PROJECT FINAL REPORT

Zheyuan Yang
Student# 1006770567
andrewzheyuan.yang@mail.utoronto.ca

Peixuan Li
Student# 1006719464
adampeixuan.li@mail.utoronto.ca

Derui Yang
Student# 1006664655
derui.yang@mail.utoronto.ca

Daixin Tian
Student# 1006661408
daixin.tian@mail.utoronto.ca

ABSTRACT

This report concludes Group 12's project. Our project is to use LSTM model to classify news. In this final report, we will describe our project in details, discuss our data processing, architecture, and baseline, and show our model choosing and tuning process. The group also evaluate on our model based on testing result, giving our in-depth discussion and finding on our model performance. The last part of the article explores some ethical considerations and project difficulties. —-Total Pages: 8

1 INTRODUCTION

Personalized fanout refers to the media notifications prioritizing the news, videos, and articles based on users' preferences acquired from the big data analysis (Morgan, 2017). Majority of media platforms are utilizing personalized fanout to satisfy users' diversified and personalized needs. However, to accomplish personalized notifications, authors are usually required to manually choose tags that are related to their articles or videos. The goal of this project is to suggest a deep-learning-based news classifier that automatically generates tags for news articles. Specifically, our product allows users to input a news article, and it outputs the predicted category for this article, from the set "technology, sports, politics, entertainment, world, automobile, and science".

News article tag classification is helpful in personalized fanout. From article authors' perspective, our product helps with adding tags to the articles such that they will not have to waste time on choosing tags for their articles. From news platforms' perspective, tags could help with better news analysis. Tags will support news platforms to push articles from different fields to targeted users according to their preferences. This could also be helpful for advertisers planting advertisement in the correct fields (Contributor, 2021).

The categorization based on machine learning, compared to categorizing articles manually, shows high efficiency because it can proceed extremely fast with modern processors. In addition, it would also exhibit high accuracy in categorization because it avoids human's subjective bias. Our product will produce objective results as long as our training dataset is given correctly with non-biased labels.

2 ILLUSTRATION/FIGURE

Our project is based on a 1-layer bidirectional LSTM model. For the input, we will normalize the short news articles to no more than 61 words. Then processing and learning those embeddings in a bidirectional LSTM layer with hidden size equal to 256. Finally, we would use a linear classifier to sort results into the 7 news categories. Overall model is illustrated in Figure 1.

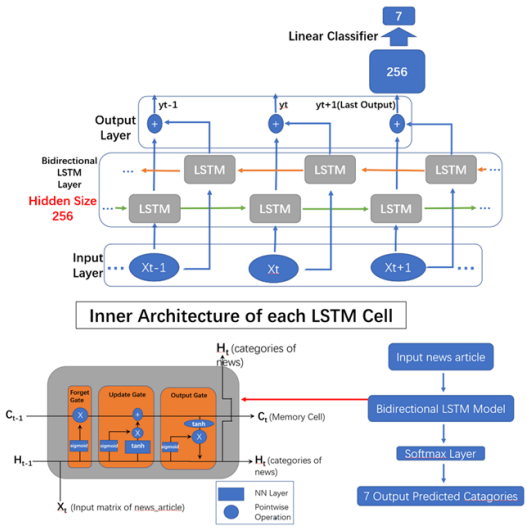


Figure 1: Illustration for overall model

3 Background

The vast expansion of the internet in recent decades has led to development of news media. Nowadays, mass amount of news are spread to users through internet. News categorization allows readers to navigate articles more quickly (Kaur & Khiva, 2016). However, manually tagging news articles is low-efficient and subjective because humans can make mistakes during classification, and categories can change over time due to large amount of new information (Salminen et al., 2019). The team define the problem and build goal of this project, based on the above background

Some researchers has put effort into this field and made much contribution. (Dilrukshi et al., 2013) uses SVM (support vector machine) to classify Twitter News. Researchers first collected short messages from Twitter API and classified them into 12 categories manually. Then, they removed unimportant words and created bags of words as features. SVM model was used because it is suitable for data with multiple dimensions. It can also use kernels to separate data linearly and find the global minimum. From the precision and recall of the result, SVM demonstrated its effectiveness in news classification.

