Document Clustering Using Attentive Hierarchical Document Representation

Arezoo Hatefi, Frank Drewes
Department of Computing Science, Umeå University, Sweden
{arezooh,drewes}@cs.umu.se

Abstract

We propose a text clustering algorithm that applies an attention mechanism on both word and sentence level. This ongoing work is motivated by an application in contextual programmatic advertising, where the goal is to group online articles into clusters corresponding to a given set of marketing objectives. The main contribution is the use of attention to identify words and sentences that are of specific importance for the formation of the clusters.

1 Introduction

Text clustering is an unsupervised machine-learning task that serves to group textual documents based on similarity. Our interest in the problem arises from the application area of contextual programmatic advertising which requires a grouping of news articles into clusters, to find appropriate online contexts for a given advertising campaign. Cluster centroids are initialized based on prior knowledge (provided by, e.g., campaign descriptions in the form of keywords) and are shifted during training to reflect the actual data.

In clustering text documents using neural methods, the most important choices affecting the result concern the feature vectors and the similarity or distance measure. A common way to create the document feature vectors is to use vectors with as many dimensions as there are relevant words in the vocabulary V, i.e., there is a dimension i_w for each $w \in V$. One then fills the i_w-th position of the vector with the *term frequency—inverse document frequency* (TF-IDF) score of w. Since the vocabularies are usually very large, this method results in high-dimensional feature vectors. In such cases, clustering according to distance metrics similar to Euclidean distance, which is popular in other types of clustering, is known to become unstable (Aggarwal et al., 2001). As a solution, dimensionality reduction and feature transformation methods (including linear transformation like Principal Component Analysis (Wold et al., 1987) and non-linear transformations such as kernel methods (Hofmann et al., 2008) have been extensively studied to map the feature vectors into a new feature space of lower dimensionality, but this also limits the expressiveness of the resulting vectors. A more recent alternative is to reduce dimensionality by nonlinear mappings corresponding to the behavior of autoencoders (Baldi, 2012), a type of deep neural network which is capable of generating compact feature vectors (Yang et al., 2017; Xie et al., 2016). Altough these and similar efforts have tried to make TF-IDF vectors more efficient by reducing their dimensionality, the intrinsic problem of these representations is that they do not account for linguistic context, word order, and inter-word interactions.

In natural-language processing (NLP), TF-IDF vectors are increasingly being replaced by word embeddings, i.e., distributed representations of words such as word2vec and GloVe. Clustering is no exception, because word embeddings have been shown to generate more informative document representations. Recently, pretrained word embeddings from unsupervised language modelling architectures like BERT (Devlin et al., 2018) (which models context using the attention mechanism of Bahdanau et al. (2014); Luong et al. (2015) have led to significant improvements on many NLP tasks. To our knowledge, these contextualized word embeddings have so far been investigated for text clustering only under the Bag Of Words (BOWs) model, which does not make use of the document structure formed by words and sentences (Park et al., 2019).

In this paper, we report on ongoing work with the aim to fill this gap by exploiting attention-based methods to improve clustering. Assume that we want to cluster documents into N clusters whose centers are initialized by N sets of keywords. We propose to use attention to generate N representations for each document, one per cluster, and to cluster the documents based on these representations. The rationale behind using cluster-specific representations is that individual words and sentences in a document differ in their information value depending on the cluster in question.

To generate the document representations, we follow Yang et al. (2016) and use a hierarchical model with several levels of attention mechanisms, two at word level and two at sentence level. Each cluster-specific document representation is obtained by first building sentence representations from word representations, and then aggregating sentence representations into a document representation, where attention allows the model to focus on semantically relevant words and sentences. Like Park et al. (2019), we use cosine similarity as the distance measure because the direction of vectors, as opposed to their magnitude, usually is what captures linguistic meaning, and also because cosine similarity yields good results even for high-dimensional spaces (see Aggarwal et al. (2001)).

