 0.1. The peak graphically was found for both LSTM and CNN models to be at approximately 0.001 but in the experiment, the value 0.01 was associated with α.

Limitations of Paper 2

The second paper puts greater importance on improving the representation of short texts in order to improve the accuracy of classification. However, a minor drawback is that sufficient information was not provided regarding the parameters involved in the process of convolution as well as k-max pooling. These parameters, such as number and size of feature detectors as well as the value of k in max pooling definitely play a role in determining the optimal accuracy of the representation of short texts.

The final results of the model should be compared with the current state of the art models to determine whether the model is noteworthy and competitive against already existing models. However minimal information was actually provided in the comparison, and the experimental model was the only one to be tested on both datasets completely. No other dataset was run and evaluated on both datasets.

Suggestions and future direction

First Paper: It would be useful to extend the existing research to find and optimize the various parameters involved in the algorithm proposed. Training the word vectors on not just one but possibly a variety of toolkits, as done in the second paper to determine its impact on the final results could prove helpful in further improving performance and results. Similar steps can be taken in varying the output dimension along with the toolkit used for training if they are known to be dependent on each other. Similarly further experimentation on finding a balance between the role of labeled and unlabeled data to achieve optimal results, that is, running the model on varying values of α may help achieve a better model.

Second Paper: A similar suggestion may be extended to the second paper, that is, further experimentation on hyperparameters to further improve the accuracy of classification. Varying the values of the size and number of features for convolution as well the value of parameter k in k-max pooling may play a role in determining the effectiveness of representation of word vectors and in turn the accuracy of the model.

In addition, it can be said that the model may compare with current state of the art models. However, a proper comparison has not been provided and further comparison on multiple datasets with multiple models should be done to the full extent. This is crucial as it is the final result that provides credibility to the model.

Future Directions

The second paper puts greater importance on improving word vector representation as the main means of improving categorization accuracy. The first paper also applies representation learning in integration with the k-means clustering algorithm to iteratively run until convergence and attempt to successfully perform clustering on short texts. In addition to the representation learning done by the semi-supervised method, the second paper uses semantic clustering to find semantic cliques and also identifies candidate semantic units. In this manner, one can say that the word vector representation in the second paper is more refined than the first paper, and factors in semantic information and context while running the representation model.

Keeping this in mind, the semi-supervised method proposed by the first paper may be improved further by employing the techniques, namely semantic clustering and identification of semantic units, used by the second paper to add quality to the word vector representation.

The very essence of the semi-supervised model is to add weightage to labeled data and integrate representation learning with the classic unsupervised k-means algorithm rather than using it alone to improve text clustering in the edge case of short texts. The results show that adding representation learning improves accuracy and AMI. Hence it is valid to attempt to further improve the representation of word vectors by refining it further using semantic clustering and identifying candidate semantic units in an attempt to improve the semi-supervised model as a whole.

Conclusion

Both research papers show promising results in the field of text mining, one dealing with text clustering and the other categorization, particularly for short texts. The first paper proposes a semi-supervised model integrating the conventional k-means algorithm used for clustering with representation learning to accurately cluster or group short texts. The second paper provides a novel method to model short texts using semantic clustering and by performing semantic composition to detect candidate semantic units.

This paper covered both papers along with the methodology and procedure followed to develop and test the models created and further pointed out limitations of both papers, as well future directions that can be taken to improve the models proposed for each individual paper, and for text mining for short texts as a whole.

REFERENCES

[1] Z. Wang, H. Mi, A. Ittycheriah, “Semi-supervised Clustering for Short Text via Deep Representation Learning”, Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, pp. 31-39, August 2016.

[2] P.Wang, J. Xu, B. Xu, C. Liu, H. Zhang, F. Wang, H. Hao, “Semantic Clustering and Convolutional Neural Network for Short Text Categorization”, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing(Volume 2: Short Papers), pp. 352-357, July 2015.