Coherence-based Second Chance Autoencoders for Document Understanding

Saria Goudarzvand[1], Gharib Gharibi[1] and Yugyung Lee[1]

[1]University of Missouri-Kansas City, Kansas City, Missouri

Abstract

The application of conventional autoencoders for textual data often leads to learning trivial and redundant representations due to the high dimensional nature of the text, sparsity, and following power-law word distribution. In order to address these challenges, we introduce a new autoencoder, termed *CSCAT* (*C*oherence-based *S*econd *C*hance *A*utoencoder for *T*ext), which uses competitive learning to select k winning neurons in the bottleneck layer that becomes specialized in recognizing specific patterns–leading to learning semantically significant representations of the text. *CSCAT* introduces a new competition learning based on a measure of consistency to eliminate incoherent features. Our experiments demonstrate that *CSCAT* achieves outstanding performance on several tasks, including classification, topic modeling, and document visualization compared to *LDA*, *k-sparse*, *KATE*, *NVCTM*, and *ProdLDA*.

Keywords

Topic Modeling, Autoencoder, Data Analysis, Second Chance Learning

1. Introduction

Deep neural networks [1] have revolutionized many domains, especially unstructured data, including computer vision [2], speech recognition [3], and text classification [4] to name a few. While most current neural network applications use supervised learning, unsupervised learning has also presented significant advances in extracting patterns in unlabeled data with reasonable efficiency. For example, unsupervised models have been used to aid information retrieval [5], discover patterns in medical datasets [6, 7], and video prediction [8].

One of the most popular unsupervised deep learning algorithms are autoencoders [9, 10]. An autoencoder is a neural network that learns data representations by reconstructing the input data at the output layer (i.e., $y^{(i)} = x^{(i)}$), where $y^{(i)}$ is the network's output (prediction) for the $x^{(i)}$ input sample. Thus, the main objective for autoencoders is to learn the important features of the input data by constraining the size of the middle layer named *bottleneck*, often by reducing its dimension less than the input layer.

While autoencoders have demonstrated significant results in several domains, most notably visual applications such as image compression [12] and denoising images [13]; it has been challenging to use autoencoders for textual data due to the text high-dimensionality and sparsity [11]. Adding to the aforementioned challenges, autoencoders are also known to learn trivial representations of the text due to the power-law word distribution [14].

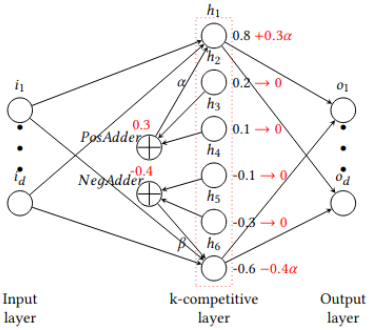

Figure 1: Example illustrating the KATE approach [11]. All layers are fully-connected. Values in Red represent activations after the competition.

Fortunately, the research community has identified the need for proper methods to utilize autoencoders for textual-data applications, leading to the emergence of several methods that address these challenges. This new body of research introduced several innovations, including neural autoregressive topic models [15], deep belief networks for topic modeling [16], and neural variational inference for text processing [17, 18]. Additionally, in order to better understand textual data and learn more semantically meaningful representations, other research established the idea of k-competitive autoencoders, which produced impressive results in the textual data domain, including *K-sparse* [19] and *KATE* (K-competitive Autoencoder for TExt) [11] depicted in Figure 1.

The primary principle behind k-competitive autoencoders is to select the top k "winner" neurons that conquer

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY

sgnbx@mail.umkc.edu (S. Goudarzvand); ggk89@mail.umkc.edu (G. Gharibi); leeyu@umkc.edu (Y. Lee)

0000-0001-9616-893X (S. Goudarzvand); ? (G. Gharibi); ? (Y. Lee)

the activation values (a.k.a. *power*) from the loser neurons. By incorporating competition among the neurons of the hidden layers, these methods aim to specialize the winner neurons in learning meaningful representations of the text. The top k winners are chosen based on some competition criteria.

For instance, K-sparse focuses on maintaining sparsity by preserving the k highest activations during training and the αk highest activations during testing, where α is a hyperparameter. Similarly, KATE selects k winners made up of the $\lceil k/2 \rceil$ largest positive activations and the $\lfloor k/2 \rfloor$ largest absolute negative activations. Those winner neurons then acquire the total energy of the loser neurons, which become inactive, i.e., set to zero. k is a hyperparameter representing the desired number of neurons to compete, and it is strongly related to the number of topics.

