Split-Encode-Attend Network for Document Classification

Jiapeng Wu
School of Computer Science
McGill University
Montreal, QC H3A 2A7
jiapeng.wu@mail.mcgill.ca

Abstract

In this work, we explore the use of hierarchical text structure and propose a paragraph level attention mechanism in document classification. Our contributions include: (i) we apply a paragraph level RNN module with attention mechanism on top of sentence-level representations, enabling the model to focus on important paragraphs when constructing document representations; (ii) we introduce document partitioning and design a framework in which paragraph encoder is trained using supervised learning before the hierarchical model is fine-tuned on document-label dataset. Experiments on three large datasets show that our model improve or comparable to classification results after combining with popular deep learning models.

1 Introduction

Text classification is an active research field in NLP. Traditional approaches of text classification utilize features generated from vector space, such as bag-of-words(Zhang et al., 2010), term-document frequency(TF-IDF)(Ramos and others,) and other task-specific lexical features(Louis and Nenkova, 2013b; Feng and Hirst, 2013; Thelwall et al., 2010; Qu et al., 2010). These features are incorporated using probabilistic models such as support vector machine(SVM) and multi-layer perceptron(MLP). In recent years, neural network approaches which leverage word order and embedded information have shown effectiveness in text classification, among which convolutional neural networks(CNN)(Conneau et al., 2016; Kim, 2014; Zhang et al., 2015; Johnson and Zhang, 2017) and recurrent neural networks(RNN)(Lee and Dernoncourt, 2016; Liu et al., 2016; Howard and Ruder, 2018) are widely studied.

However, while RNNs trained with stochastic gradient descent have been successfully applied to short text classification problems(dos Santos and Gatti, 2014; Wang et al., 2016; Wang et al., 2015), they have difficulty learning long-term dependencies encoded in the documents due to the vanishing gradient(Hochreiter and Schmidhuber, 1997). RNNs propagates gradients to all words in the documents sequentially, resulting in close-to-zero gradient at words far away from the end of the document.

Previous works conducted experiments using hierarchical structured neural network to derive document representations in NLP tasks(Yang et al., 2016; Tang et al., 2015; Roy et al., 2018). They regard document as a sequence of sentences thus build sentence representations independently before aggregating them to produce document representation. Inspired by RST theory(Mann and Thompson, 1988), we further exploit the hierarchical structure by treating document as a sequence of *paragraphs*, where each paragraph is composed of a cluster of semantically related sentences, or in a simpler case, a long sequence of tokens.

It is more likely that a paragraph contains significant features indicative of class labels. Sentence level polarity assessments of Yelp and IMDb dataset(Angelidis and Lapata, 2018) show that a large proportion of sentences in a document agree with the document's sentiment polarity(positive or negative). For long document classification tasks such as linguistic quality classification, paragraphs across a single document naturally represent the same writing style of the author. Additionally, due to the larger span of paragraph, more semantic and syntactic information can be encoded in paragraph representation. Compared to previous works on text classification based on sentence representations, splitting documents into paragraphs results in shorter sequence of segment hidden vectors thus reduces the length of gradient propagating path.

We test the hypothesis that paragraph partitioning and aggregation helps improve learning capacity for document classification tasks.

We first partition long documents into paragraphs each annotated with the label associated with the document it comes from, constructing a *paragraph dataset*. We train a paragraph classifier using supervised learning on the this dataset, then use it to generate representations for each paragraph. The hidden representations of all paragraphs are aggregated to build document representation, which is finally fed into a dense softmax layer for document classification.

We observe that different paragraph in a document has different degree of informativeness. Further, we acknowledge that the importance of individual paragraph is context-dependent, i.e. related to the position of each paragraph in the discourse structure. To incorporate the observed scheme into our model, we introduce attention mechanism(Bahdanau et al., 2014; Zhou et al., 2016; Xu et al., 2015) at paragraph level to make model pay dissimilar attentions to paragraphs.