In the next section, we describe how we aim to use attention in order to create document representations that serve as a basis for clustering. Sections 3 and 4 describe the clustering method and the datasets and evaluation method we intend to use. Section 5 concludes the paper.

2 Attention-based Hierarchical Document Representation

The overall architecture of the attention-based hierarchical network for generating a document representation is shown in Figure 1. This architecture includes two levels: the first consists of a word encoder and a word-level attention layer which output sentence representations. The second level, which lies on top of the first, consists of a sentence encoder and a sentence-level attention layer which produce document representations. We describe these layers in detail in the following sections.

2.1 Encoder Layers

The architecture of the word and sentence encoders corresponds to a single encoder layer of the BERT model by Devlin et al. (2019). These layers compute the attentive transformed representation of all positions in the input sequence using a multi-head self-attention mechanism followed by a position-wise fully connected, feed-forward network.

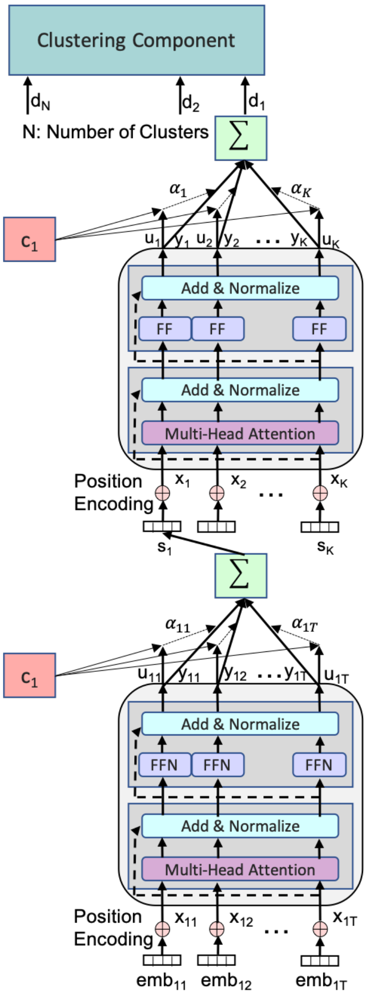

Figure 1: The proposed architecture for document clustering using word-level and sentence-level attentions.

The main building block of the multi-head attention framework by Vaswani et al. (2017) is scaled dot-product attention (Lu et al., 2016), which operates on the query Q and key K of dimension d_k, and the value V of dimension d_v as follows:

$$Attention(Q, k, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V \ .$$

As we encode a position of the input sentence, the self-attention mechanism determines how much focus to place on other parts of the input. The vectors Q, K, and V are created by linearly projecting input embeddings by three weight matrices which are updated during the training process, namely $W^Q, W^K \in \mathbb{R}^{d_{model} \times d_k}$ and $W^V \in \mathbb{R}^{d_{model} \times d_v}$.

In the multi-head attention framework with $n \in \mathbb{N}$ attention heads, n copies are created for each triple (Q, K, V), using separate learned projections. Then, a scaled dot-product attention is applied to each version, yielding n versions of d_v

dimensional output values. The final values are produced by concatenating and, once again, projecting the output values:

$$\begin{aligned} &\mathit{MultiHead}(Q,k,V) = \\ &\qquad \mathit{Concat}(\mathit{head}_1,\ldots,\mathit{head}_n)W^O \\ &\mathit{head}_i = \mathit{Attention}(QW_i^Q, KW_i^K, VW_i^V) \ . \end{aligned}$$

In addition, the matrix $W^O \in \mathbb{R}^{nd_v \times d_{model}}$ is updated during the training process.

The output of the attention sub-layer is fed to a convolutional neural network consisting of two transformations with a Rectified Linear Unit (ReLU) activation in between which is applied on each position separately and identically:

$$\begin{aligned} &\mathit{FFN}(x) = \\ &\qquad \mathit{Conv}(\mathit{ReLU}(\mathit{Conv}(x)+b_1))+b_2 \ . \end{aligned}$$

A residual connection (He et al., 2016) followed by a layer normalization (Ba et al., 2016) is applied around each of these two sub-layers. Thus, the final output of each sub-layer is computed by:

$$\mathit{Sublayer}_{out} = \mathit{LayerNorm}(x + \mathit{Sublayer}(x))$$

where $\mathit{Sublayer}(\text{-})$ denotes function computed by the sub-layer.

