Proceedings Paper:
Jiang, Y., Wang, Y., Song, X. et al. (2020) Comparing topic-aware neural networks for bias detection of news. In: De Giacomo, G., Catala, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A. and Lang, J., (eds.) Proceedings of 24th European Conference on Artificial Intelligence (ECAI 2020). ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 Aug - 02 Sep 2020, Santiago de Compostela, Spain. Frontiers in Artificial Intelligence and Applications, 325. International Joint Conferences on Artificial Intelligence (IJCAI), pp. 2054-2061. ISBN: 9781643681009. ISSN: 0922-6389. EISSN: 1879-8314.

https://doi.org/10.3233/FAIA200327

ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200327

Comparing Topic-Aware Neural Networks for Bias Detection of News

Ye Jiang[1], Yimin Wang[2], Xingyi Song[1] Diana Maynard[1]
Department of Computer Science[1]
School of Mathematics and Statistics[2]
University of Sheffield
Sheffield , UK
{yjiang18, yimin.wang, x.song, d.maynard}@sheffield.ac.uk

Abstract. The commercial pressure on media has increasingly dominated the institutional rules of news media, and consequently, more and more sensational and dramatized frames and biases are in evidence in newspaper articles. Increased bias in the news media, which can result in misunderstanding and misuse of facts, leads to polarized opinions which can heavily influence the perspectives of the reader. This paper investigates learning models for detecting bias in the news. First, we look at incorporating into the models Latent Dirichlet Allocation (LDA) distributions which could enrich the feature space by adding word co-occurrence distribution and local topic probability in each document. In our proposed models, the LDA distributions are regarded as additive features on the sentence level and document level respectively. Second, we compare the performance of different popular neural network architectures incorporating these LDA distributions on a hyperpartisan newspaper article detection task. Preliminary experiment results show that the hierarchical models benefit more than non-hierarchical models when incorporating LDA features, and the former also outperform the latter.

1 INTRODUCTION

News media typically present biased accounts of news stories, and different ideologies might be presented by different news publications. Detecting bias in the news articles is essential to journalists and researchers for understanding how the presented news stories reflect opinions and attitudes which can heavily influence the readers' perspectives [30]. A growing number of people are consuming biased news, since the hyperpartisan [28] framing style, which exhibits extreme bias, is particularly prone to widespread dissemination on social media. This kind of content has also been identified as a source of increased polarization among the public [23], and consequently leads to further biases in selecting content and in the overall tone of news reporting [16]. Such bias in the news media tends to result in misunderstanding and misuse of facts. Not only is this a factor in swaying individuals' voting preferences [10], but has also even led to ethnic violence [25].

Traditionally, methods such as Latent Semantic Analysis [6], probabilistic Latent Semantic Analysis (pLSA)[11] and Latent Dirichlet Allocation (LDA) [2] have been implemented to infer the semantic meaning of documents through a set of topic representations. Such representations convert text into vector representation which make it feasible for machines to "understand" the semantics of text for tasks such as document summarization [31], document classification [21] and clustering [13]. Methods based on Bag-of-Words (BoW) are frequently used to calculate the statistical features in the document collection. These transform the text data into numeric data that enables a large set of documents to be automatically structured, explored, grouped or clustered based on the word occurrences. However, such document representations suffer from dimensional sparsity, and BoW-based models ignore the contextual information in the text [40], i.e., the relationship between a target word and its surrounding words.

Recently, neural network-based models, which have been proposed in order to generate low-dimensional vector representations, and which are also able to capture semantic word relationships, have been found to outperform most BoW-based models [22, 35]. For instance, the Continuous Bag of Words (C-BoW) model [24] encodes each word into a fixed length vector representation based on other words surrounding the target word. However, such models suffer from the disadvantage that they do not utilize the word co-occurrence of the entire corpus. Specifically, they only scan the textual information within a local context window, which fails to make use of statistical information of the whole corpus. GloVe [27] attempts to resolve this by implementing both global matrix factorization and local content window-based methods; however, our proposal uses a different approach that combines the global co-occurrence information with semantic features of local content windows. Another problem is that many neural network models [39, 5] ignore the hierarchical features of a document, such as the structural relationship between word and sentence, or sentence and document. In an attempt to resolve these issues, we propose a combination of hierarchical frameworks that capture structural features on both word and sentence level, and also incorporate LDA distributions on each level separately.

In order to evaluate the proposed topic-aware hierarchical document representation, we implement a document classification task based on the publicly accessible dataset from the Hyperpartisan News Detection Task.[1] The documents in this corpus are by nature more challenging for learning models than those typically used for traditional document classification (e.g., IMDB, Amazon reviews) for a number of reasons. First, the documents in the hyperpartisan corpus have widely varying length. This means that either the maximum sequence length must be used to fully represent the longest

[1] https://pan.webis.de/semeval19/semeval19-web/index.html

document, which causes a high computational cost, or alternatively a significant information loss will be incurred if the sequence length is restricted to a manageable number of initial tokens from the document. Second, partisanship is more complex than aspects like sentiment to discover, so the learning models require complex text representation to fully capture the subtle semantics.