From the related work done by other researchers, we learned that SVM could be a possible way for our model or could be used as baseline model. Most importantly, the team could learn the 'stop words' idea from their work and implement to our data processing part. In fact, the team did use stop words while processing data.

4 Data Processing

4.1 Collecting and splitting dataset

We get our dataset from Kaggle, a total of seven .csv files of daily news collected by Yadav (2021). We combine the first two files, named inshort_news_data-1.csv and inshort_news_data-2 into a large .csv file containing in total 6380 samples as our training set; inshort_news_data-3.csv is used as validation set; inshort_news_data-7.csv acts as testing set. The entire dataset contains 9500 samples, and 70% of them are split to training set, 15% to validation set, and 15% to test set. Lengths of each set is shown on the left of Figure 2.

4.2 Formatting dataframe and Encoding of news_category

The dataset includes three columns namely news_headline, news_article, and news_category. For our project, only article contents and its category will be considered. Therefore, we use read_csv

function from Pandas library to read and extract the news_article and news_category columns (Pan, 2022). The extracted dataframe is shown on the right of Figure 2.

The news_category array contains seven classes, namely technology, sports, politics, entertainment, world, automobile, and science. We can encode them into "class 0,1,2,3,4,5,6" by using the get_dummies function, which outputs as a Pandas dataframe (Pan, 2022). Then we use to_numpy function to convert the one-hot-encoding dataframe into a numpy array for convience of further operations.

```
print('Num training articles: ', len(train_data))
print('Num validation articles: ', len(val_data))
print('Num testing articles: ', len(test_data))
```

```
Num training articles:  6380
Num validation articles:  1560
Num testing articles:  1560
```

```
train_data.head()
```

	news_article	news_category
0	DeepMind's AI system 'AlphaFold' has been reco...	technology
1	Microsoft Teams will stop working on Internet ...	technology
2	China, in response to reports of US adding Chi...	technology
3	The global smartphone sales in the third quart...	technology
4	The European Union (EU) is hoping that US Pres...	technology

Figure 2: Sizes of train, validation, and test sets (left), and example of dataset after first-stage formatting (right).

4.3 Filtration and Digitization of news_article

To make it easier to analyze, we will firstly split articles into arrays of words by using split function in Pandas library. However it will be noticeable that the occurrence of "a", "the", "this", "me", "you", and etc has much higher frequency than other words. These are called stop words and should be filtered out before processing with our data for enhancing the efficiency and accuracy of text classifications. We will use a stop word list created by Larsyencken (2011). The filtered data is shown in Figure 3.

```
print(X_train[0])
```

```
['deepmind's', 'ai', 'system', ''alphafold'', 'recognised', 'solution', ''protein', 'folding',', 'grand', 'challe
```

Figure 3: Example of a filtered training sample.

Programs cannot comprehend texts directly, therefore we need Global Vectors for Word Representation (GloVe) as our tool to digitize words (Pennington et al., 2014). Specifically, we would build embedding matrix for each word using the 6B GloVe table with dimension of 50 from the torchtext package, so that each matrix represents the feature of that word and the Cosine Distance between two matrices represents how related these two words are.

5 Architecture

We mainly use Long Short-Term Memory (LSTM) as our primary model, and the model with the best performance is also an LSTM neural network. We also try to use a transformer with multihead attention and positional encoding. It has a decent performance, but in general, the validation accuracy is still lower than the validation and test accuracy of the LSTM model. Figure 4 shows our hyperparameter tuning process.