In addition, since our attention-based encoder layer does not use the order of the sequence, we make the position-related information available for it by encoding positions into d_{model} dimensional vectors and then adding these to the word and sentence embeddings. For generating position encodings, we apply the method proposed by (Vaswani et al., 2017), that uses sine and cosine functions of different frequencies. Consequently,

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{model}}) \end{aligned}$$

where pos is the position in the sequence and i is the dimension.

2.2 Attention Layers

Assume we want to group documents into N clusters. So, we generate N different representations for each document attending to one of the cluster centroids each time. In the following, we describe how we generate document representations $d_j, j \in [1,N]$ with respect to a cluster c_j with cluster centroid cc_j.

We assume a document d has K sentences s_i. In turn, s_i consists of T_i words w_{it} $(t = 1,\ldots,T_i)$. At first, we embed the words into vectors using a pretrained GloVe embedding matrix W_e:

$$emb_{it} = W_e w_{it}, \ \ t \in [1,T_i] \ .$$

Then we encode word positions into vectors through the encoding matrix W_{pos} created using the method by Vaswani et al. (2017), and add the position encodings to word embeddings:

$$\begin{aligned} pos_{it} &= W_{pos} t, \ \ t \in [1,T_i] \\ x_{it} &= emb_{it} + pos_{it} \ . \end{aligned}$$

We feed input vectors to the word encoder layer to obtain the contextual word embeddings:

$$y_{it} = \mathit{Encoder}_{word}(x_{it}), \ \ t \in [1,T_i] \ .$$

Not all the words of the sentence contribute equally to the sentence representation calculated with respect to a specified cluster: the more similar a word is to the cluster centroid, the more able is it to represent the sentence. So, we propose a word-level attention mechanism based on similarities to the cluster centroid for assessing the relative importance of different words. First, we apply a projection layer followed by a nonlinearity on contextual word embeddings y_{it} to obtain their hidden representations u_{it}. Then, we employ the cosine similarity measure to compute the similarity between hidden vector u_{it} and centroid cc_j. We normalize the similarities of all sentence words with a SoftMax function, and use them as weights in a weighted sum of word representations y_{it} to form sentence vector s_i:

$$\begin{aligned} u_{it} &= tanh(W_w y_{it} + b_w) \\ \alpha_{it} &= \frac{exp(u_{it}^T cc_j)}{\sum_t exp(u_{it}^T cc_j)} \\ s_i &= \sum_t \alpha_{it} y_{it} \ . \end{aligned}$$

Given the sentence vectors s_i, we can produce a document vector in a similar way. We obtain the position encodings $pos_i, i \in [1,K]$, of the sentences through the position encoding matrix W_{pos}, add these vectors to sentence vectors, and feed the results to sentence encoder to get the contextual sentence embeddings y_i:

$$\begin{aligned} pos_i &= W_{pos} i, \ \ i \in [1,K] \\ x_i &= s_i + pos_i \\ y_i &= \mathit{Encoder}_{sent}(x_i) \ . \end{aligned}$$