We perform an evaluation by comparing different popular neural network architectures, with and without incorporating LDA-based distributions, and also compare these with non-hierarchical structures. The code of the proposed model LDA-HAN[2] is available for replicability. Theoretically, the models incorporating LDA distributions should enrich the feature space by adding co-occurrence statistics features and local topic probability distribution on the word and sentence level respectively. Our experimental results demonstrate that the proposed topic-aware document representation outperforms traditional ones, and also that the inclusion of the LDA features has greater impact on the hierarchical representations.

2 RELATED WORK

Traditional BoW-based approaches have often been used to classify newspaper articles. Rubin et al. [29] used a BOW representation with a Support Vector Machine (SVM) to classify satirical news articles. Fortuna et al. [7] also represented news articles in the vector space model by using Term Frequency-Inverse Document Frequency (TF-IDF) weighting, and utilized SVM to identify the bias in describing events in news articles, while Budak et al. [3] used SVM to quantify news bias in a large set of political articles. Meanwhile, LDA has been combined with traditional feature engineering-based methods in many document classification tasks. Wu et al. [36] combined LDA with SVM to classify Chinese news, outperforming the models which generate high-dimensional feature space such as TF-IDF models. Li et al. [18] implemented LDA with a softmax regression to overcome the high dimensional problems of the news text. Kim et al. [14] regarded the document-topic distribution from LDA as a document representation in which both word frequencies and semantic information are considered, to enhance the performance of document classifiers.

Recently, neural network approaches have been combined with LDA for generating document representations. Liu et al. [20] applied LDA to build topic-based word embeddings based on both words and their topics. Xu et al. [37] also implemented LDA to capture topic-based word relationships and then integrated it into distributed word embeddings. Wang and Xu [34] implemented LDA-based text features as input to a deep neural network to detect automobile insurance fraud. Narayan et al. [26] introduced a topic-aware convolutional neural network to generate summaries from online news articles. LDA was used to generate document-topic distributions and word-topic distributions separately, and a CNN was then incorporated to encode and decode the document representations.

However, such approaches generate document representations without considering the characteristics of document structure hierarchically. To address this issue, a Hierarchical Attention Network (HAN) [38] has been previously proposed, which can capture the hierarchical features on both word level and sentence level through a stacked RNN architecture. This outperformed many other baseline models, and indicates that such prior hierarchical information has the potential to enrich document representations, especially when the document sizes are in a wide range.

[2] https://github.com/yjiang18/LDA-HAN

Hierarchical models have been implemented by many natural language processing (NLP) downstream tasks. Li et al. [17] implemented a hierarchical auto-encoder on both word and sentence level, decoding each representation to reconstruct the original paragraph. Gao et al. [9] constructed a hierarchical convolutional attention model that utilized a combination of self-attention and target-attention. Abreu et al. [1] combined RNN with CNN in a hybrid hierarchical attentional neural network for the document classification task. Zheng et al. [40] compared different hierarchical encoders in documents with differing lengths, and revealed that for document classification, hierarchical frameworks outperform the corresponding neural network models without the hierarchical architecture. They also indicated that the benefits resulting from the hierarchical architecture become more significant as the document length increases. However, these approaches only consider the word embeddings as the input to the encoding layers. Founta et al. [8] utilized a wide variety of available metadata, combining them with word embeddings to enhance the model performance for the task of abusive language detection. Finally, Chen et al. [4] combined word embeddings with WordNet to obtain more relevant occurrences for each sense. Unlike the unified model, which takes different features as inputs to several models independently, our model combines the word embedding with LDA distributions as additive features to the learning model simultaneously.

3 METHODOLOGY

Hierarchical frameworks utilize the document structural features such as the relation between word and sentence, and between sentence and document. Meanwhile, the LDA model generates different distributions which can be used as additional information for encoding document representation. In order to investigate the effectiveness of a learning model which encodes documents hierarchically and incorporates LDA distributions, this paper compares different neural network structures with/without the inclusion of LDA distributions. We first establish three different neural network structures (i.e., CNN, RNN and Transformer) without considering structural features, and then compare these three networks with/without LDA distributions. We also apply two hierarchical models to evaluate the combination of structural features and LDA distributions.

3.1 LDA Distributions

The LDA model generates topic-word distribution and document-topic distribution simultaneously. The former is shared between all documents and contains global word co-occurrence features in the whole corpus, while the latter is the local distribution over the topics for a given document, and is independent of all other documents. These two distributions can be used as additional features in the word level and sentence level encoder layer in the hierarchical frameworks. Each sentence is represented by implementing a specific neural network architecture to encode the combination of word embeddings and transposed topic-word distributions. Similarly, the document is then represented by encoding all sentence representations which are generated from the previous step. Finally, the document representation is concatenated with document-topic distribution as an additional feature to make the final prediction.