We conduct sentiment analysis and linguistic quality classification experiments on three large datasets, using various neural text classification models combined with our proposed training procedure. We show that when class labels of individual paragraphs agree with their mother document, probabilistic models trained on paragraph datasets using simple relabeling scheme can produce informative representations for paragraphs, which not only improves document classification but also supports paragraph level classification. Results show that learning paragraph classifiers separately improves document classification results compared to neural network models trained directly on document-label datasets.

2 Related Work

Supervised learning methods based on bag of words and lexical features have been widely studied in previous works(Louis et al., 2012, Tang et al., 2014, Pang et al., 2002, Wang and Manning, 2012). In recent years, neural network models have achieved superior results in various text classification tasks due to their generalization capabilities without arduous feature engineering procedures(Lawrence et al., 1998). Kim(2014) proposed CNN model for sentence classification by applying convolution operators on matrix representations of input sentences. Zhang et al.(2015) explored the use of one-hot embedding of characters as input of CNN and showed improvement on text classification accuracy. While such shallow CNN models are fast to train and easy to generalize, Johnson(2017) studied the effective deepening of CNN architecture and proposed deep pyramid convolution neural network. For fast convergence, we will use word vector based shallow CNN proposed by Kim in our experiment.

Tai et al.(2015) applies tree-structured LSTM cells over syntax parse trees for classification. Li et al.(2015) explores a simple two-hierarchy bidirectional LSTM(BiLSTM) architecture(Graves and Schmidhuber, 2005), where the encodings of the coherent sub-sentences are first computed individually before fed into to phrase-level BiLSTM to generate sentence representations. This architecture outperforms both single-level BiLSTM and tree-structured LSTM models on phrase-level sentiment evaluation tasks. They deduced that sub-sentences better preserve semantic structures and errors are back-propagated to individual tokens using fewer steps in hierarchical models than in standard models. Similarly, Tang et al.(2015) built sentence representations using separate encoders and then fed them into sentence-level recurrent network. Feature representations of documents are obtained via mean pooling over hidden representations of sentences.

Yang et al.(2016) proposed Hierarchical Attention Network(*HAN*) leveraging the benefits of text hierarchy and self attention mechanism. HAN can been seen as a hierarchical recurrent structure where hierarchies share the same module structure: gated RNN(GRU)(Chung et al., 2014) with self-attention mechanism which computes a real-valued weight with respect to RNN hidden state at each timestamp. Their work incorporated structure bias into model design and showed that not all sentences in a document are equally important for classification decision.

Multiple Instance Learning(MIL) aims at solving problems where labels associated with groups of instances or *bags*, while instance labels are unobserved. An aggregation function is utilized to combine instance predictions in order to predict bag-level labels. The goal is either to label bags(Maron and Ratan, 1998) or to infer bag and instance labels at the same time(Kotzias et al., 2015). Angelidis et al.(2018) introduced the multiple instance learning network(*MILNet*) focusing on the latter variant of MIL. While HAN combines sentence representation using attention mechanism, MILNet directly compute the weighted sum of segment level class distributions as the document class distribution, from which classification is decided by selecting the class associated with highest probability. The distributions are computed using the same GRU module with attention introduced in HAN model. Details of HAN and MILNet are described in section 3.2.

Our work is different from previous hierarchical neural networks in the following aspects: while the previous works focus on jointly learning segment encoder and document classifier via document level supervision, we view the former as a component of the latter and train them sequentially using different level of supervisions. We first train the segment classifier on paragraph dataset, which can be seen as a supervised pretraining process. We then use the pretrained weights to initialize the segment encoding module of the hierarchical document classifier. The pretrained segment encoder and document classifier are jointly fine-tuned with gradient-based learning using maximum-likelihood objective function.

In addition, our model is able to generate paragraph-level label predictions through learned segment classifier, which can be applied in tasks such as aspect-level sentiment classification.

3 Methods

The overall architecture of our model is illustrated in Figure 1. Our model leverages intra-network knowledge transfer to improve document classification. We describe the structure of our model and different components in detail in the following sections.

