Natural Language Processing Classification with Deep Learning

MSDS 458 – Assignment 3

Siyuan Liu

Email: siyuanliu2022@u.northwestern.edu

February 19th, 2022

Abstract

Text classification has been one of application areas under Natural Language Processing (NLP) domain where deep learning has achieved great progress and becomes dominant approach in past few years. As one of deep learning model structure, Recursive Neural Network (RNN) and its deviants such as Long Short-Term Memory (LSTM) and Gate Recurrent Units (GRU) have been proved to outperform other models thanks to its advantageous capability capture the sequential correlations. In this research, the AG's news topic classification dataset that comprises of documents in four topic classes including World, Sports, Business, and Sci/Tech is used for training purposefully structured neural network model variants for classification. The primary objective of this research is to compare the performance of RNN, LSTM, and CNN on text classification based on differentiated model architecture and explore how these factors influence the model performance in terms of both accuracy and time.

Key Words:

Text classification; Neural network; Recursive Neural Network (RNN), Long Short Term Memory (LSTM)

1. Introduction

Text classification has been one of focused areas under Natural Language Process (NLP) where deep neural network has achieved great success and outperformed traditional machine learning models as it leverages different approach for feature extraction and classifier principles (Otter, et al., 2020). As the variable-length text such as sentences, paragraphs and documents could be represented by vectors with fixed length, neural network models leverage this technique to extract features from text context and and then combine them with the different architectures of neural networks (Liu et al., 2016). The flow of continuous words is then treated as sequence of vectors, that could be fed into the neuron network model for it to learn the correlation. Sequence-based models construct sentence representations from word sequences by taking in account the relationship between successive words (Johnson and Zhang, 2015). Recursive Neural Network model and its deviants such as LSTM and GRU have been widely used for text classification task and achieved promising results.

In this study, multiple models have been designed with differentiated architecture to experiment with both text representation and feature extraction layer such as LSTM, Simple RNN and CNN and explore how variants of these structures make impact on the model performance. The dataset employed is from AG News Class dataset, which has been purposefully manipulated and labelled. A snapshot of the dataset is shown as Figure 1.

	description	label
0	AMD #39;s new dual-core Opteron chip is designed mainly for corporate computing applications, including databases, Web services, and financial transactions.	3 (Sci/Tech)
1	Reuters - Major League Baseball\Monday announced a decision on the appeal filed by Chicago Cubs\pitcher Kerry Wood regarding a suspension stemming from an\incident earlier this season.	1 (Sports)
2	President Bush #39;s quot;revenue-neutral quot; tax reform needs losers to balance its winners, and people claiming the federal deduction for state and local taxes may be in administration planners #39; sights, news reports say.	2 (Business)
3	Britain will run out of leading scientists unless science education is improved, says Professor Colin Pillinger.	3 (Sci/Tech)
4	London, England (Sports Network) - England midfielder Steven Gerrard injured his groin late in Thursday #39;s training session, but is hopeful he will be ready for Saturday #39;s World Cup qualifier against Austria.	1 (Sports)
5	TOKYO - Sony Corp. is banking on the \$3 billion deal to acquire Hollywood studio Metro-Goldwyn-Mayer Inc...	0 (World)
6	Giant pandas may well prefer bamboo to laptops, but wireless technology is helping researchers in China in their efforts to protect the engandered animals living in the remote Wolong Nature Reserve.	3 (Sci/Tech)
7	VILNIUS, Lithuania - Lithuania #39;s main parties formed an alliance to try to keep a Russian-born tycoon and his populist promises out of the government in Sunday #39;s second round of parliamentary elections in this Baltic country.	0 (World)
8	Witnesses in the trial of a US soldier charged with abusing prisoners at Abu Ghraib have told the court that the CIA sometimes directed abuse and orders were received from military command to toughen interrogations.	0 (World)
9	Dan Olsen of Ponte Vedra Beach, Fla., shot a 7-under 65 Thursday to take a one-shot lead after two rounds of the PGA Tour qualifying tournament.	1 (Sports)

Figure 1. Examples from the AG News Topic Class training set.