3.2 Model Specifications

Let D denote a document consisting of a sequence of sentences (s_1, s_2 , ... , s_m); Meanwhile, let s_i denote a sentence consist-

ing of words ($w^1_{s_i}$, $w^2_{s_i}$,..., $w^n_{s_i}$) where $i \in [1, m]$, we embed s_i into a distributional space x = (x_1, x_2, ... , x_n) where $x_j \in \mathbb{R}^k$, $j \in [1, n]$ and k is the dimension of word embedding. Meanwhile, the LDA model generates topic-word distribution, which are transposed as $tw = (tw_1, tw_2, ..., tw_n)$ where $tw_j \in \mathbb{R}^t$ (t denotes number of topics) and the document-topic distribution can be denoted as $dt = (dt_1, dt_2, ..., dt_d)$ where $dt \in \mathbb{R}^{d \times t}$. We train all the models to minimize their cross-entropy error:

$$\ell(\tilde{y}) = \sum_{p=1}^{c} y_p \log(\tilde{y}_p) \tag{1}$$

where $y, \tilde{y}$ are the ground-truth label and predicted label respectively, c denotes number of classes.

3.2.1 LDA based Non-Hierarchical Models

Three different network structures are implemented as the encoding layers in the LDA-based non-hierarchical models. Figure 1 depicts the overall model structure. Formally, each document representation is generated from the initial tokens in the document. This is an aggregation of all the word embeddings x to the encoding layer. Meanwhile, the LDA model also takes text input to generate topic-word distribution and document-topic distribution simultaneously. Next, the transposed topic-word distribution tw is concatenated with word embeddings as the input to the encoding layer. The document-topic distribution dt is then concatenated with the generated document representation. Finally, a Fully Connected (FC) layer with softmax activation and Adam optimizer is made for the final classification.

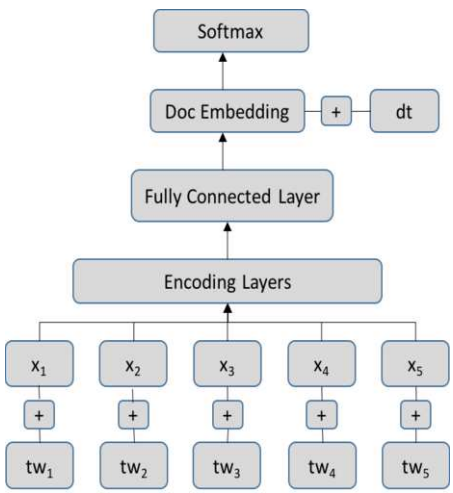

Figure 1. LDA based non-hierarchical models structure

CNN: For a possible variant CNN structure, Kim's implementation [15] is adopted as the baseline CNN model. It consists of 128 filters and 3 different convolutional filter sizes $h \in$ [2,3,4] with ReLU activation, with each convolutional layer followed by a max-pooling layer. The results from the max-pooling layers are concatenated, going through a Fully Connected (FC) layer with 50 hidden units. Formally, the convolutional layer using different filter operators $W_{h,j} \in \mathbb{R}^{h \times k}$ is applied to a window of h words to produce a new feature c^h_j at the word level:

$$c^h_j = ReLU\left((x_{j:j+h-1} \oplus tw_{j:j+h-1}) \circ W_{h,j} + b_{h,j}\right) \tag{2}$$

where the notation $\circ$ and $\oplus$ denote element-wise multiplication and concatenation respectively, $ReLU$ denotes the nonlinear function, $b_{h,j}$ is a bias term. Then, the max-over-time pooling function is used to capture the most important feature $\tilde{c}^h_j$:

$$\tilde{c}^h_j = Max\left(c^h_j\right) \tag{3}$$

The final feature maps are formed by concatenating all $c_j = (\tilde{c}^1_j, \tilde{c}^2_j, ..., \tilde{c}^h_j)$, then the document representation d can be generated by a FC layer:

$$d = ReLU\left(c_j \circ W_j + b_j\right) \tag{4}$$

where W_j is a weight matrix and b_j is a bias term. Finally, the document representation d is concatenated with dt to make the final prediction in a softmax layer.

Self-Attentive RNN: We apply self-Attentive LSTM [19] as the baseline RNN model. It consists of two LSTMs with 50 hidden units and a dropout of probability 0.2 in each direction. In addition, the self-attention layer has 100 hidden units for the outputs from LSTM, and is then followed by an FC layer with 32 hidden units and ReLU non-linearity.