K-sparse is vulnerable to the "dead hidden neurons" problem caused by adding too much sparsity (low k values), and therefore some neurons can never be updated in the back-propagation process. While this issue can be addressed by incorporating a sparsity scheduling technique, this solution adds significant overhead during the learning process. In contrast, KATE was built on top of K-sparse and solved the dead hidden neurons problem. However, its competition considers the largest positive and negative activations (the weakest neurons are loser neurons) only–leading to ignoring some essential knowledge preserved in low signal neurons that are never selected as winners. Indeed, our research proves that some of the neurons that maintain weak signals during early training cycles might hold important information on representative features.

To this end, we present CSCAT (**C**oherence-based SCAT), a novel autoencoder that builds on earlier work in k-competitive learning called SCAT [20]. CSCAT achieves two main innovations over the previous k-competitive learning methods. First, it provides a second chance for the *weakest* neurons to reveal their potential, i.e., important topics that would otherwise be ignored. Second, a coherence-based filtration technique that removes non-coherent neurons from the competition process. Our extensive evaluation demonstrates that these two innovations can lead to better results compared to the prior work in this domain. To summarize, our work contributes the following:

- A novel idea of a coherence-based criterion for filtering neurons that are eligible to enter the learning competition produced by the SCAT layer. This process prevents neurons from a low-coherence score to more than $k/2$ other neurons entering the competition. We hypothesize that eliminating not coherent features during the training phase will result in better topic representations.
- A thorough evaluation and comparison to KATE, K-sparse, LDA [21], NVCTM [22] and ProdLDA. The evaluation tasks include topic modeling, topic coherence score, document classification, and visualization using three datasets: 20 Newsgroups, Wiki10+, and Reuters dataset.

2. RELATED WORK

For topic modeling of document collections, Latent Dirichlet Allocation (LDA) has gained prominence. By constructing a probability distribution across words, the model seeks to reveal the hidden structure of documents as a combination of topics. Non-parametric learning [23], sparsity [24, 25] and efficient inference [26]are only a few of the LDA versions that have been developed. The fundamental flaw in the LDA is that the order of words was not taken into account because of the underlying use of "bag of words" [27]. To solve this issue, the Topic Keyword Model (TKM) was created, which takes into account the position word i in a context [28]. TKM fully utilized the critical idea of a joint probability $D \times W$ from the aspect model [29] to highlight certain aspects of the topics in the documents. TKM conceives the main ideas of the aspect model, but in text documentations, the position i of a word was also taken into consideration. A word's context was taken into account. This means that if a word appears repeatedly in the same document but with different neighboring words, each occurrence may have a different probability. In [30] , a new version of LDA called ProdLDA was released. This topic model substitutes the mixture model used in LDA with a product of expert distribution across particular words. In terms of topic coherence score and qualitative assessment, ProdLDA creates better topics than regular LDA. When the model was tested based on accuracy, however, the results were not similar, as shown in table 3.

Even with ideal reconstructions, autoencoders often only extract simple representations of text data; however, by adding proper regularization to the models, more meaningful representations can be generated. Many autoencoder versions have lately been proposed based on this premise [19, 31, 32]. K-competitive autoencoders, such as KATE, are recent autoencoders that perform well on text classification tasks. KATE (K-competitive Autoencoder for TExt) builds on k-sparse for learning meaningful representations by introducing competition among hidden layer neurons. KATE's approach is to select k winner neurons composed of $ceil k/2$ largest positive activations and $floor k/2$ largest absolute negative activations, which then gain the energy of loser neurons.

Overall, CSCAT's technique is fairly similar to that of traditional k-competitive autoencoders. However, we

choose the winners from among the strongest and weakest positive and negative neurons, guaranteeing more equal competition and giving the weakest negative and positive neurons a second chance. Second, before starting the competitive process, we offer a filtration mechanism that filters out incoherent neurons. This guarantees that the winning neurons are distinctive and coherent.