Our final model, which is `LSTM_news_classifier_3(50, 256, 7)`, has one bidirectional LSTM layer and a linear layer to classify news into 7 categories. The output of the LSTM layer is the last output in output layer. The hidden size of our bidirectional LSTM is 256. Also, in this training, the batch size is 256, and the learning rate is 0.01. We also use Mean Square Error (MSE Loss) as our loss function and Adam as our optimizer. Here is the code of our final model.

```
1 class LSTM_news_classifier_3(nn.Module):
2     def __init__(self, input_size, hidden_size, num_class):
3         super(LSTM_news_classifier_3, self).__init__()
```

```
        self.name = "LSTM_3"
        self.hidden_size = hidden_size
        self.rnn = nn.LSTM(input_size=input_size, hidden_size=hidden_size
    , batch_first=True, bidirectional=True)
        self.fc = nn.Linear(2 * hidden_size, num_class)
    def forward(self, x):
        h0 = torch.zeros(2, x.size(0), self.hidden_size)
        c0 = torch.zeros(2, x.size(0), self.hidden_size)
        out, (h_n, c_n) = self.rnn(x, (h0, c0))
        return self.fc(out[:,-1,:])
News_model = LSTM_news_classifier_3(50, 256, 7)
```

A bidirectional LSTM layer can learn information in two directions: from past to present and from future to present. In this way, it can learn more information (Hassani). We also try to stack multiple layers of LSTM but find that it cannot improve the overall performance.

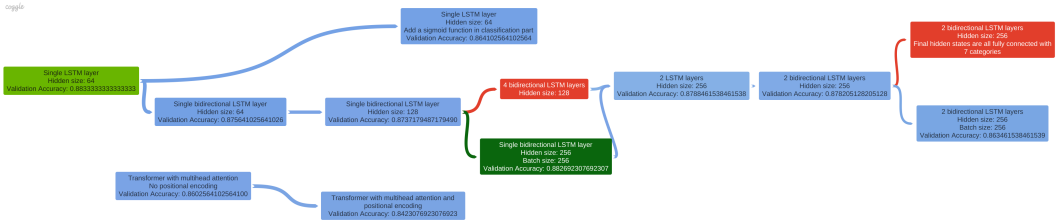
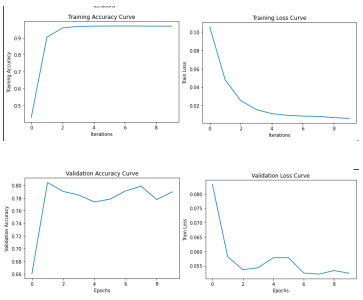


Figure 4: The process of hyperparameter tuning of each type of model, along with their validation accuracy. Red boxes represent models that have poor performance; the dark green box is our final model; the light green box shows the second best model.

6 Baseline Model

The team used a fully-connected ANN model as the baseline model. The advantage of using an ANN model is that ANN model is one of the most basic and widely used model in classification problems. Moreover, ANN model has easy-to-tune hyperparameters and flexible structure. The team tunes the number of layers, activate function and other hyperparameters, and finally made the structure of 3 hidden layers, with an activation function of ReLU. The input of the model has 61 words in one sample, each word with 50 features. The output has 7 channels corresponding to 7 classes, see in Figure 5 right.

The training result shows that the ANN model has a final training accuracy of around 97%, 78% validation accuracy, see in Figure 5 left. Qualitatively, our ANN baseline model requires a slow learning rate. If the learning rate is too fast, the validation accuracy could happen not to improve at all. Comparing our final LSTM model to the ANN model, we currently has a validation accuracy of more than 85% which is better than the ANN model, as well as the training accuracy.




```
class ANN(nn.Module):
    def __init__(self):
        super(ANN, self).__init__()
        self.layer1 = nn.Linear(50*61, 1000)
        self.layer2 = nn.Linear(1000, 200)
        self.layer3 = nn.Linear(200, 30)
        self.layer4 = nn.Linear(30, 7)
    def forward(self, img):
        flattened = img.view(-1, 50*61)
        activation1 = self.layer1(flattened)
        activation1 = F.relu(activation1)
        activation2 = self.layer2(activation1)
        activation2 = F.relu(activation2)
        activation3 = self.layer3(activation2)
        activation3 = F.relu(activation3)
        activation4 = self.layer4(activation3)
        return activation4
```

Figure 5: Training and validation curves of ANN baseline model(left), and ANN baseline model architecture(right)

7 Quantitative Results

The validation accuracy and loss of some models with different model types and hyperparameter settings are shown in table 1. The highest validation accuracy of every hyperparameter setting is also shown in figure 4.