Formally, the forward $\overrightarrow{r_n}$ and backward $\overleftarrow{r_n}$ hidden states at the word level can be obtained by using bidirectional LSTM:

$$\overrightarrow{r_n} = \overrightarrow{LSTM}\left(x_{1:n} \oplus tw_{1:n}\right) \tag{5}$$

$$\overleftarrow{r_n} = \overleftarrow{LSTM}\left(x_{1:n} \oplus tw_{1:n}\right) \tag{6}$$

Then the $\overrightarrow{r_n}$ and $\overleftarrow{r_n}$ can be concatenated as $r_n = (\overrightarrow{r_n}; \overleftarrow{r_n})$, thus each document is encoded as $\tilde{r}_n = (r_1, r_2, ..., r_n)$ where $\tilde{r}_n \in \mathbb{R}^{n \times 2u}$ (u is the hidden unit for each unidirectional LSTM), which is then passed to attention mechanism to get annotation matrix α_n:

$$\alpha_n = softmax\left(W_{s2} Tanh(W_{s1} \tilde{r}^T_n)\right) \tag{7}$$

where $W_{s1} \in \mathbb{R}^{p \times 2u}, W_{s2} \in \mathbb{R}^{l \times p}$ (p is the number of neuron units, l denotes to use l times attention) are parameters to learn the important components of the document. The annotation matrix $\alpha_n \in \mathbb{R}^{l \times n}$ multiply $\tilde{r}_n$ to compute the l weighted sums to get the final document representation d.

$$d = \sum_n \alpha_n \tilde{r}_n \tag{8}$$

Finally, the document representation d is concatenated with dt to make the prediction in a softmax layer.

Transformer: We implement the encoder part of Transformer [32] to evaluate its performance on the document classification task. We first calculate the Positional Embeddings (PE) with 300 dimensions for the input, and sum the PE with the original word embeddings instead of concatenation. For the multi-head self-attention, we use a total of eight heads, where each head has 16 units. We then take the average of each step of the output sequence from the self-attention layer, followed by an FC layer with 32 hidden units and ReLU non-linearity. Formally, we use the scaled-dot-product attention to compute the most pertinent information to that document:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{k}})V \tag{9}$$

where Q, K, V are 'query', 'key' and 'value' embeddings which concatenate word embeddings x_n with word-topic distribution wt_n. Thus, the final document representation can be formed by multihead attention:

$$\begin{aligned} Multihead(Q, K, V) &= [head_1, head_2, ..., head_n] \\ where \quad head_n &= Attention(Q_j, K_j, V_j) \end{aligned} \tag{10}$$

The final output is the concatenation of the outputs from each head, which is then concatenated with dt to make the final prediction in a softmax layer.

3.2.2 LDA based Hierarchical Models

In this section, we utilize two different hierarchical models to investigate the document representation with/without the LDA features. Figure 2 depicts the overall hierarchical framework structure. The hierarchical models take word and sentence representation as inputs at different phases. The word-topic distribution tw is concatenated with word embeddings x, and is aggregated to a sentence representation to the encoding layer. The document representation can then be formed by aggregating all the sentence representations s. The document-topic distribution dt is concatenated with the generated document representation. An FC layer with softmax activation and Adam optimizer is used for the final classification.

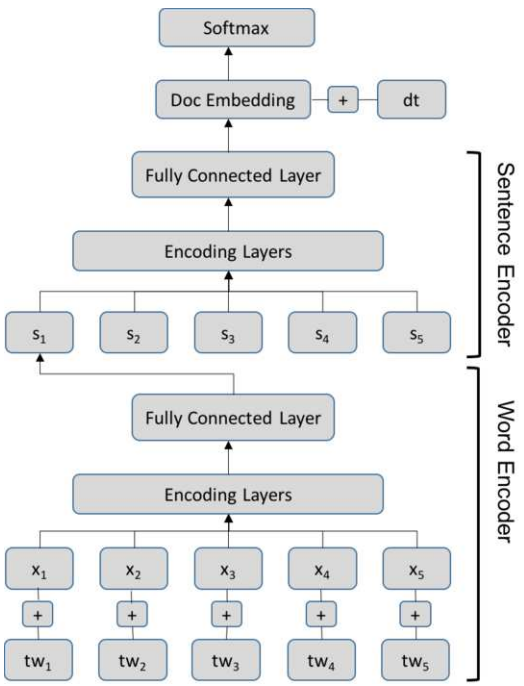

Figure 2. Hierarchical model structure

ESRC: We implemented a similar structure to the ELMo Sentence Representation Convolutional Network (ESRC) [12] for the hierarchical Convolutional framework, but using the pre-trained GloVe embeddings instead of the ELMo embeddings in order to compare them with other hierarchical models. Formally, the word encoder has 128 filters and 7 different convolutional filter sizes $h \in$ [1,2,3,4,5,6,7] with ReLU activation, followed by a batch normalization and a max-pooling layer. The results from the max-pooling layers are concatenated and passed to an FC layer with 32 hidden units and ReLU activation to form a sentence representation. The sentence encoder takes each sentence representation as the input, with the same structure as the word encoder. Similar to Kim's CNN, the encoding convolutional layer using different filter operators $W_{h,j} \in \mathbb{R}^{h \times k}$ is applied to a window of h words to produce a new feature $c^h_{x_j}$ at the word level:

$$c^h_{x_j} = BN\left(ReLU\left((x_{j:j+h-1} \oplus tw_{j:j+h-1}) \circ W_{h,j} + b_{h,j}\right)\right) \tag{11}$$

where the notation $\circ$ and $\oplus$ denote the element-wise multiplication and the concatenation respectively, $ReLU$ denotes the nonlinear function, $b_{h,j}$ is a bias term; we also add a batch normalization BN on top of the convolutional layer.

Then, the max-over-time pooling function is used to capture the most important feature $\tilde{c}^h_{x_j}$:

$$\tilde{c}^h_{x_j} = Max\left(c^h_{x_j}\right) \tag{12}$$

The final word-level feature maps are formed by concatenating all $c_{x_j} = (\tilde{c}^1_{x_j}, \tilde{c}^2_{x_j}, ..., \tilde{c}^h_{x_j})$, then the sentence representation s_i can be generated by an FC layer:

$$s_i = ReLU\left(c_{x_j} \circ W_j + b_j\right) \tag{13}$$

where W_j is a weight matrix and b_j is a bias term. Then, the final document representation d can be obtained similarly: we first obtain sentence-level feature maps $c^h_{s_i}$ by convoluting the sentence sequence using different filter operators, followed by batch normalization:

$$c^h_{s_i} = \left(c^h_1, c^h_2, ..., c^h_{s_{i:i+h-1}}\right) \tag{14}$$

Then, the max pooled features can be obtained:

$$\tilde{c}^h_{s_i} = Max\left(c^h_{s_i}\right) \tag{15}$$

Finally, after concatenating all $\tilde{c}^h_{s_i}$ to obtain c_{s_i} the document representation d can be formed as:

$$d = ReLU\left(c_{s_i} \circ W_i + b_i\right) \tag{16}$$

where W_i is a weight matrix and b_i is a bias term, $ReLU$ is the non-linear function. Finally, the document representation d is concatenated with dt_i to make final predictions in a softmax layer.

HAN: We implement the Hierarchical Attention Network [38] for the hierarchical RNN framework. The word-encoder Bi-LSTM has 100 dimensional hidden units with a dropout of probability 0.2. The sentence encoder has the same structure as the word encoder, except that it has an extra FC layer with 32 hidden units and ReLU nonlinearity.

Formally, the forward $\overrightarrow{r_{x_n}}$ and backward $\overleftarrow{r_{x_n}}$ hidden states at the word level can be obtained by using bi-directional LSTM:

$$\overrightarrow{r_{x_n}} = \overrightarrow{LSTM}\left(x_{1:n} \oplus tw_{1:n}\right) \tag{17}$$

$$\overleftarrow{r_{x_n}} = \overleftarrow{LSTM}\left(x_{1:n} \oplus tw_{1:n}\right) \tag{18}$$

Then the $\overrightarrow{r_{x_n}}$ and $\overleftarrow{r_{x_n}}$ can be concatenated as $r_{x_n} = (\overrightarrow{r_{x_n}}; \overleftarrow{r_{x_n}})$. Together with attention matrix α_n, they are used to calculate the importance of each word. The sentence representation s_m is formed by

$$\alpha_n = softmax(W_{n2} tanh(W_{n1} \circ r_{x_n})) \tag{19}$$

$$s_m = \sum_n \alpha_n r_{x_n} \tag{20}$$

where W_{n1}, W_{n2} denotes the context vector jointly learning the importance of each word in the sentence. Similarly, the document representation d can be also formed by:

$$\overrightarrow{r_{s_m}} = \overrightarrow{LSTM}\left(s_{1:m}\right) \tag{21}$$

$$\overleftarrow{r_{s_m}} = \overleftarrow{LSTM}\,(s_{1:m}) \tag{22}$$

Then the $\overrightarrow{r_{s_m}}$ and $\overleftarrow{r_{s_m}}$ can be concatenated as $r_{s_m} = (\overrightarrow{r_{s_m}}; \overleftarrow{r_{s_m}})$. Together with attention matrix α_m, they are used to calculate the importance of each sentence. The document representation d is formed by

$$\alpha_m = softmax(W_{m2} tanh(W_{m1} \circ r_{s_m})) \tag{23}$$

$$d = \sum_m \alpha_m r_{s_m} \tag{24}$$

where W_{m1}, W_{m2} denotes the context vector jointly learning the importance of each sentence in the document. The document representation d is concatenated with dt to make final predictions in a softmax layer.