2. Literature review

Recurrent neural network (RNN) has been proved to be able to capture sequential correlation as it is trained on sequence of vectors and able to capture the temporal correlation. To deal with the gradient vanishing problems, more advanced variants have been developed such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (CRU) (Chung et al, 2014). They are often used for text classification as they can memorize previous output and learn contextual characteristics (Zhou, 2022). In additional, Convolutional Neural Network (CNN) has also been studied for text classification purpose by Yoon Kim (Kim 2014).

Feature extraction is one of key steps that directly impact the model performance, it refers to the phase where textural data is transformed into vectors which could be understood by the model. Neural Network Language Model (NNLM) as one of earlier studies was firstly proposed by Bengio et al. in 2003. It basically uses a uses a three-layer feedforward neural network to parse through the corpus (Bengio, 2003). Later on, GloVe, a global word vector model was proposed by Pennington et al., which pays more attention to global text information, forming an association matrix through the word segmentation results of the text and decomposing it to obtain a set of distributed word vectors (Pennington et al., 2014).

3. Method

3.1. Data Collection & Exploration

The dataset used for this study is AG from TensorFlow, which comprises of more than 1 million news articles collected by ComeToMyHead from more than 2000 news sources. The AG's news topic classification dataset spreads across 4 largest classes including World, Sports, Business, and Sci/Tech. Each class contains 30,000 training samples and 1,900 testing samples. The total number of training samples is 120,000 and testing 7,600.

3.2 Data Preprocessing and Exploration

Data is pre-loaded from TensorFlow. Since there is no validation dataset from the original source, we split the training data and held 5% for validation, yielding 114,000 instances for training and 6,000 for validation. An exploratory data analysis was conducted to investigate the corpus size, token extraction and distribution.

There are total 3,909,695 words in the corpus of 127,600 news articles. The tokens per document basis distribute left skewed with majority of documents having 20 to 40 tokens shown in Figure 2.

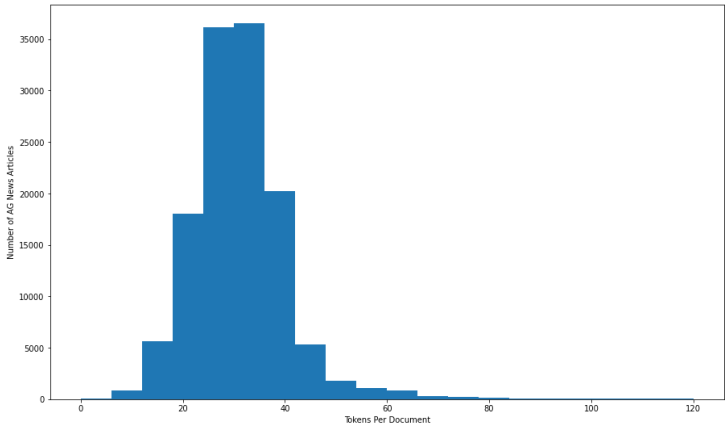

Figure 2. Token per document distribution

3.3 Model Construction and Training

Given the differentiated purpose this study attempts to achieve, we set up four experiments with each experiment focusing on different study target and experimenting several models. Experiment A concentrates on encoding process by which the input data is generated for following feature extraction. Three variables including vocabulary size (1,000, 2,000, 3,000), customization of vocabulary (non-edited, edited) and output sequence length (default, arbitrary) have been considered to build up exhaustive combinations of encoder configuration, which yields 12 alternative encoders. To make the fair comparison, the feature extraction and classification parts are same across all models using one Bidirectional LSTM layer follow by DNN layers as shown in Figure 3.

Model	Encoder	Model
Model 1	1000 vocabulary size, no edit of vocabulary, default output sequence length	Bidirectional LSTM, 1 layer
Model 2	1000 vocabulary size, no edit of vocabulary, arbitrary output sequence length	Bidirectional LSTM, 1 layer
Model 3	1000 vocabulary size, customized vocabulary by removing stopwords (NLTK), default output sequence length	Bidirectional LSTM, 1 layer
Model 4	1000 vocabulary size, customized vocabulary by removing stopwords (NLTK), arbitrary output sequence length	Bidirectional LSTM, 1 layer