Table 1: Validation accuracy and loss of two LSTM models and two transformers

Model type	Model name	Validation accuracy	Validation loss
LSTM	A simple LSTM (not bidirectional, single layer) Hidden size: 64	0.883333333	0.029753529
	Single bidirectional LSTM layer Hidden size: 256 Batch size: 256	0.882692308	0.029438837
Transformer with multihead attention	No positional encoding	0.86025641	0.055829914
	With positional encoding	0.842307692	0.055194185

We compare models and different hyperparameter settings mainly based on the validation accuracy. Since our datasets are balanced, we don't need to use precision or recall. From the table, we can see that LSTM has a better performance than the transformer.

Two hyperparameter settings of LSTM model coloured in light green and dark green have validation accuracy that is higher than 0.88, and it is difficult to tell which one is better since validation accuracy has some randomness, and their difference in validation accuracy is so small. We also consider the validation loss. From the table, we can see that even though the first one has a slightly higher validation accuracy, it has a higher average validation loss.

We select the second LSTM model as our final model (Single bidirectional LSTM layer, the dark green one in figure 4). Even though its validation accuracy is slightly lower, it has a bidirectional layer that can increase the depth and complexity of the model. It also has a lower average validation loss.

Figure 6 includes the training curves and validation curves of our final model. From the training curve, we can see that with a greater number of epochs, training accuracy is approaching to 1, and training loss is approaching to 0. It shows that our model could fit the training data. Validation curves have more fluctuations. It begins to overfit slowly after epoch 20. Generally, validation accuracy fluctuates around 0.85, and validation loss fluctuates around 0.035 after epoch 20.

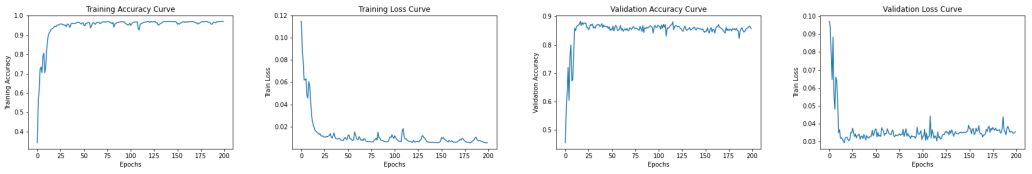


Figure 6: Training and validation curves of our final model

We perform a test on this model using the test dataset, and the test accuracy is 0.88333333. Its test accuracy has a similar value with the validation accuracy, which proves the performance of our final model is stable.

8 Qualitative Results

Our model is designed to classify different categories of news articles. Although we train it with articles with 61 words, it could also deal with news of any size. Thus, the actual application could be

extended to news titles, paragraphs in different lengths. As shown in Fig 7 and Table 2, our model perform well in length less than 61 words, and in length more than 61 words.

Figure 7: GUI for Predicted tags of text with different lengths

	News Article	Length	Prediction	Result
1	..State Livestock Development Board..	60 words	politics	True
2	..AI Can Diagnose Dementia..	13 words	science	True
3	Honda plans to launch two electric sports cars..	132 words	automobile	True
4	..Lionel Messi was omitted from the 30-man list..	205 words	sports	True

Table 2: Sample Outputs of Some Training Data

The first reason that we use the bidirectional LSTM. It could learn order-sensitive texts effectively(Zhou et al., 2016),as shown in Fig 7.

As seen in Table 2, the second reason that our model performs well, especially for paragraph in different lengths, is training data size (61 words) after normalization. News usually are short, and 61 words are enough for authors to conclude news in most situations. Otherwise, the reader can at least know the news categories. Therefore, our model could successfully learn necessary features of news under different categories within 61 words, and be able to classify news in different lengths with information learned.