4 EXPERIMENTS

We split the dataset into training, evaluation and test sets with a ratio of 8:1:1. We perform 10-fold cross-validation on the training set, then fine-tune and obtain the best performing model based on the evaluation set. The final scores are obtained based on the average of 5 predictions on the test set.

4.1 Dataset

The Hyperpartisan News Detection dataset[3] contains two parts. The *By-Publisher* corpus contains 750K articles which were automatically classified, based on a categorization of the political bias of the news provider. The *By-Articles* corpus contains 1,273 articles which were annotated manually. Although the *By-Publisher* corpus has great potential in training deep learning models due to its significant size, a previous study [12] revealed that there is no significant correlation between the two corpora, in the sense that training a learning model on the *By-Publisher* corpus leads to low performance in the task of predicting partisanship on the *By-Article* corpus. Thus, in this paper all models are only trained on the *By-Article* corpus, as this is more reliable based on its manual annotation assessment [33], and it is also the official ranking corpus for the task. This paper only uses the training set (645 articles) of the *By-Article* corpus, as the rest (628 articles) of the corpus is unavailable to the public (only used for system evaluation). We calculate statistics of `By-Article` as shown in Table 1, and the document length distribution as shown in Figure 3.

Dataset	Hyperpartisan By-Article set
No. of classes	2
No. of documents	645
No. of average sentences/document	31.17
No. of maximum sentences in document	257
No. of average words/sentence	121.13
No. of maximum words in document	5906
No. of average words/document	615.99
No. of words in vocabulary	26135

Table 1. Statistics of dataset

As discussed previously, such a large differentiation in document size makes it impractical to directly use word-level representations as the input, as most news articles have no limitation on sequence length compared to other types of sources (e.g., reviews, tweets, etc). In order to calculate the compromise between representing a summary of the article and as much of its full content as possible, we use the initial 512 tokens to represent each article in the LDA-based non-hierarchical models. For the hierarchical models, we take a maximum of 100 words per sentence, and 30 sentences per document.

4.2 Preprocessing

We extract the title and article text from the original XML file, and represent each article as a sequence of sentences. The text paragraphs are split into sentences, and white spaces are normalized. We used the pre-trained GloVe model[4] to generate word embeddings, and the Gensim LDA model with 425 topics to generate topic-word distribution and document-topic distribution. We use the coherence model to find the optimal number of topics for our LDA model, as shown in Figure 4.

4.3 Results and Discussion

The results, presented in Table 2, show that, on average, the models incorporating LDA distributions outperform the other models. Specifically, the non-hierarchical models have difficulty handling a wide range of document lengths, especially if document sequences are truncated which could potentially cause information loss. Accordingly, the use of hierarchical frameworks, which summarize the importance both on the word level and sentence level features by the corresponding encoders, leads to an improvement in accuracy. Interestingly, the accuracy of the transformer alone is higher than the transformer incorporating LDA distributions, although the transformer models are generally lower than others on accuracy. On the other hand, Attentive-RNN achieves the highest accuracy out of all the non-hierarchical models, especially when it incorporates LDA features. However, the ESRC model gets lower accuracy than most of the non-hierarchical models. The accuracy of this is, however, increased by adding LDA features, and the LDA-ESRC models are better than all the non-hierarchical models. This indicates that the hierarchical frameworks incorporating LDA distributions could improve model performance in terms of accuracy. This is also proved by the LDA-HAN model, which has better accuracy than the HAN model.

Model	Accuracy
Transformer	72.12%
LDA-Transformer	71.56%
CNN	72.95%
LDA-CNN	73.47%
Attentive-RNN	73.63%
LDA-Attentive-RNN	73.75%
ESRC	71.81%
LDA-ESRC	73.69%
HAN	75.69%
LDA-HAN	**76.52%**

Table 2. Performance comparison between models. The best model accuracy is marked in **bold**

Although we see that most of the models can be improved by adding LDA features, the hierarchical frameworks can achieve greater improvement from them. The non-hierarchical models can