Model 5	2000 vocabulary size, no edit of vocabulary, default output sequence length	Bidirectional LSTM, 1 layer
Model 6	2000 vocabulary size, no edit of vocabulary, arbitrary output sequence length	Bidirectional LSTM, 1 layer
Model 7	2000 vocabulary size, customized vocabulary by removing stopwords (NLTK), default output sequence length	Bidirectional LSTM, 1 layer
Model 8	2000 vocabulary size, customized vocabulary by removing stopwords (NLTK), arbitrary output sequence length	Bidirectional LSTM, 1 layer
Model 9	3000 vocabulary size, no edit of vocabulary, default output sequence length	Bidirectional LSTM, 1 layer
Model 10	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length	Bidirectional LSTM, 1 layer
Model 11	3000 vocabulary size, customized vocabulary by removing stopwords (NLTK), default output sequence length	Bidirectional LSTM, 1 layer
Model 12	3000 vocabulary size, customized vocabulary by removing stopwords (NLTK), arbitrary output sequence length	Bidirectional LSTM, 1 layer

Figure 3. Model Design & Structure – Encoder Comparison (Experiment A)

Experiment B aims to test out regular RNN structure. The encoder is set same across all models, which is identified from Experiment A as best encoder. Five models are designed with various architectures based on Simple RNN and bidirectional training configuration as illustrated in Figure 4.

Model	**Encoder**	**Model**
Model 1	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	1 Simple RNN (64), 1 DNN (64)
Model 2	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	1 Bidirectional Simple RNN (64), 1 DNN (64)
Model 3	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	2 Stacked Bidirectional Simple RNN (64, 32), Dropout (0.5), 1 DNN (64)
Model 4	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	2 Stacked Simple RNN (64, 32), Dropout (0.5), 1 DNN (64)

Model 5	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	3 Bidirectional Simple RNN (128, 64, 32), Dropout(0.5), 1 DNN (64)

Figure 4. Model Design & Structure – RNN Comparison (Experiment B)

Experiment C moves focus to LSTM. Similar to Experiment B, same encoder is utilized across all models followed by various model architectures for feature extraction. What's different is the functional layer is replaced with LSTM and configured by stacking more layers or applying bidirectional training. Details are shown in Figure 5.

Model	Encoder	Model
Model 1	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	1 LSTM (64), 1 DNN (64)
Model 2	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	1 Bidirectional LSTM (64), 1 DNN (64)
Model 3	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	2 Stacked Bidirectional LSTM (64, 32), Dropout (0.5), 1 DNN (64)
Model 4	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	2 Stacked LSTM (64, 32), Dropout (0.5), 1 DNN (64)
Model 5	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	2 Stacked LSTM (128, 64), 2 Dropout (0.3), 1 DNN (64), L2 Regularizer (0.001)
Model 6	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	2 Stacked Bidirectional LSTM (128, 64), 2 Dropout (0.5), 1 DNN (64), L2 Regularizer (0.001)

Figure 5. Model Design & Structure – LSTM Comparison (Experiment C)

For Experiment D, two CNN-architected models are tested out with same encoder configuration in Experiment B and C as shown in Figure 6.

Model	Encoder	Model
Model 1	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	1 CNN (64), 1 MaxPooling, 1 DNN (64)
Model 2	3000 vocabulary size, no edit of vocabulary, arbitrary output sequence length (128)	2 CNN (64, 128), 2 MaxPooling, 2 Droput (0.3), 1 DNN (64)

Figure 6. Model Design & Structure – CNN Comparison (Experiment D)

Dropout layer and regularization (L2 with learning rate 0.001) are utilized for models with stacked RNN/LSTM/CNN layers to mitigate overfitting problems. Given the computational complexity of model variants, optimizer and loss functions are not considered for optimal tesification in this study.

4. Results

Experimental models are evaluated with performance metrics being traced, including accuracy and loss across train, validation and test set. Figure 7 illustrates the results from Experiment A, in which the primary focus is on experimenting with different encoders and their associated impact on model performance with all else set the same.

