CSCI 5541: Homework 3

Manpreet Singh
sing1174@umn.edu

1 Generative Classifier

1.1 Encoding Type:

UTF-8 encoding type was chosen its the recommended one and supports a wider range of characters.

1.2 Information in the Model:

The authors' generative models leverages NLTK's **word_tokenize** function to convert sentences into sequences of words, which can then be processed for n-gram modeling.

Padding with NLTK function **pad_both_ends** adds special tokens to the beginning and end of sequences to ensure that all sequences are of a uniform length and that the model can learn context better.

The function **padded_everygram_pipeline**, generates all possible n-grams from the padded tokenized sentences, enabling the author model to learn the likelihood of word combinations.

Each author model is designed to work with unigrams, bigrams as well as trigrams using (n=3) by default. However, it can be easily adjusted to n-grams for (n=2 or n=1) as well. The models capture the conditional probabilities of sequences of words based on the specified n-gram size.

1.3 Smoothing/Interpolation/Backoff:

Initially, **MLE** author models were tested on the development set, however, the accuracies on dev set were below random chance (i.e. below 25%). So a number of smoothing, interpolation and backoff models (Toolkit, 2024b) were trained and tested.

- **Laplace Smoothing:** adds one to the count of each n-gram, ensuring that no n-gram has a probability of zero, which helps avoid undefined probabilities.

- **Lidstone Smoothing:** allows for a customizable smoothing parameter (gamma). In the code, it's initialized with a gamma value of 0.05.

- **Witten-Bell Interpolation:** this combines counts from observed and unobserved events based on their context.

- **Stupid Backoff:** backs off to lower-order n-grams when model encounters an unseen n-grams

1.4 Results in development mode

The table below shows the results on development set of each of the four author models. Results from the four different types of generative models are reported:

Author	Laplace	Lidstone	Witten-Bell	Stupid Back-off
Austen	79.85%	69.31%	53.85%	51.23%
Dickens	57.12%	67.58%	58.65%	55.27%
Tolstoy	54.11%	58.87%	54.78%	53.13%
Wilde	50.21%	55.73%	51.73%	50.43%

Table 1: Accuracy results of author models on the development set

1.5 Top 5 representative features for each N-gram model

Table 2 shows the top 5 most representative features of the **StupidBackoff** author models. The top 5 n-grams along with the model's probability scores are displayed in the table.

1.6 Failure Cases for generative models

Author classification task : A few failure cases for the Laplace n-gram model are shown in Table 3. The failure cases indicate that the model

Author	Features	Probability Score
Austen	(’the’, ’night’, ’before’)	1.0
	(’fall’, ’in’, ’love’)	1.0
	(’i’, ’don’, ’”)	1.0
	(’don’, ’”, ’t’)	1.0
	(’,’, ’remember’, ’,’)	1.0
Dickens	(’fall’, ’in’, ’love’)	1.0
	(’i’, ’don’, ’”)	1.0
	(’don’, ’”, ’t’)	1.0
	(’,’, ’remember’, ’,’)	1.0
	(’”, ’it’, ’was’)	1.0
Tolstoy	(’i’, ’don’, ’”)	1.0
	(’don’, ’”, ’t’)	1.0
	(’minister’, ’and’, ’his’)	1.0
	(’‘’, ’i’, ’”)	1.0
	(’can’, ’”, ’t’)	1.0
Wilde	(’curtain’, ’was’, ’lying’)	1.0
	(’lying’, ’.’, ’he’)	1.0
	(’how’, ’you’, ’men’)	1.0
	(’can’, ’fall’, ’in’)	1.0
	(’fall’, ’in’, ’love’)	1.0

Table 2: Top 5 features of author models with probability scores for each n-gram.