[3] *https://zenodo.org/record/1489920.XcVDj9Hgrew*

[4] 6 billion words, 300 dimensions

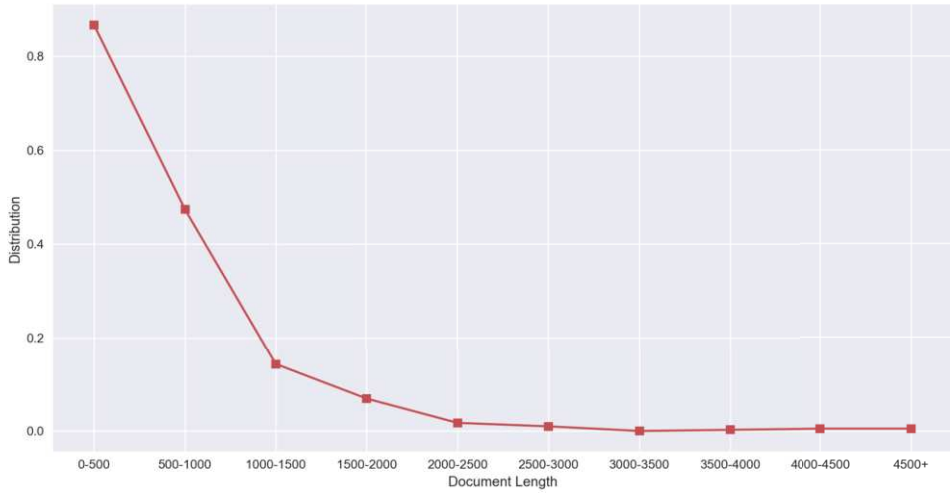

Figure 3. Document size distribution

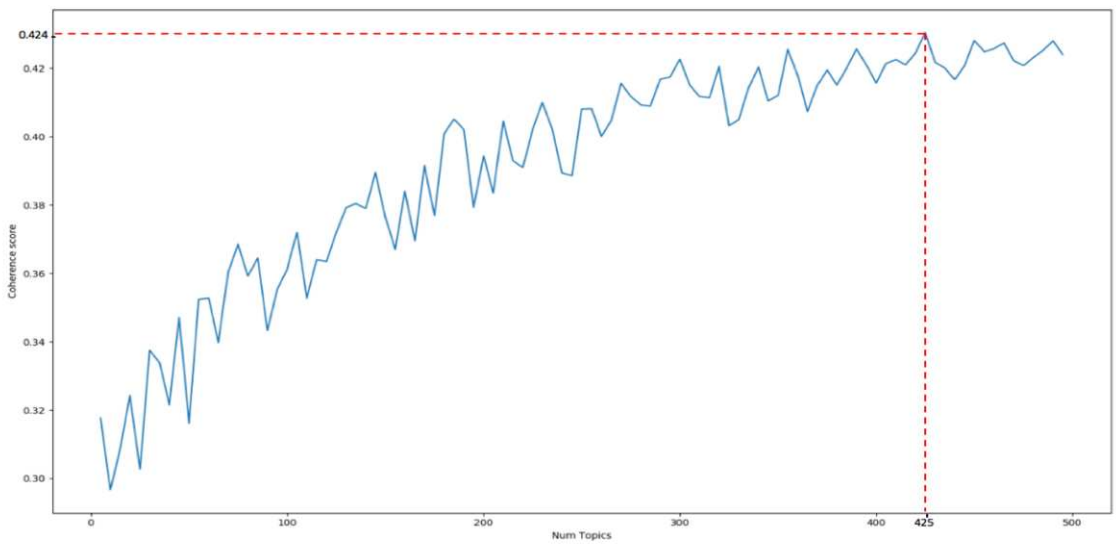

Figure 4. Coherence scores in 500 topics

achieve an improvement of around 0.32% on accuracy, while the hierarchical models can achieve around 1.36% improvement. Specifically, the hierarchical models consider both word-level and sentence-level information separately, and the topic-word distribution enriches the word-level features by adding word occurrence topic distribution through the vocabulary. On the other hand, the document-topic distribution provides local topic distribution, which is independent of all other documents, to increase feature spaces for the final softmax prediction layer, and leads to better accuracy on the document classification task.

5 CONCLUSION

In this paper, we explore the performance of different popular neural network structures with/without incorporating LDA distributions on the recently introduced Hyperpartisan News Detection dataset. This study investigates how the hierarchical models take advantage of the structural features of document to generate a better document representation compared with non-hierarchical models. Meanwhile, the models that include LDA distributions could enrich the feature space by adding global word co-occurrence topic distribution and local document topic probability on word and sentence level respectively.

We first evaluate the non-hierarchical model with/without LDA

features. The results demonstrate that most of the non-hierarchical models improved their accuracy when combined with LDA features, except for the Transformer model. On the other hand, most of the hierarchical models achieved better accuracy than non-hierarchical models, and also showed greater improvement when combined with the LDA. This indicates that the hierarchical model has the advantage of handling longer document sequences and reducing information loss by incorporating structural features in the document. Moreover, the benefits resulting from the LDA distributions can be strengthened in the hierarchical models. In conclusion, the combination of hierarchical frameworks and LDA distributions could significantly improve model performance in document classification.

ACKNOWLEDGEMENTS

Research partially supported by a Grantham Centre for Sustainable Future Scholarship, a Google Faculty Research Award 2017, and projects funded by the European Commission's Horizon 2020 research and innovation programme under grant agreements No. 654024 SoBigData and No. 825297 WeVerify.