Model	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss	Process_Time
Model1	0.8587	0.3860	0.8570	0.3948	0.8474	0.4133	1,194s
Model2	0.8579	0.3943	0.8557	0.3981	0.8493	0.4154	1,373s
Model3	0.8496	0.4114	0.8403	0.4365	0.8461	0.4360	1,955s
Model4	0.8612	0.3816	0.8608	0.3926	0.8538	0.4085	2,772s
Model5	0.9002	0.2796	0.8818	0.3299	0.8796	0.3343	2,669s
Model6	0.8911	0.3057	0.8800	0.3392	0.8787	0.3417	2,446s
Model7	0.8473	0.3578	0.8610	0.4025	0.8572	0.4017	1,557s
Model8	0.8814	0.3326	0.8633	0.3796	0.8655	0.3818	3,495s
Model9	0.9026	0.2776	0.8863	0.3192	0.8857	0.3240	1,316
Model10	0.9041	0.2690	0.8867	0.3166	0.8862	0.3226	2,109
Model11	0.8945	0.3002	0.8772	0.3557	0.8778	0.3507	1,804s
Model12	0.8948	0.2894	0.8732	0.3703	0.8755	0.3657	4,589s

Figure 7. Summary Table of Performance Score – Experiment A (Encoder Comparison)

Model 10 yields highest test accuracy 0.8862, slightly better than others. The observation of results yields findings in two perspectives,

- Increasing vocabulary size from 1000 to 3000 has improve model performance by approximately 2%-4% without increasing the overfitting gap, indicating that adding more information into features can help better classify the document topic as shown in Figure 8.

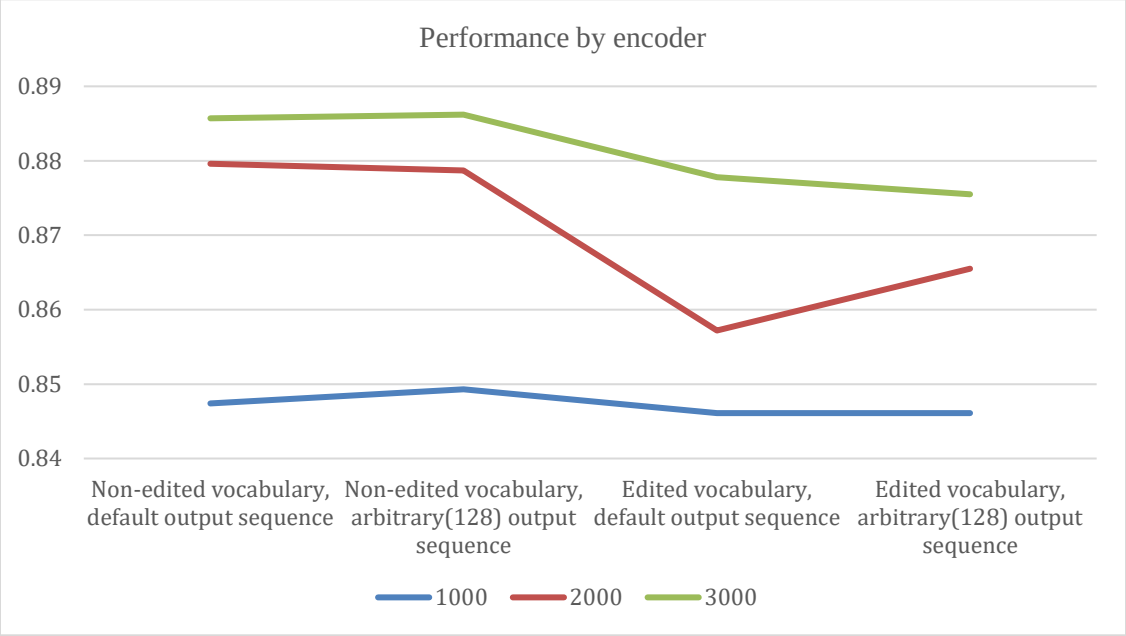

Figure 8. Accuracy score by encoder – vocabulary size

- Encoders that was generated from original corpus yields better results than the ones that are customized by removing stop words (NLTK) such as “the”, “a”, suggesting removing the stop words might cause loss of information and then reduction of model performance, a deeper dive into the stop words is needed.

The model 3 from Experiment B return the best test accuracy 0.8838 as shown in Figure 9, slightly better than other models. Overall, the bidirectional training outperforms one directional training indicated by the performance surplus as Model 2 with 1 layer of bidirectional Simple RNN yield 0.8800 test accuracy better than 0.8762 of Model 1 with one layer Simple RNN, while Model 3 with stacked two layers of Simple RNN achieved 0.8838 outperforming Model 4 with 0.8751 accuracy score, which suggest that bidirectional

training can enable the model to better capture the semantic context hence enhancing the classification accuracy.