Unsupervised learning has seen a lot of success with generative models for learning from unlabeled data. Deep Belief Networks (DBN) are a type of deep generative model in which the input data is reconstructed using a deep autoencoder based on the top two layers of a direct acyclic graph [33]. Maaloe et al. [16] introduced a topic modeling approach based on DBN. The neural variational inference (NVI) approach makes the deep generative framework, such as variational autoencoders, suitable for topic modeling [17]. Neural Variational Document Model (NVDM) is a variational autoencoder based neural network for document modeling [17]. One disadvantage of NVDM is that it ignores the correlation between the topics. Liu et al. [22] presented the Neural Variational Correlated Topic Model (NVCTM), a centralized transformation mechanism that reshapes topic distributions to express links between topics. NVCTM consists of two components: the inference network with a centralized transformation flow and a multinomial softmax generative model. NVCTM's efficiency in capturing perplexity, topic coherence, and document categorization tasks has been proven through rigorous testing. Although this model frequently earns a high coherence score, its classification performance is inferior to that of other similar models.

3. Approach

Autoencoders draw their technical advantage from constraining a bottleneck layer, often by reducing its dimensions, to force the neural network to learn representative features from the data, and then used to reconstruct the data at the output layer. However, latent representation layers usually learn the minimal set of trivial, redundant features required to reconstruct the input data. When it comes to topic modeling, features are frequently chosen based on the most common words based on power-law word distributions, which might hinder the whole process and lead to ignoring important topics linked to less frequent terms. Thus, we propose a competitive learning approach that not only encourages the competition among the most significant activation values but also (1) gives a second chance to the neurons with the weakest activations and (2) inactivates the neurons with the lowest coherence during training phase. Figure 2 illustrates a toy example of the training process in CSCAT.

The competition criterion in our study is based on a unique finding in the Neuroscience area that has already spawned numerous novel deep learning approaches. Mingorance et al. [34] discovered that the kinase JNK (c-Jun N-terminal protein kinase) gives the weaker neurons a *second chance* before choosing the neurite that best meets the criteria to produce an Axon. Weak neurons will never have a chance to form an Axon unless there is a fair allocation of power. Without this fair redistribution of power, weak neurons will never receive a chance to form an Axon. Using this analogy, we designed our k-competitive learning approach to provide the weakest activations a second chance and then selecting the neurons that activate after energy redistribution. Otherwise, neurons with low power will never make into the autoencoder's latent features.

Our experiments reflect the findings of [35] from the Neuroscience domain into the deep learning domain and prove the correctness of our initial hypothesis–that some essential features might be buried in neurons with low activation values that never receive a chance to appear in the fully-trained network due to initialization randomness or initial low frequency of important words. Based on this idea, we suggested SCAT in a prior work and then extended it with unique coherence-based filtering mechanism in the CSCAT, which we present in this paper. We explain the approach of CSCAT in the following.

3.1. Definition

We define CSCAT as a neural network accepting an input vector $x \in \mathbb{R}^d$ with d-dimensions, and $W \in \mathbb{R}^{d \times m}$ is the weight matrix, and $h_1, h_2, ..., h_m$ are the m hidden layers, and $\hat{x} \in \mathbb{R}^d$ is the output vector. The activation values at the hidden layers are calculated as $z = g(Wx + b)$, where g represents the activation function and b is the bias at the encoder side. We use $tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ as the activation function for the hidden neurons and $sigmoid(x) = \frac{1}{1+e^{-1}}$ as the activation function for the output neurons. The output neurons are defined as $\hat{x} = g(W^T z + c)$, where W^T is the weight matrix obtained by weight tying–sharing–and c is the bias at the decoder side. We use the binary cross-entropy loss function, $l(x, \hat{x})$, as defined in Equation 1, where V is the vocabulary of the dataset.

$$l(x, \hat{x}) = -\sum_{i \in V} x_i log(\hat{x}_i) + (1 - x_i) log(1 - \hat{x}_i) \quad (1)$$

Given a vocabulary V and the number of times, n_i, a word i is mentioned, the input vectors, x_i, are calculated as given in Equation 2.

$$x_i = \frac{\log(1 + n_i)}{\max_{i \in V} \log(1 + n_i)} \; for \, i \in V \quad (2)$$

Given our model definition, the CSCAT approach goes through the following steps during the training phase at

