Spring 2022
Introduction to Artificial Intelligence

Homework 2

Mar. 22, 2022

Introduction

The goal of this programming assignment is to 1) learn how to preprocess a text dataset for sentiment classification, 2) implement a basic language model, **bi-gram**, 3) get a taste of a complete supervised learning process, and 4) compare the results and performance with SOTA method (DistilBert in this assignment). Please make sure you understand the concept of n-gram and BERT before working on HW2.

The details of each file are listed at the end of this document. **Please finish codes between # Begin your code and # End your code**. Feel free to add additional functions if you need them, but note that do not modify the existing codes. Note that you should implement the required function in each part by yourself instead of using existing methods. Feel free to ask if you are not sure about this.

Data

The dataset in this assignment is the IMDB dataset, which consists of 50k movie reviews. You can find it in the “data” folder. The example and statistics of the dataset are as below.

	Train	Test
Positive	20,000	5,000
Negative	20,000	5,000
Total	40,000	10,000

Review	**Sentiment**
Saw the move while in Paris in May 2006 ... It is important to have some understanding the French society of Today to really enjoy the humor of this movie…	Positive
Scary Movie 2 was a grave disappointment. Simply referencing movies, like Mission Impossible 2 does not lead to comedy …Overall, a very poorly done movie.	Negative

Requirements

Part 0: Implement at least three different preprocessing methods (15%)

- We provide the code of removing stopwords in preprocess.py.
- You need to implement the other three preprocessing methods in the "preprocessing_function" function (e.g., stemming).
- You can use the existing list you want (e.g., stopwords provided by nltk).
- Hint: Try to observe the dataset and you will find there are some symbols that can be eliminated.

Part 1: Implement the bi-gram language model (25%)

- Implement the bi-gram model in the "get_ngram" function in model.py.
- You may need to modify the outputs in the "train" function, but you need to keep self.model at least.
- You can run model.py solely for checking the correctness of your implementation with the simple input. We also provide an explanation in the code for reference only.

Part 2: Implement perplexity computation (10%)

- Compute the perplexity of testing data in the "compute_perplexity" in model.py.
  - Use **base 2** when doing logarithm.
- You can run model.py solely for checking the correctness of your implementation with the simple input. It is easy to check by calculating by hand.
- You will encounter zero division error, state how you solve this issue in the report.

Part 3: Implement feature selection and conversion (10%)

- To fine-tune for sentiment classification, you need to convert each sentence to the corresponding embedding for training a Naive Bayes classifier. However, if we use all pairs as features, we need huge memory and the dimension would be sparse.
- Thus, you should use only the number of features feature_num and convert the sentence to embedding. The size of train_corpus_embedding is (n_samples_train, n_features) and the size of test_corpus_embedding is (n_samples_test, n_features).
- Here is an example. Assume the features are [(I saw), (saw a), (an apple)], the embedding of the tokenized sentence ['[CLS]', 'I', 'saw', 'a', 'saw', 'saw', 'a', 'saw', '.'] will be [1, 2, 0] since the bi-gram of the sentence contains [([CLS] I), (I saw), (saw a), (a saw), (saw saw), (saw a), (a saw), (saw .)] The number of (I saw) is 1, the number of (saw a) is 2, and the number of (an apple) is 0.

Part 4: Implement other types of n-gram models (Bonus) (15%)

- Try to implement other types of n-gram models (e.g., uni-gram, tri-gram, and bi-directional n-gram) and observe the performance.
- You need to discuss your results in the report.

Report (40%)

- The goal of writing a report is to learn to analyze the questions and your observations. We rate your report mainly based on the discussions and analysis. You **don't need to paste your code and explanation of your code** in this assignment, but you will only get some scores (or even zero scores) if your code cannot run successfully as the requirements.
- You are required to submit a report and it can be written in Chinese or English.
- Save the report as a **.pdf** file
- The report should **at least** include the following items.
  1. Give an input and output example after applying each preprocessing method in the report. For example, the input sentence is “Here is the dog.” and the output after removing stopwords is “Here dog.”.
  2. The performance of using different numbers of feature_num in sentiment classification.
  3. Discuss what you observed with perplexity, F1-score, precision, and recall of different methods in the report.
  4. Discuss the difference between the bi-gram model and DistilBert (a lightening variant of BERT). You might need to run extra experiments if you give some hypotheses. Some required discussions are given below.
     a. What are the reasons you think bi-gram cannot outperform DistilBert?
        i. State how to win if you outperform DistilBert. TAs would like to see your methods!
     b. Can bi-gram consider long-term dependencies? Why or why not?
     c. Would the preprocessing methods improve the performance of the bi-gram model? Why or why not?
     d. If you convert all words that appeared less than 10 times as [UNK] (a special symbol for out-of-vocabulary words), would it in general increase or decrease the perplexity on the previously unseen data compared to an approach that converts only a fraction of the words that appeared just once as [UNK]? Why or why not?
  5. Describe problems you meet and how you solve them.

Discussion

TAs had opened a channel HW2 討論區 on Microsoft Teams of the course, you can ask questions about the homework in the channel. TAs will answer questions in the channel as soon as possible.

Discussion rules:

1. Do not ask for the answer to the homework.
2. Check if someone has asked the question you have before asking.
3. We encourage you to answer other students' questions, but again, do not give the answer to the homework. Reply to the messages to answer questions.

4. Since we have this discussion channel, do not send emails to ask questions about the homework unless the questions are personal and you do not want to ask publicly.

Submission

1. **The deadline for this homework is 4/11 (Mon.) 23:55:00.**
2. Please submit one zip file that contains all the Python code files (i.e., main.py, preprocess.py, model.py, bert.py), and report with the format hw2_{StudentID}.pdf (e.g., hw2_109123456.pdf). Note that you should not include the dataset in your zip file.
3. Submit the zip file with the filename of hw2_{StudentID}.zip (e.g., hw2_109123456.zip).
4. Late submission leads to a score of (original score)*0.85^{days}, for example, if you submit your homework right after the deadline, you will get (original score)*0.85 points.
5. **We only accept one zip file**, wrong format, or naming format cause -10 points to your score (after considering late submission penalty).
6. TA will run your code file automatically by "run.sh". The environment will be Ubuntu 20.04 and Python 3.8, which is the same as the environment you had prepared in HW0. **Make sure your code can run in this environment correctly.**
7. **Plagiarism** is not allowed! You will get a huge penalty if we find that.
8. If there is anything you are not sure about submission, ask in the discussion forum.

Files

File name	**Description**
main.py	The main code of this assignment.
preprocess.py	The code preprocessing the text.
model.py	The code for the n-gram model. You can run this code solely for unit tests.
bert.py	The code with DistilBert for sentiment classification.
run.sh	The script for testing this assignment.
data folder	Train, test datasets.

References

- Introduction to Bert by Prof. Hung-Yi Lee.
- [DLHLP 2020] Language Modeling by Prof. Hung-Yi Lee.
- Language Modeling post by Lena Voita.
- Beyond Computing 講座 (2022/04/09)