Model	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss	Process_Time
Model1	0.9123	0.2537	0.8773	0.3509	0.8762	0.3617	1,090s
Model2	0.9085	0.2598	0.8903	0.3154	0.8800	0.3312	899s
Model3	0.9071	0.2896	0.8890	0.3283	0.8838	0.3388	1.698s
Model4	0.8989	0.3129	0.8780	0.3485	0.8751	0.3640	1,002s
Model5	0.9169	0.2493	0.8888	0.3261	0.8818	0.3531	3.080s

Figure 9. Summary Table of Performance Score – Experiment B (RNN Comparison)

Additionally, adding more layers of RNN does not bring performance gain but oppositely cause overfitting problem as indicated by Model 5 performance in Figure 10. There is considerable discrepancy between training and validation accuracy after epoch 3 training. The time of training increase by 3 times as computation complexity increase with more parameters.

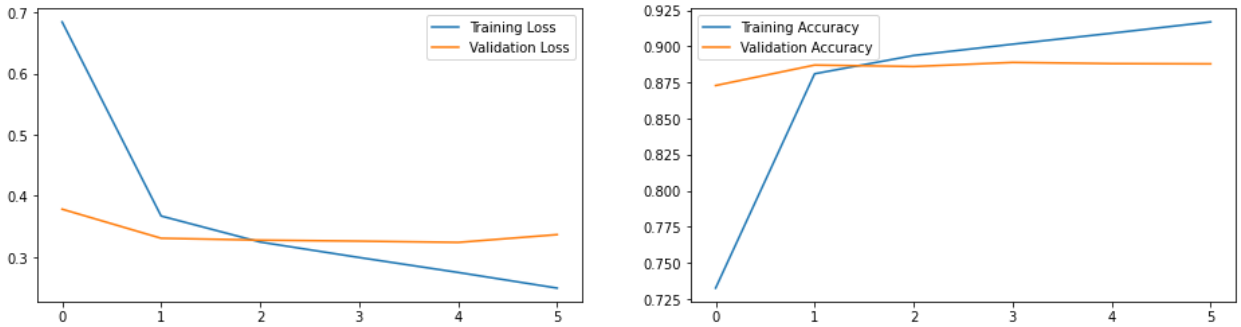

Figure 10. Performance plot of model 5 (three layer of bidirectional Simple RNN) – Experiment B

The model 5 from Experiment C return the best test accuracy 0.8853 as shown in Figure 11, slightly better than other models. Consistent with Experiment B, model with bidirectional LSTM layers outperforms ones with single direction layers based on comparison of first four models. Additionally, increasing units in LSTM layers does not show significant performance gain as we can see the model 5 and achieve less than 0.1% test accuracy gain (0.8853) up from model 4 (0.8851).

Model	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss	Process_Time
Model1	0.9022	0.2827	0.8868	0.3211	0.8839	0.3312	921s
Model2	0.9084	0.2604	0.8908	0.3131	0.8887	0.3148	1,813s
Model3	0.8998	0.3086	0.8850	0.3376	0.8791	0.3538	2,830s
Model4	0.9075	0.2807	0.8865	0.3286	0.8851	0.3376	3,718s
Model5	0.8989	0.3088	0.8827	0.3526	0.8853	0.3547	3,046s
Model6	0.91536	0.2595	0.8868	0.3365	0.8838	0.3464	8,258s

Figure 11. Summary Table of Performance Score – Experiment C (LSTM Comparison)

In Experiment D, two models was trained and tested. Both models delivered promising results but Model 2 with two layers of Conv1D achieved slightly higher accuracy score 0.8907 as shown in Figure 12.

Model	Train_Accuracy	Train_Loss	Val_Accuracy	Val_Loss	Test_Accuracy	Test_Loss	Process_Time
Model1	0.9090	0.2535	0.8903	0.3088	0.8883	0.3147	424s
Model2	0.9163	0.2341	0.8912	0.3085	0.8907	0.3139	1,254s

Figure 12. Summary Table of Performance Score – Experiment D (CNN Comparison)

If we conduct a horizontal comparison across Simple RNN, LSTM and CNN, surprisingly the model with 2 stacked CNN yielded highest test accuracy 0.8907 with shortest amount of training time 1,254s followed by model with 2 stacked bidirectional LSTM. However, the LSTM model takes longer to train than the other two models as shown in Figure 13.