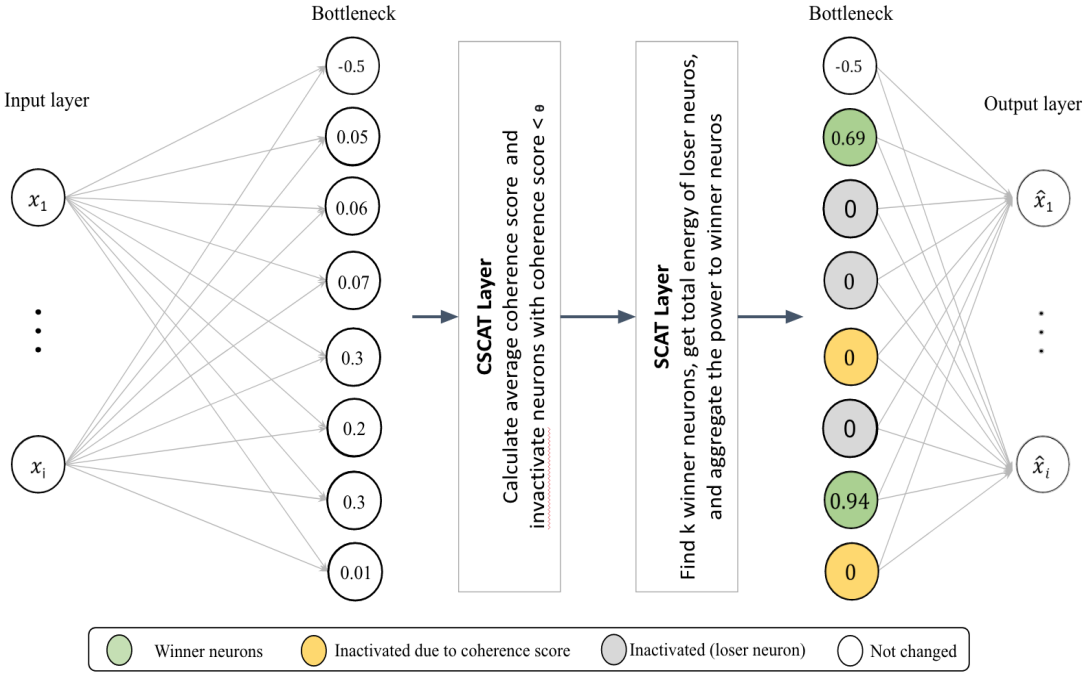

Figure 2: Example illustrating CSCAT approach. All layers are fully-connected, but the connections are light-colored for illustration purposes.

the bottleneck layer (see Algorithm 1): (1) filter out the neurons based on a given coherence measurement, (2) select top k winner neurons, (3) inactivate loser neurons and aggregate their power to the winner neurons, and then continue the regular training process. Refer to Table 2 for a list of notations used in the rest of the paper. We further explain each of the steps, as mentioned earlier in the following.

3.2. Coherence-Based Rule

One of the major issues in clustering particularly in topic modeling is that the final topic words are not coherent. In fact, the association among the top words per topic could be a good indication of the highly correlated words. In training phase, we want to ensure that the words learned by the model are logically consistent per topic.

Point-wise mutual information [36] is one measure of the statistical independence of observing two words in close proximity. Given a learned W, the practice to extract top-N most probable words for each topic is to take the most positive entries in each column. We define the topic coherence metric NPMI [37] in Equation 3 as follows:

$$npmi(T_i) = \sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{\log \frac{P(T_{w_i}, T_{w_j})}{P(T_{w_i})P(T_{w_j})}}{-\log P(T_{w_i}, T_{w_j})} \tag{3}$$

where T_{w_i} and T_{w_j} are the topic word i and j in the sets of filtered topics. w is the list of top-N words for a topic. For a model generating m topics, the overall npmi score is an average over all topics. However, since we incorporate the coherence score into the training phase, we consider the coherence of each topic separately. Thus, top-N word of each topic T_i will get a coherence score. This score will be compared with the *mean* of scores, we refer to it as θ, and the topics that have coherence score less than the mean will get inactivated during training phase. This process helps in eliminating topics that may not lead to a coherent topic.

$$\theta = \sum_{i=1}^{m} \frac{npmi(T_i)}{m} \tag{4}$$

where m is number of topics. Thus, we compare each $npmi(T_i)$ with θ and select those that meet our condition; having coherence greater or equal the average of coherence across all topics.