3.1 Problem formulation

Let D_d denote a collection of documents, where each document d is associated with a label $y_d \in [1, c]$, c is the number of classes. Each $d = (p_1, p_2, \dots p_n)$ can be seen as a sequence of partitions, generated by partitioning policy discussed in 3.2. Our aim is to construct document classifier C_d that predicts document label $\hat{y_d}$ by selecting the class with maximum probability according to the distribution $P_d = (P_d^{(1)}, P_d^{(2)}, ..., p_d^{(c)})$.
Under MIL model setting, P_d is calculated by aggregating paragraph distributions $P_1, P_2, ...P_n$

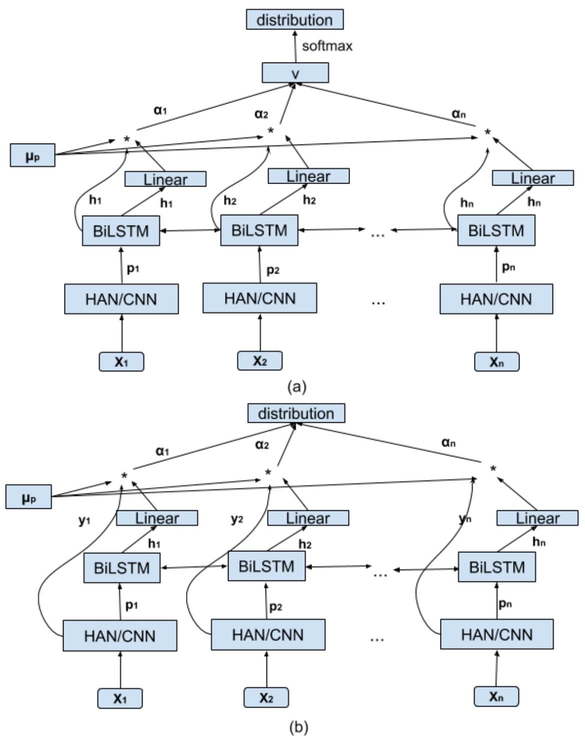

Figure 1: (a) Hierarchical Attention Network (b) Multiple Instance Learning Network

where P_i is obtained by a paragraph-level classifier:

$$P_i = \hat{g}_{\theta_p}(v_i) \tag{1}$$

$$P_d = \hat{f}_{\theta_d}(P_1, P_2, ...P_n) \tag{2}$$

Where v_i denotes the feature representation of paragraph p_i, f_θ and g_θ are single-layer feed-forward neural network in our model design.

A non-MIL hierarchical classifier predicts document label conditioned on the document feature representation. The paragraph representations are constructed independently by paragraph encoder:

$$v_i = \hat{C}_p(x_i) \tag{3}$$

Document level representation is calculated by combining paragraph representations using an aggregation function:

$$v_d = \hat{C}_d(v_1, v_2, ...v_n) \tag{4}$$

$$P_d = \hat{f}_{\theta_d}(v_d) \tag{5}$$

Where x_i denotes the word-vector input of paragraph p_i, C_p and C_d denote paragraph and document encoders respectively. We will discuss each component in details in section 3.2.

3.2 Split-Encode-Attend Network

Based on the intuitive assumption that paragraphs contain significant indicators of document class labels, our proposed split-encode-attend model involves the following components: (i) a neural classifier that is able to predict paragraph-level class label, trained on paragraph dataset (ii) an attention-based aggregation function producing document prediction from paragraph information. MILNet uses attention module to compute the weights for each paragraph's class distribution which are used to compute the weighted sum of paragraph distributions. HAN derives attention weights with respect to each paragraph's hidden representation to construct document level predictors. We illustrate both models in detail and compare their classification performances in section 4.4.