To reward sentences that are more important for representing document d regarding cluser c_j, we introduce a sentence level attention mechanism that computes sentence importance as the similarity between the sentence hidden vector u_i and cluster centroid cc_j. Sentence hidden vector u_i is generated by applying a projection layer followed by a nonlinear layer on the sentence contextual representation s_i. For measuring similarities, again we use $cosin$ similarity and normalize them with a SoftMax function. Finally, we compute document representation d_j as a weighted sum of the sentence representations based on their importance weights:

$$u_i = tanh(W_s y_i + b_s)$$

$$\alpha_i = \frac{exp(u_i^T cc_j)}{\sum_i exp(u_i^T cc_j)}$$

$$d_j = \sum_i \alpha_i y_i \quad .$$

3 Document Clustering

Consider a set of M documents $D = \{D_m\}_{m=1}^{M}$. Each document has N different representations d_{kj} ($k \in [1, M]$, $j \in [1, N]$) which are generated using the method proposed in the previous section. To assign d_k to a cluster, we compute the cosine similarity between the cluster-specific document representations d_{kj} and the corresponding cluster centroids cc_j. This results in an N-dimensional similarity vector s_k. By applying a SoftMax function on this vector, each dimension s_{kj} can be perceived as the probability of assigning document d_k to cluster c_j:

$$s_{kj} = \frac{d_{kj} \cdot cc_j}{\|d_{kj}\| \|cc_j\|} \qquad p_{kj} = \frac{exp(s_{kj})}{\sum_j exp(s_{kj})}$$

where $\cdot$ denotes the dot product and $\|.\|$ denotes the length of the vector. We suppose the correct cluster for each document is the dimension with the highest probability in its similarity vector. We call this cluster the soft target of the document and denote it with $\hat{t}$, i.e.

$$\hat{t}_k = \underset{j}{argmax}(p_{kj}) \quad .$$

For optimizing model parameters, including the cluster centroids θ, we use Stochastic Gradient Descent (SGD) together with an objective function based on Negative Log Likelihood (NLL).

$$NLLL = \underset{\theta}{min} \sum_{k=1}^{L} NLL(p_k, \hat{t}_k) \quad .$$

Since the computed soft targets $\hat{t}_k$ of documents are inaccurate, in every training batch $\{d_i\}_{i=1}^{B}$, we only use the $L < B$ documents with the highest soft target probabilities for computing loss function.

We also investigate another approach for updating cluster centroids. After assigning all documents of batch b to clusters, for each cluster c_j, we choose W documents with the highest probabilities and extract G words with the highest attentions (computed while generating the document representation) from each of them. The updated cluster centroid cc_j will be the average of the preceding centroid and the embeddings of extracted words.

For the initialization of cluster centroids we consider two options. Since this research is motivated by an application in which we have N intended clusters roughly described by keywords, we can initialise the cluster centroids with the average of the embeddings of those cluster keywords. The second option is to use any standard centroid initialization algorithm like the seed strategy proposed by Arthur and Vassilvitskii (2007).

4 Dataset and Evaluation Metrics

To be able to compare our results with previous work in the literature, we will use labeled document datasets available for document classification and question answering, namely “Yahoo Answers” (Zhang et al., 2015), “FakeNewsAMT” (Pérez-Rosas et al., 2018), and “SQuAD 1.1” (Rajpurkar et al., 2016).

Since the evaluation of unsupervised clustering accuracy without ground truth is difficult (Palacio-Niño and Berzal, 2019), we will evaluate our model by applying it to datasets with document labels, using the labels for measuring clustering accuracy, but not for training or clustering.

5 Conclusions

As mentioned in the introduction, the approach described in this paper is work in progress. In particular, we have not yet been able to evaluate the proposed method as the first author is currently implementing it. As soon as the implementation is complete, experiments will be conducted to evaluate the method as described in Section 4.

Acknowledgment. We thank Johanna Björklund for valuable discussions and proofreading and the reviewers for helpful comments.

References

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1: Long and Short Papers*, pages 4171–4186.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 770–778.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. 2008. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 289–297. Curran Associates, Inc.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Julio-Omar Palacio-Niño and Fernando Berzal. 2019. Evaluation metrics for unsupervised learning algorithms. *CoRR*, abs/1905.05667.

Jinuk Park, Chanhee Park, Jeongwoo Kim, Minsoo Cho, and Sanghyun Park. 2019. ADC: Advanced document clustering using contextualized representations. *Expert Systems with Applications*, 137:157–166.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.