REFERENCES

[1] J. Abreu, L. Fred, D. Macêdo, and C. Zanchettin. Hierarchical Attentional Hybrid Neural Networks for Document Classification. *arXiv preprint arXiv:1901.06610*, 2019.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[3] C. Budak, S. Goel, and J. M. Rao. Fair and balanced? quantifying media bias through crowdsourced content analysis. *Public Opinion Quarterly*, 80(S1):250–271, 2016.

[4] X. Chen, Z. Liu, and M. Sun. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, 2014.

[5] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain, Apr. 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/E17-1104`.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[7] B. Fortuna, C. Galleguillos, and N. Cristianini. Detection of bias in media outlets with statistical learning methods. In *Text Mining*, pages 57–80. Chapman and Hall/CRC, 2009.

[8] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM Conference on Web Science*, pages 105–114. ACM, 2019.

[9] S. Gao, A. Ramanathan, and G. Tourassi. Hierarchical convolutional attention networks for text classification. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2018.

[10] M. Gentzkow. Polarization in 2016. *Toulouse Network for Information Technology Whitepaper*, 2016.

[11] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[12] Y. Jiang, J. Petrak, X. Song, K. Bontcheva, and D. Maynard. Team Bertha von Suttner at SemEval-2019 Task 4: Hyperpartisan News Detection using ELMo Sentence Representation Convolutional Network. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 840–844, 2019.

[13] M. Keller and S. Bengio. Theme topic mixture model: A graphical model for document representation. In *PASCAL workshop on text mining and understanding*, number CONF, 2004.

[14] D. Kim, D. Seo, S. Cho, and P. Kang. Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec. *Information Sciences*, 477:15–29, 2019.

[15] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[16] N. Landerer. Rethinking the logics: A conceptual framework for the mediatization of politics. *Communication Theory*, 23 (3):239–258, 2013.

[17] J. Li, M.-T. Luong, and D. Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

[18] Z. Li, W. Shang, and M. Yan. News text classification model based on topic model. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pages 1–5. IEEE, 2016.

[19] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[20] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun. Topical word embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[21] Y. Lu, Q. Mei, and C. Zhai. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*, 14(2):178–203, 2011.

[22] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Won, and M. Cha. Detecting rumors from microblogs with recurrent neural networks. *Ijcai*, 2016.

[23] A. Marwick and R. Lewis. Media manipulation and disinformation online. *New York: Data & Society Research Institute*, 2017.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[25] M. R. Minar and J. Naher. Violence originated from Facebook: A case study in Bangladesh. *arXiv preprint arXiv:1804.11241*, 2018.

[26] S. Narayan, S. B. Cohen, and M. Lapata. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. *arXiv preprint arXiv:1808.08745*, 2018.

[27] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[28] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein. A stylometric inquiry into hyperpartisan and fake news. *arXiv*

preprint arXiv:1702.05638, 2017.

[29] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*, pages 7–17, 2016.

[30] A. Spence and N. Pidgeon. Framing and communicating climate change: The effects of distance and outcome frame manipulations. *Global Environmental Change*, 20(4):656–667, 2010.

[31] J. Steinberger and M. Křišťan. Lsa-based multi-document summarization. In *Proceedings of 8th International PhD Workshop on Systems and Control*, volume 7, 2007.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[33] E. Vincent and M. Mestre. Crowdsourced measure of news articles bias: Assessing contributors' reliability. In *Proceedings of the 1st Workshop on Subjectivity, Ambiguity and Disagreement in Crowdsourcing, and Short Paper Proceedings of the 1st Workshop on Disentangling the Relation Between Crowdsourcing and Bias Management (SAD 2018 and CrowdBias 2018) co-located with the 6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2018), Zürich, Switzerland, July 5, 2018*, pages 1–10, 2018. URL `http://ceur-ws.org/Vol-2276/paper1.pdf`.

[34] Y. Wang and W. Xu. Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud. *Decision Support Systems*, 105:87–95, 2018.

[35] W. Wei, X. Zhang, X. Liu, W. Chen, and T. Wang. pkudblab at SemEval-2016 Task 6 : A Specific Convolutional Neural Network System for Effective Stance Detection. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016. doi: 10.18653/v1/s16-1062.

[36] X. Wu, L. Fang, P. Wang, and N. Yu. Performance of using LDA for Chinese news text classification. In *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1260–1264. IEEE, 2015.

[37] H. Xu, M. Dong, D. Zhu, A. Kotov, A. I. Carcone, and S. Naar-King. Text classification with topic-based word embedding and convolutional neural networks. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 88–97. ACM, 2016.

[38] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*, 2016.

[39] W. Yin and H. Schütze. Multichannel variable-size convolution for sentence classification. *arXiv preprint arXiv:1603.04513*, 2016.

[40] J. Zheng, F. Cai, W. Chen, C. Feng, and H. Chen. Hierarchical neural representation for document classification. *Cognitive Computation*, 11(2):317–327, 2019.