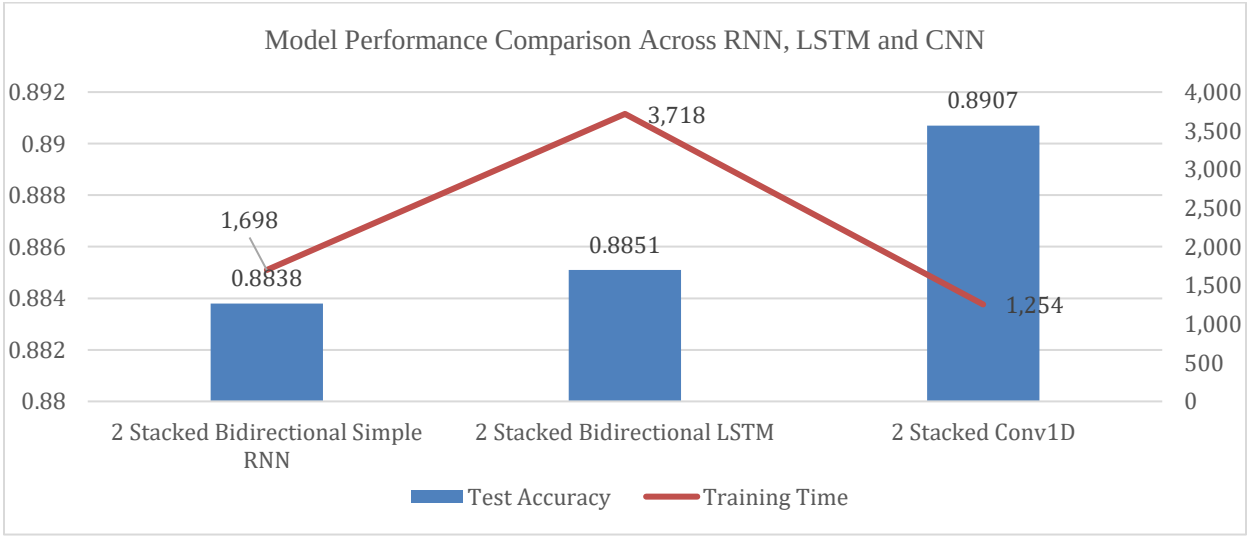

Figure 12. Summary Table of Performance Score – RNN / CNN / LSTM

5. Conclusions

Overall, both CNN and RNN including LSTM delivers promising performance on text classification task in this study. Models trained on bidirectional structured layers across LSTM, Simple RNN outperforms ones with single directional layers. LSTM models on average take longer to train due to computational complexity driven by the size of weights. Mostly the model constructed with two layers of neural network can handle the task quite well while stacking more layers on top does not improve the performance significantly and computing cost effectively. Further investigation is needed to get better understanding on encoder construction and tuning of other hyperparameters such as optimizer and loss function.

Reference:

Otter, Daniel W., Julian R. Medina, and Jugal K. Kalita. "A Survey of the Usages of Deep Learning for Natural Language Processing." *IEEE Transactions on Neural Networks and Learning Systems* 32, no. 2 (2021): 604–24. https://doi.org/10.1109/tnnls.2020.2979670

Choi, Iksoo, Jinhwan Park, and Wonyong Sung. "Character-Level Language Modeling with Gated Hierarchical Recurrent Neural Networks." *Interspeech 2018*, 2018. https://doi.org/10.21437/interspeech.2018-1727.

Zhou, Hai. "Research of Text Classification Based on TF-IDF and CNN-LSTM." *Journal of Physics: Conference Series* 2171, no. 1 (2022): 012021. https://doi.org/10.1088/1742-6596/2171/1/012021.

Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. https://doi.org/10.3115/v1/d14-1181.

Bengio, Yoshua, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. "Neural Probabilistic Language Models." *Innovations in Machine Learning*, n.d., 137–86. https://doi.org/10.1007/3-540-33486-6_6.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global Vectors for Word Representation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. https://doi.org/10.3115/v1/d14-1162.