Document Partitioning We partition the documents into paragraphs and label each paragraph with the same class label associated with the document. Formally, we split each document-label pair (D_i, l_i) into a list of paragraph-label pairs $(p_{i_1}, l_i), (p_{i_2}, l_i), ..., (p_{i_n}, l_i)$ where $D_i = (p_{i_1}, p_{i_2}, \ldots p_{i_n})$ and l_i is the label associated with document D_i. We denote this constructed dataset D_p. In dataset where natural partition of paragraphs are given by the author, we simply use delimiters as paragraph boundaries to split the documents. Otherwise we use unsupervised text segmentation(Glavaš et al., 2016) which segments text into semantically coherent fragments by computing cosine similarities of all pairs of sentence vectors. A max clique among sentences is firstly constructed based on similarities scores then gets merged into non-overlapping sequences of contiguous sentences.

Paragraph Encoding The paragraph classifier is composed of text encoder C_p and single layer neural network g_{θ_p}. We train the classifier on dataset D_p to derive v_i and P_i, the hidden representation and class distribution for paragraph p_i respectively. Among various neural network classifiers, we select shallow CNN(Kim, 2014), Hierarchical Attention Network(Yang et al., 2016) and bidirectional LSTM(Graves and Schmidhuber, 2005) in our experiments. For instance in convolutional neural network experiment setting, we have:

$$v_i = CNN(p_i) \tag{6}$$

Paragraph Classification Paragraph class distribution $P_i = (P_i^{(1)}, P_i^{(2)}, ..., p_i^{(c)})$ is obtained using a softmax classifier:

$$Pi = softmax(W_p v_i + b_p) \tag{7}$$

where W_p and b_p are classifier g_{θ_p}'s parameters.

Document Encoding and Classification In the non-MIL setting, document-level prediction is be obtained by conditioning on feature representation v_d given by $\hat{C}_d$(Equation (5) and (6)). Inspired by HAN, we use Bidirectional LSTM with attention weights as C_d.

$$h_i = BiLSTM(v_i) \tag{8}$$

To focus on the paragraphs that contributes the most to classification decision, we use attention mechanism and introduce document level context vector μ_d:

$$\mu_i = tanh(W_d h_i + b_d) \tag{9}$$

$$\alpha_i = \frac{exp(\mu_i^T)\mu_d}{\sum_{j=1}^{n} exp(\mu_j^T)\mu_d} \tag{10}$$

$$v_d = \sum_{i=1}^{n} \alpha_i h_i \tag{11}$$

That is, we first feed the paragraph annotation h_i through a one-layer MLP to get μ_i, the transformed representation of h_i. The context vector μ_d is randomly initialized and learned during training, which can be thought of as a query vector recognizing the importance of a paragraph. An attention weight α_i is computed as normalized similarity between μ_i and μ_d. The final representation of document v_d is calculated as the weighted sum of paragraph hidden states using the attention weights.

The document classification conditioned on v_d:

$$P_d = softmax(W_f v_d + b_f) \tag{12}$$

We use the negative log likelihood of the document prediction as training loss:

$$L = -\sum_d \log P_d^{(y_d)} \tag{13}$$

In the MIL setting, document-level prediction can be obtained by directly taking average of paragraph class distributions: $P_d^{(k)} = \frac{1}{n}\sum_{i=1}^{n} P_i^{(k)}, \forall k \in [1, c]$. This as a naive approach to combine paragraph-level distributions without considering the different importance of each paragraph. MILNet directly take the weighted sum of paragraph distributions as document class distribution:

$$P_d^{(k)} = \sum_{i=1}^{n} \alpha_i P_i^{(k)}, \forall k \in [1, c] \qquad (14)$$

where the attention weights are calculated using Equations (8), (9) and (10). Other components are the same as those in the non-MIL setting.

4 Experiments

In this section we describe the data used to evaluate the performance of our model. We also give detailed description of our experiment settings and compared models.

4.1 Datasets

We evaluate our model on three large-scale datasets. The IMDb corpus of movie reviews was obtained from (Maas et al., 2011). Each movie review is associated with a binary label indicating positive or negative overall sentiment. The Yelp'13 polarity corpus was introduced in Yelp Dataset Challenge and contains customer reviews of local business. The NTY Scientific Journal Corpus(NYT SJ) compiled by (Louis and Nenkova, 2013a) is composed of long scientific articles, each document is assigned a binary label indicating its linguistic quality. The statistics of these datasets are summarized in Table 1.