Algorithm 1: Approach of Training Phase

procedure *Training Phase*:
  for *e in epochs* **do**
    $z = tanh(Wx + b)$
    $H = cscat\_layer(z)$
    $H = scat\_layer(k, H)$
    $\hat{z} = power\_aggregation(z, H)$
    $\hat{x} = sigmoid(W^T\hat{z} + c)$
    $loss = criterion(x, \hat{x})$
    $back\_propagation(W, W^T, loss)$

function *cscat_layer(z)*:
  for *each neuron in z* **do**
    $H \leftarrow \{neuron \mid npmi(neuron) > \theta\}$

function *scat_layer(k, H)*:
  s_p = get_strongest_positive($k/4, H$)
  s_n = get_strongest_negative($k/4, H$)
  w_p = get_weakest_positive($k/4, H$)
  w_n = get_weakest_negative($k/4, H$)
  $H \leftarrow [s_p, s_n, w_p, w_n]$

function *power_aggregation(z, H)*:
  for *each neuron* $\in z$ *and* $\notin H$ **do**
    total_energy += E(neuron)
    E(neuron) = 0
  for *each neuron* $\in H$ **do**
    E(neuron) += total_energy

Table 1
The datasets included in our experiments

Dataset	20 Newsgroups	Reuters	Wiki10+
Train size	11314	554414	19972
Test size	7532	250000	1972
Validation size	1000	10000	1000
Vocabulary size	2000	5000	2000

Table 2
Notations

Notation	Description
m	Number of dimension (topics)
k	Number of winner neurons
z_s	Set of strongest activations
z_w	Set of weakest activations
n	Number of highest activations per topic
E	Energy of the activations

3.3. Selecting K-Competitive Neurons

After filtering the neurons using their coherence scores obtained in the previous step, we select the top strongest and weakest, positive and negative activations per dimension m among the eligible vectors. The selected top k neurons, referred to as winners, will gain the activation values of the loser neurons.

In particular, we select $k/2$ top strongest activation values (positive and negative) and $k/2$ weakest activation values (positive and negative) neurons. Selecting the neurons with the weakest activations in our approach plays a critical role in identifying features that otherwise are buried in weak signals. This is mainly supported by the fact that weak activation values might be caused by, especially in early training epochs: (1) initialization randomness and (2) representing rare (less frequent) words with small values in the vector space. To ensure that weakest activations have a real potential to become representative features, we track their behavior over training cycles and only keep the neurons that illustrate improvement over time. Otherwise, they are removed from the competition process. For example, lets assume that $z_w^p = \{z_{w_1}, z_{w_2}, \ldots, z_{w_k}\}$ is the set of weakest selected neurons in the previous iteration. We will re-evaluate the values of these activations, after being considered winners and aggregated new power, to only keep those that grow in power during subsequent iterations, as follows:

$$|z_{w_i}^p| \leq |z_{w_i}^{p+1}| \; for \; i \in m \tag{5}$$

3.4. Neuron Power Aggregation

After winner neurons are selected, they add the total activation values from all loser neurons to their current activation value (we refer to this process as neuron power aggregation). Loser neurons are then inactivated (i.e., assigned the value 0). Algorithm 1, *power_aggregation* function defines this step; where it first calculates the total energy of the loser neurons, assign them to 0, and finally adds this total energy to the winner neurons. Note that the base case scenarios are not included in the algorithm for simplicity.

4. Experiments

We evaluate the performance of CSCAT on several tasks compared to the current state-of-the-art models. First, we briefly discuss the used datasets and the relevant baseline models. All experiments were performed using Nvidia Titan RTX GPU with 64G RAM. We implemented our models using Keras version 2.2.4 [38] with TensorFlow 1.13 backend [39]. We used an internal model management tool, called ModelKB, to keep track of our experiments [40, 41]. We used three datasets in our experiments: 20 Newsgroups [42], Reuters [43], and Wiki10+ [44]. The details about the datasets are listed in Table 1.