However, our model may be confused between 2 special cases. All of the 2 cases are raised due to similarity in contents. First case is between technology and science news, and the second case is between politics and world news. According to Table 3, the model wrongly predicts 2 politics news as world, and make mistakes in classifying between technology and science news.

	News Article	Prediction	Ground Truth
1	..House Republicans pushed back on the FBI	world	politics
2	..or denied the legitimacy of the 2020 election	world	politics
3	Debris from a SpaceX capsule found in Australia	science	technology
4	..restores cell, organ function in pigs after death..	technology	science

Table 3: Wrong Outputs of Some Training Data

9 Evaluate Model on New Data

9.1 New Data Collection

Fifty never-before-seen news articles were collected manually: 45 are from Internet and 5 are fabricated by us. Data are formatted the same way as training data, stored in a .csv file with a news_article column containing article contents and a news_category column for its categorical labels. New Data collection is intended to be balanced with around 7 articles for each category.

For data from Internet, we choose to articles with approximately 60 words or extract roughly 60 words from a news article, which ensures our new data have similar length as training data. This is to avoid our model not able to predict accurately when input article has too few words. Similarly, the fabricated articles are also deliberately kept at around 60 words.

9.2 New Data Testing and Evaluation

Besides the testing set split from the original dataset, we also use the 50 new data to test the performance of our model. Since the dataset is small, we manually input each case to see if the output is the same as it category. This way allows us to directly observe failures or differences between predicted and actual label.

The test result on new data meets our expectations. Our model has 44 correct predictions out of 50 new data, which has exactly the same accuracy as our test split dataset of 88%. After checking each case, it is observable that our model performs well on majority cases, but minor errors will occur between technology and science, and between politics and world.

Out of 6 incorrect predictions, one is predicting technology as science, one predicting automobile as technology, one taking entertainment as world, and one taking a world news talking about Alibaba as technology. The rest two are taking politics as world news. This was as expected because we have recognized that these two pairs of categorization might be overlapping and confusing. But this error is insignificant because both category predictions are reasonable and appropriate when facing with these types of overlapping categorization.

10 Discussion

10.1 General Performance

Quantitatively, our model has a training accuracy of 93%, validation accuracy of around 88% which happens at epoch 16, and test accuracy of around 88%. From the training curve introduced before, we can observe that our model is actually learning and improving, with loss approaching to 0 and accuracy approaching to 1.

Qualitatively, among the seven classes, our model has good performances on most of the classes including sports and entertainment. However, interestingly, the model is not good at doing classification between news of world and politics. It is possible and expected, since the team learns that it could be because politics news and world news have certain similarity, sharing similar key words and overlap in some areas, which make the model get confused to these two classes. In further steps, the team my develop the 7 classes into more precise and accurate categories.

In a general scale, our model, with more than 8.5 of the tags our of 10 predictions are correct, performs well. However, it still has a certain gap from actually being used on current news media. In further steps, if the team could develop to make the test accuracy reach 90% or higher, then the model will be more professional and possible to be used on news website.

10.2 Learning and Reflection

The team has learned a lot about textual classification as well as LSTM model in this project. Textual classification is different from image classification has external relation between words and words, which requires the team to introduce hidden states or attention system to solve the problem. What's more the texts has more complex situations than where some words with abnormal frequency are obstructing the model learning correctly, such as 'a', 'the' with high frequency. The team has to introduce stop words to ignore these words.

LSTM model is performing great on textual classification, compared with most other models that the team has tested. Its gating mechanism solve the exploding or vanishing gradient problem in vanilla RNN network. The team also put some effort on transformer to see if transformer does a better job than LSTM model. However, the results is not that positive. Maybe it is because the team does not have enough time to tune hyper parameters of transformers, while it is also possible that transformer using positional encoding does not perform as well as LSTM using hidden states, on solving relationship between words.

While tuning hyper parameters of LSTM model, the team found that increasing complexity of the model may not lead to increasing accuracy. A correct tuning direction is always important. As mentioned above, the team has tried multiple ways tuning but only part of the methods work.