4.2 Model Comparison

4.2.1 Baseline model

BOW and TFIDF + MLP We extract top 1000 words in each corpus to construct bag-of-word and TF-IDF features. We perform grid search over various n-gram parameters in the feature vectors and different sizes of hidden layers to obtain the best results on validation set.

4.2.2 Aggregation function

We compare the paragraph attentive pooling with the following paragraph information aggregation methods:

Majority: Class with highest number of votes among paragraph predictions(denoted with subscript *maj*).

Average: Normalized sum of probability distributions calculated for all paragraphs, denoted with subscript *avg* (section 3.2).

Mean Pooling: Instead of computing attention weights, we take the mean pooling over hidden states of all paragraphs to obtain the document representation, which is used as input for the document classification layer(*mean* subscript).

Max pooling: Same as mean pooling except we use max pooling over hidden states of paragraphs(*max* subscript).

MILNet: MILNet outputs the weighted sum of probability distribution for each paragraph. The attention weights are trained using document level supervision. Details are described in section 3.2(*mil* subscript).

4.3 Preprocessing

We obtained the paragraphs via splitting at delimiters: newline or XML paragraph tag </p> in Yelp dataset and NYT Scientific Journal corpus respectively. Since the paragraph splits developed by authors are not included in IMDb dataset, we obtained document partitions using GraphSeg(Glavaš et al., 2016) with minimum segment size of 1 sentence and relatedness threshold of 0.5.

In both document dataset and paragraph dataset, we first split each piece of text into sentences and tokenize each sentence using Stanford CoreNLP toolkit(Manning et al., 2014) . For NYT SJ corpus we performed lemmatization, punctuation and digit removal to identify high-level text quality. For IMDb and Yelp'13 we preserved emotion-heavy characters such as "!", "?" and emojis that are indicative of overall sentiment. We used 300-dimentional pretrained Glove embedding(Pennington et al., 2014). We replaced words appearing less than 5 times in the training set with a unique UNK tokens.

4.4 Model Training and Evaluation

For paragraph class prediction, we trained CNN, BiLSTM with mean/max pooling and HAN on paragraph dataset using SGD optimizer for 20 epoches. Mini-batches of 32 documents was used to fasten training speed. We used 300-dimentional pre-trained word embedding glove(Pennington et al., 2014). We fine tuned parameters on the validation set of New York Time Scientific corpus and obtained the following parameters: for CNN paragraph encoder, we used window sizes of 3, 4, 5 words, each with 100 feature maps. For both paragraphs and document encoders, we used the same set of parameters. We apply bidirectional LSTM with 50-dimensional hidden layers for each direction. The resulting size of attention vector is

Dataset	**NYT SJ**	**IMDb**	**Yelp**
Training # Documents	19253	22.25K	504K
Validation # Documents	2140	2.25K	56K
Testing # Documents	2377	25K	38K
Average # Words	1273.30	227.76	145.56
Maximum # Words	12731	506	1166
Average # Sentences	48.41	11.48	2.09
Maximum # Sentences	622	80	55
Average # Paragraphs	24	3.7	2.73
Maximum # Paragraphs	219	24	89
# Vocabulary Size(no UNK)	111K	60K	128K

Table 1: Training Dataset Statistics

therefore set as 50*2=100. Dropout regularization is introduced with dropout rate 0.5 on paragraph LSTM hidden states and input features for final classification.

We evaluate models in two ways. First, we combine different aggregation functions with various architectures after document partitioning and training paragraph classifiers. For each neural model(CNN, BiLSTM, HAN) we use the document classification results as baseline. We compare them with the results obtained using split-encode-attend training procedures. We also evaluate different aggregation methods on each dataset in order to decide the best aggregation techniques.