4.1. Baseline Models

The results of our CSCAT model are compared to the following models:

Table 3
Document Classification Results

Model	20 Newsgroups			Wiki10+			Reuters		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
LDA [21]	0.42	0.50	0.46	0.72	0.45	0.56	0.70	0.49	0.44
k-sparse [19]	0.42	0.42	0.42	0.72	0.45	0.56	0.80	0.50	0.62
NVCTM [22]	0.57	0.56	0.57	-	-	-	-	-	-
KATE [11]	0.70	0.70	0.70	**0.73**	0.45	0.56	0.70	0.49	**0.61**
ProdLDA [11]	0.53	0.53	0.53	0.49	0.45	0.47	0.51	0.56	0.47
CSCAT (ours)	**0.71**	**0.71**	**0.71**	0.61	**0.59**	**0.62**	**0.81**	**0.52**	**0.61**

1. LDA [21]: a probabilistic topic model that uses the bag-of-words technique to model a topic and a mixture of topics to model a document.
2. K-sparse [19]: an autoencoder that enforces sparsity in the hidden layers by keeping the k highest activities in the training phase and the αk highest activities in the testing phase. k-sparse uses linear activation functions, while the non-linearity in the model derives from the selection of k highest activities.
3. NVCTM [22]: a novel model that proposes the idea of centralized transformation flow to capture the correlations among topics by reshaping topic distributions. The implementation of this model is not available, so we compared our results to the results reported in their original paper.
4. KATE [11]: a shallow autoencoder model with a competitive hidden layer selects the k largest positive neurons and largest absolute negative neurons. Moreover, KATE uses an additional hyperparameter α to amplify the energy value.
5. ProdLDA [30]: a new topic model that replaces the mixture model in LDA with a product of expert.

4.2. Quantitative Analysis

In this section, we analyze the performance of CSCAT compared to the models mentioned above on two tasks: multi-class classification using the dataset of 20 Newsgroups and multi-label classification using the Wiki10+ and Reuters datasets. The results of both tasks are reported in Table 3. We also compare and report the topic coherence scores of these models.

4.2.1. Multi-class classification

This task included training a simple softmax multi-class classifier with a cross-entropy loss function on the 20 Newsgroups dataset. The classification precision, recall, and F1 scores are listed under the 20 Newsgroups column in Table 3. We set the number of topics to 50, and n (number of highest positive activations to consider for the coherence comparison among topic vectors) for all the experiments is set to 15. Changing n from 10 to 50 had little differences in the model's performance, so we kept $n = 15$, which achieved best results. Also, note that we did not run the experiment on the NVCTM model, rather we obtained these results from its research paper.

Table 4
Coherence Score Evaluation Results

Model	T = 25	T = 50
LDA [21]	0.112	0.140
k-sparse [19]	0.093	0.090
NVCTM [22]	**0.180**	**0.176**
KATE [11]	0.073	0.101
ProdLDA [11]	0.251	0.240
CSCAT (ours)	0.151	0.118

It is obvious from the table that competition-based autoencoders achieve better results than conventional models, such as LDA. For example, KATE achieves 70% for all three measurements outperforming NVCTM, K-sparse, and LDA, but CSCAT outperform all listed models achieving 0.71 scores on all three measures.

4.2.2. Multi-label classification:

we implemented a multi-label logistic regression classifier with a cross-entropy loss function to evaluate the models on the multi-label classification task using Wiki10+ and Reuters datasets. The precision, recall, and F1 scores of these experiments are also listed in Table 3. Note that due to the missing implementation of the NVCTM model, we could not reproduce the results reported in the original paper.

We observe from the table that there are some inconsistencies among the results of this task. We believe that this is because those two datasets, i.e., Wiki10+ and Reuters, are highly-imbalanced. Thus, there exist some differences among the precision on the one hand and the recall on the other hand. KATE wins the precision accuracy in the Wiki10+ task while CSCAT wins both the recall and F1 scores. We also observe that CSCAT significantly outperform the rest of the models.

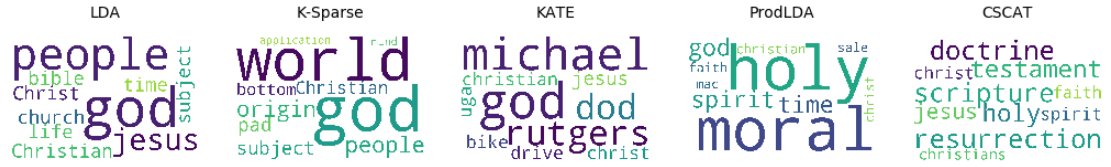

Figure 3: Topic visualization: Religion

Figure 4: Topic visualization: Politics

4.2.3. Topic Coherence

We used a topic coherence measurement that is known to have a human-level judgment, called Normalized Pointwise Mutual Information (NPMI) [45]. We evaluated NPMI across all the models using the 20 Newsgroups. We extracted the top-10 words per topic and then computed the NPMI scores as illustrated in Equation (6), using topic numbers, $T = 25, 50$.