11 ETHICAL CONSIDERATION

The team decided to do classification of news tag, which should be neutral and objective. The tags should not orient the news to any biased field, but fairly showing the correlations of news. The quality and objectiveness of tag classifications, made by trained artificial intelligence, could be affected by labeling news in the training dataset, ineffective training model and low result accuracy.

If the training dataset, including validation and test dataset is biased, tag of which is not objective enough, the training result could be misled and oriented. There is no way that an unfair news label could train an objective model. Meanwhile, the model part also directly related to neutrality of news classification. Low test accuracy resulted in plenty errors of classification, which obviously shows and spreads false information to users.

To avoid getting into such ethical issues, the team should collect data from professional news media with high reputations to ensure the fairness of news dataset, including labels and news titles. In further steps, the team will also need to work on accuracy and effectiveness of primary training model, in order to provide neutral, accurate classifications.

12 PROJECT DIFFICULTY

In the data processing part, we find training, validation, and testing datasets by ourselves. To clean our data, we eliminate stop words in each input so that the model can be trained more effectively and accurately. To transform articles into numerical inputs, we use the Glove embedding. We also find some new data and create some data by ourselves to further test the performance of our model.

When developing and training our model, we tried two types of models: LSTM and transformer. After we finished developing those two types of models, we selected the model with the best performance. When we are developing and training LSTM models, we have much more hyperparameters tuning. (See figure 4) Here are the following ways of tuning hyperparameters.

- Changing hidden size
- Making LSTM model bidirectional
- Stacking multiple LSTM layers
- Adding an activation function
- Changing the batch size
- Using the last hidden state $\mathbf{H}_n$ as the output of the LSTM layer instead of the last output.

We also add the positional encoding in our transformer, even though it is not so effective at last.

Generally, the validation accuracy of most models in this project is around 85%, and the validation accuracy of our final model is above 88%, which means that the performance of our model is quite decent. However, I find that the validation accuracy of every model cannot reach 90%. I believe this is mainly due to the classification of news in datasets.

The news articles in datasets we find are classified into seven categories: technology, sports, politics, entertainment, world, automobile, and science. The category "world" is too broad, and the category "automobile" is too narrow. The difference between "science" and "technology" is subtle. For example, a politics news article could be both a world news article and a politics news article, so in most cases, our model predicts a politics news article as a world news article. Another example is that a science news article is often predicted as a technology news article, and vice versa. The way of news classification could hinder the development of our model.

REFERENCES

Pandas. https://pandas.pydata.org/docs/index.html, 2022.

Contributor. The importance of tags in online news media, Jul 2021. URL https://whatsnewinpublishing.com/the-importance-of-tags-in-online-news-media/.

Inoshika Dilrukshi, Kasun De Zoysa, and Amitha Caldera. Twitter news classification using svm. In *2013 8th International Conference on Computer Science Education*, pp. 287–291, 2013. doi: 10.1109/ICCSE.2013.6553926.

Kaveh Hassani. Week 8 recurrent neural networks - part ii.

Sandeep Kaur and Navdeep Kaur Khiva. Online news classification using deep learning technique. *International Research Journal of Engineering and Technology (IRJET)*, 3(10):558–563, 2016.

Larsyencken. stopwords.txt. https://gist.github.com/larsyencken/1440509#file-stopwords-txt, 2011.

Andrew Morgan. Personalized notifications at twitter. https://www.infoq.com/news/2017/06/personalized-twitter/, June 2017.

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

Joni Salminen, Vignesh Yoganathan, Juan Corporan, Bernard J. Jansen, and Soon-Gyo Jung. Machine learning approach to auto-tagging online content for content marketing efficiency: A comparative analysis between methods and content type. *Journal of Business Research*, 101:203–217, 2019. ISSN 0148-2963. doi: https://doi.org/10.1016/j.jbusres.2019.04.018. URL https://www.sciencedirect.com/science/article/pii/S0148296319302607.

Kishan Yadav. News classification. https://www.kaggle.com/datasets/kishanyadav/inshort-news, 2021.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*, 2016.