In our second sets of experiments, we focus on using fixed number of sentences in each paragraph instead of heuristic-based paragraph partitioning. We are interested to find out in what way and to what degree the paragraph partitioning improves classification.

5 Results and Discussion

Table 2 summarizes the document classification results using partition method described in section 4.3. The first block reports the classification accuracy using bag-of-words features + MLP model with hyper-parameters selected on validation set.

The second to the fourth block present the classification results using CNN, BiLSTM and HAN as paragraph classification model respectively. Each set of experiments using a particular architecture differentiates between models that are trained only on document-label dataset and models trained on paragraph dataset and document dataset sequentially as described in previous sections. We illustrate results of different aggregation functions(from simple averaging probability to attentive pool-

Dataset	IMDb	Yelp	NYT SJ
BOW + MLP	85.50	92.00	**90.61**
CNN	87.75	95.60	87.80
CNN_{maj}	84.19	93.88	83.09
CNN_{avg}	87.19	95.61	83.13
CNN_{attn}	**87.91**	96.14	87.34
CNN_{mil}	86.20	nan	84.98
CNN_{mean}	87.66	**96.19**	**87.93**
CNN_{max}	87.35	96.06	86.96
BiLSTM	**90.73**	96.38	87.80
$\text{BiLSTM}_{\text{maj}}$	87.20	93.84	87.88
$\text{BiLSTM}_{\text{avg}}$	89.63	95.53	87.76
$\text{BiLSTM}_{\text{attn}}$	90.01	96.60	**88.64**
$\text{BiLSTM}_{\text{mil}}$	88.21	96.61	88.14
$\text{BiLSTM}_{\text{mean}}$	90.28	96.60	88.26
$\text{BiLSTM}_{\text{max}}$	90.07	**96.62**	87.80
HAN	**91.01**	96.12	87.63
HAN_{maj}	86.96	92.74	86.50
HAN_{avg}	90.26	94.80	87.08
HAN_{attn}	89.86	**96.16**	**87.67**
HAN_{mil}	89.01	95.91	87.46
HAN_{mean}	90.12	96.11	86.62
HAN_{max}	90.44	95.01	87.43

Table 2: Document classification accuracy(in percentage)

Dataset	IMDb	Yelp	NTY SJ
CNN	75.54	84.65	83.13
BiLSTM	78.68	**85.34**	**84.85**
HAN	**79.24**	84.65	84.66

Table 3: Paragraph classification results based on partitioning given by author

Dataset	IMDb	Yelp	NTY SJ
CNN	72.35	84.37	85.33
BiLSTM	**76.18**	**85.18**	**86.06**

Table 4: Paragraph classification results using fixed length partitioning

Dataset	IMDb	Yelp	NYT SJ
CNN_{attn}	60.85	95.54	**87.72**
CNN_{mean}	69.68	95.61	87.59
CNN_{max}	**79.69**	**95.65**	87.25
$\text{BiLSTM}_{\text{attn}}$	87.17	**96.27**	88.14
$\text{BiLSTM}_{\text{mil}}$	84.64	96.22	86.87
$\text{BiLSTM}_{\text{mean}}$	87.29	96.16	87.84
$\text{BiLSTM}_{\text{max}}$	**87.46**	96.18	**88.30**

Table 5: Document classification accuracy using fixed length paragraph partition(in percentage)

ing)combined with each base classifier and compare them with the results obtained using the base model. The highest classification accuracies within each group of models are bold.

Paragraph level pooling leads to better performances on Yelp and NYT SJ datasets. We observe that there is no significant difference in classification performance between three paragraph level pooling methods(with subscripts *attn*, *mean* and *max*) across all experiments. In IMBD dataset experiment, the original BiLSTM and HAN models outperform our proposed model extension, while in all other experiments our model extension prevails the original model. MILNet results in the least competitive performances and occasionally encounters from numerical computing problem during loss computation.

We notice that simple average of individual paragraph distribution results in classification performances comparable to the original model. This method requires zero training at paragraph level aggregation, thus can be seen as a effective alternative of training document level classifier.