The results of the NPMI are illustrated in Table 4. We notice that CSCAT achieves scores of 0.151 for 25 topics and 0.118 for 50 topics compared to NVCTM, which scores of 0.180 and 0.176 for 25 and 50 topics, consecutively. However, this higher coherence score in NVCTM comes with a lower classification accuracy compared to both SCAT and CSCAT, as explained in the previous subsection in addition to lower performance at the document visualization task, as we explain in the following subsection. Overall, CSCAT achieves the second-best coherence score results among the results of the models.

$$npmi(N) = \sum_{j=2}^{N} \sum_{i=1}^{j-1} \frac{\log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-\log P(w_i,w_j)} \tag{6}$$

4.3. Qualitative Analysis

In this section, we illustrate that our models can learn semantically meaningful representations from textual data. We compare our results to the baseline models, including LDA, K-sparse, and KATE and ProdLDA using the 20 Newsgroups dataset, with the number of topics set equal to 25. The results are listed in figure 3, 4 and 5 for religion, politics and sports.

We can observe from the figures that our CSCAT model generates the most semantically meaningful topics. Here, we show three topics. The top 10 words learned from the *Religion* category: “resurrection”, “doctrine”, “scripture”, “testament”, “holy”, “jesus”, “spirit”, “christ” and “faith” are strongly related to Religion. CSCAT also learns meaningful representation for the *Sport* category, including words like “players”, “baseball”, “playoffs”, “leafs”,

Figure 5: Topic visualization: Sport

"scoring", "league", and "scored" and under *Politics* topic words like "congress", "senate", "clinton", "president", "secretary", "administration" which represent the most meaningful representations among the rest of the words generated by other models.

5. Conclusions

CSCAT is a new autoencoder for textual data based on the concept of competitive learning, in which only k neurons of the bottleneck layer engage in the learning process while the rest are inactivated. Those *winning* neurons become highly specialized in learning specific properties as a result of the competition. Unlike prior techniques that introduced competition between the strongest positive and negative neurons, our method removes extremely incoherent neurons first and then adds a competition for the highest and lowest positive and negative neurons in the autoencoder's bottleneck layer.

Our thorough experiments showed that our method delivers very close or higher performance on a variety of textual data applications, such as classification and topic modeling. Furthermore, compared to the baseline models we examined in this paper, our model returns more semantically meaningful topics. Our approach can also be used to reduce the dimensionality of textual data.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436.

[2] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, J. Walsh, Deep learning vs. traditional computer vision, in: Science and Information Conference, Springer, 2019, pp. 128–144.

[3] Z. Zhang, J. Geiger, J. . Pohjalainen, A. E.-D. Mousa, W. Jin, B. Schuller, Deep learning for environmentally robust speech recognition: An overview of recent developments, ACM Transactions on Intelligent Systems and Technology (TIST) 9 (2018) 1–28.

[4] J. Liu, W.-C. Chang, Y. Wu, Y. Yang, Deep learning for extreme multi-label text classification, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 115–124.

[5] X. Wei, W. B. Croft, Lda-based document models for ad-hoc retrieval, in: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006, pp. 178–185.

[6] S. Goudarzvand, J. S. Sauver, M. M. Mielke, P. Y. Takahashi, Y. Lee, S. Sohn, Early temporal characteristics of elderly patient cognitive impairment in electronic health records, BMC medical informatics and decision making 19 (2019) 149.

[7] S. Goudarzvand, J. S. Sauver, M. M. Mielke, P. Y. Takahashi, S. Sohn, Analyzing early signals of older adult cognitive impairment in electronic health records, in: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2018, pp. 1636–1640.

[8] M. Hosseini, A. S. Maida, M. Hosseini, G. Raju, Inception lstm for next-frame video prediction (student abstract), in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 13809–13810.

[9] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Advances in neural information processing systems, 2007, pp. 153–160.

[10] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[11] Y. Chen, M. J. Zaki, Kate: K-competitive autoencoder for text, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 85–94.

[12] F. Huszar, L. Theis, W. Shi, A. Cunningham, Lossy image compression with compressive autoencoders (2020).