In linguistic classification task, BOW + MLP model is better than all CNN and RNN based models. The word count features indicate contents such as topic and sentiment, which is particularly powerful combined with MLP model in long document classification.

Table 3 summarizes the paragraph classification accuracies using each neural classifier. The results indicate the chances of correctly identifying a paragraph as being a part of a positive/negative document.

We observe that the paragraph classification accuracy for IMDb dataset is relatively low. Comparing with table 2, we suspect that this caused the document classification accuracy of paragraph classifier + aggregation function model to be lower than results of the original model. Additionally, the IMDb paragraph dataset is not obtained by directly splitting at the delimiters. We thus devise experiments aiming to test whether the paragraph classification accuracy is positively correlated with the document classification accuracy. Moreover, we examine if the document classification results vary with different paragraph partitioning methods. We construct a new paragraph dataset in which each paragraph contains a fixed amount of sentences, with the last paragraph of each document being the only exceptions as they contain all the remaining sentences of the document. We designate 3 as the length of each paragraph. Other experiment settings follow the description in section 4.4. The paragraph and document classification results are shown in Table 4 and Table 5 respectively.

Figures in Table 4 suggests that (1) fixed-length partition doesn't impair paragraph classification performance compared to natural paragraph split; (2) Graphseg text partitioning algorithm results in higher paragraph classification accuracy on IMDb dataset than fixed-length partitioning does.

We use the models trained on paragraph dataset obtained using fixed-length partitioning as underlying encoders of paragraph level aggregation layer. The results in Table 5 are worse than those in Table 2 in general. We therefore conclude that paragraph classification result is not the only factor of the choice of paragraph partitioning method.

The choice of partition method also depends on the semantic properties of dataset and the goal of classification task, as the degree to which a paragraph agrees with the document label is also determined by the specific content of the paragraph. The paragraph classification results on IMDb dataset show the disagreed sentiments between documents and their belonging sentences, which is again reflected in the low accuracies in document classification task. One needs to be cautious choosing when to include paragraph into consideration for document classification, considering above-mentioned peculiarities.

6 Conclusion

In this work, we present a neural network model training procedure within the framework of hierarchical neural network. As a departure from the commonly used neural network architecture trained using only document-level supervision, we train a paragraph level classifier and integrate it into the hierarchical model before the whole model is fine-tuned on the document-label set. We explored multiple paragraph partition methods and paragraph information aggregation functions. Experiment results demonstrate that our model extension improves or resembles results obtained by various neural architecture on Yelp dataset and New York Times Scientific Journal corpus.

References

Stefanos Angelidis and Mirella Lapata. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association of Computational Linguistics*, 6:17–31.

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

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

Vanessa Wei Feng and Graeme Hirst. 2013. Patterns of local discourse coherence as a feature for authorship attribution. *Literary and Linguistic Computing*, 29(2):191–198.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. Unsupervised text segmentation using semantic relatedness graphs. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 562–570.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606. ACM.

Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. 1998. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Annie Louis and Ani Nenkova. 2013a. A corpus of science journalism for analyzing writing quality. *Dialogue & Discourse*, 4(2):87–117.

Annie Louis and Ani Nenkova. 2013b. What makes writing great? first experiments on article quality prediction in the science journalism domain. *Transactions of the Association for Computational Linguistics*, 1:341–352.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Oded Maron and Aparna Lakshmi Ratan. 1998. Multiple-instance learning for natural scene classification. In *ICML*, volume 98, pages 341–349.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 913–921. Association for Computational Linguistics.

Juan Ramos et al. Using tf-idf to determine word relevance in document queries.

Deboleena Roy, Priyadarshini Panda, and Kaushik Roy. 2018. Tree-cnn: A deep convolutional neural network for lifelong learning. *arXiv preprint arXiv:1802.05800*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558.

Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. 2015. Semantic clustering and convolutional neural network for short text categorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 352–357.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428–2437.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

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.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52.

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.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.