[13] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of machine learning research 11 (2010) 3371–3408.

[14] S. Zhai, Z. M. Zhang, Semisupervised autoencoder for sentiment analysis, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[15] H. Larochelle, S. Lauly, A neural autoregressive topic model, in: Advances in Neural Information Processing Systems, 2012, pp. 2708–2716.

[16] L. Maaloe, M. Arngren, O. Winther, Deep belief nets for topic modeling, arXiv preprint arXiv:1501.04325 (2015).

[17] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: International conference on machine learning, 2016, pp. 1727–1736.

[18] C. Zhang, J. Butepage, H. Kjellstrom, S. Mandt, Advances in variational inference, IEEE transactions on pattern analysis and machine intelligence (2018).

[19] A. Makhzani, B. Frey, K-sparse autoencoders, arXiv preprint arXiv:1312.5663 (2013).

[20] S. Goudarzvand, G. Gharibi, Y. Lee, Scat: Second chance autoencoder for textual data, arXiv preprint arXiv:2005.06632 (2020).

[21] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of machine Learning research 3 (2003) 993–1022.

[22] L. Liu, H. Huang, Y. Gao, Y. Zhang, X. Wei, Neu-

ral variational correlated topic modeling, in: The World Wide Web Conference, ACM, 2019, pp. 1142–1152.
[23] D. M. Blei, T. L. Griffiths, M. I. Jordan, The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, Journal of the ACM (JACM) 57 (2010) 7.
[24] J. Zhu, E. P. Xing, Sparse topical coding, arXiv preprint arXiv:1202.3778 (2012).
[25] J. Eisenstein, A. Ahmed, E. P. Xing, Sparse additive generative models of text (2011).
[26] K. Canini, L. Shi, T. Griffiths, Online inference of topics with latent dirichlet allocation, in: Artificial Intelligence and Statistics, 2009, pp. 65–72.
[27] M. Bahrani, H. Sameti, A new bigram-plsa language model for speech recognition, EURASIP Journal on Advances in Signal Processing 2010 (2010) 308437.
[28] J. Schneider, M. Vlachos, Topic modeling based on keywords and context, in: Proceedings of the 2018 SIAM International Conference on Data Mining, SIAM, 2018, pp. 369–377.
[29] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Machine learning 42 (2001) 177–196.
[30] A. Srivastava, C. Sutton, Autoencoding variational inference for topic models, arXiv preprint arXiv:1703.01488 (2017).
[31] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, arXiv preprint arXiv:1511.05644 (2015).
[32] A. Makhzani, B. J. Frey, Winner-take-all autoencoders, Advances in neural information processing systems 28 (2015) 2791–2799.
[33] Y. Bengio, Learning deep architectures for AI, Now Publishers Inc, 2009.
[34] A. Mingorance-Le Meur, Jnk gives axons a second chance, Journal of Neuroscience 26 (2006) 12104–12105.
[35] H. Jiang, Y. Rao, Axon formation: fate versus growth, Nature neuroscience 8 (2005) 544–546.
[36] G. Bouma, Normalized (pointwise) mutual information in collocation extraction, Proceedings of GSCL 30 (2009) 31–40.
[37] N. Aletras, M. Stevenson, Evaluating topic coherence using distributional semantics, in: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers, 2013, pp. 13–22.
[38] F. Chollet, et al., Keras, 2015. URL: https://github.com/fchollet/keras.
[39] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: https://www.tensorflow.org/, software available from tensorflow.org.
[40] G. Gharibi, V. Walunj, R. Alanazi, S. Rella, Y. Lee, Automated management of deep learning experiments, in: Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning, 2019, pp. 1–4.
[41] G. Gharibi, V. Walunj, R. Nekadi, R. Marri, Y. Lee, Automated end-to-end management of the modeling lifecycle in deep learning, Empirical Software Engineering 26 (2021) 1–33.
[42] K. Lang, Newsweeder: Learning to filter netnews, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 331–339.
[43] D. D. Lewis, Y. Yang, T. G. Rose, F. Li, Rcv1: A new benchmark collection for text categorization research, Journal of machine learning research 5 (2004) 361–397.
[44] A. Zubiaga, Enhancing navigation on wikipedia with social tags, arXiv preprint arXiv:1202.5469 (2012).
[45] J. H. Lau, D. Newman, T. Baldwin, Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 530–539.