Sentence	Predicted Author	True Author
“To whom?”	Tolstoy	Wilde
‘I know that, but you might tell me.’	Dickens	Wilde
must be a good-looking chap.”	Austen	Wilde
“Yes; but I can’t help it. You don’t know what I have suffered waiting	Wilde	Tolstoy
little, I mean a good deal, a great deal—forty three thousand.”	Austen	Tolstoy

Table 3: Failure cases for author prediction Laplace n-gram model

misclassifies sentences particularly when dealing with common expressions or short sentences. Sentences that lack context, distinctiveness or share similar styles across authors result in higher misclassification rates, as seen in Wilde being confused with Tolstoy and Austen.

2 Discriminative Classifier

The primary objective of the Discriminative Authorship Classifier is to accurately predict the author of a given sentence among Austen, Dickens, Tolstoy or Wilde. Thus we have a total of **k=4 labels** due the above four authors present. **Huggingface pre-trained tokenizer DistilBERT** (Sanh et al., 2020) has been finetuned on the author’s text files to get the sequence classification model.

2.1 Data Preparation

For preparing the train and evaluation dataset, first **we map the author labels from string to numeric** values. Then utilizing sklearn’s train_test_split, the labelled dataset is split into train set with 90% data and eval set with the remaining 10%.

To prepare the text data for input into the model, the sentences are tokenized using pre-trained HuggingFace tokenizer i.e **DistilBERT**. Finally, the tokenized datasets are converted into the appropriate format for PyTorch, specifying the columns to be used as input (input_ids, attention_mask) and output (label).

Metric	Value
Number of examples	5600
Batch size	16
Evaluation loss	0.6958
Evaluation accuracy	71.32%
Evaluation runtime	32.47 seconds
Evaluation samples per second	172.49
Evaluation steps per second	10.78
Epoch	1.0

Table 4: Development Set Evaluation Results.

2.2 Training with Huggingface Trainer

For training the discriminative classifier on author prediction, the following hyperparameters were employed.

- Batch Size for train and eval set: 16
- Learning Rate: 3e-5
- Optimizer: AdamW
- Weight Decay: 0.01
- Number of epochs: 1
- Learning rate scheduler: linear

2.3 Results in development mode

In the development mode, there were **5600 samples (10%)** of entire dataset in evaluation set, and with 1 epoch of training the **evaluation accuracy was 71.32%**.

2.4 Analysis of Failure cases - Discriminative model

Table 5 below shows a few examples of incorrect author predictions by the Discriminative classifier.

The misclassifications highlights a few key issues with the discriminative classifier's performance in predicting the correct author:

- **Inability to Capture Nuances:** Model struggles to grasp the subtle differences in writing styles and themes among the authors. For instance, Jane Austen's commentary on social norms is often mistaken for the more dramatic tones of Charles Dickens.
- **Short Inputs and Ambiguity:** Short sentences can pose significant challenges, particularly when it lacks context. A phrase like “shocking affair” could fit various narratives depending on its surrounding content.

Input Text	True Author	Predicted Author
beau, Nancy,' my cousin said t'other day, when she saw him crossing the	Austen	Dickens
shocking affair.	Austen	Tolstoy
“"We shall have to hurry," he said, "the train won't wait."	Dickens	Tolstoy
"Is there any other man like him?" she asked with a smirk.	Wilde	Austen
"She had been silent, but not because she was afraid."	Tolstoy	Dickens
His health was failing, and every day seemed longer than the last.	Tolstoy	Austen
“"You must marry her, there is no other way."	Dickens	Austen
“"We are none of us perfect, sir, but we do our best."	Dickens	Tolstoy
“He was a strange, lonely man, with no desire to fit in with society."	Wilde	Dickens

Table 5: Failure Cases for Author Prediction by Discriminative classifier

- **Contextual Overlap:** Many classic authors explore similar linguistic themes, such as love, morality, and society and the model struggles to distinguish between them.

2.5 Discriminative models analysis

Discriminative models focus exclusively on learning the **conditional probability** $P(Y|X)$ which directly predicts the label Y given the input features X. These models emphasize the decision boundary between classes rather than the underlying data distribution.

Few examples of success cases for both generative and discriminative models are shown in Tables 6 and 7.

3 Acknowledgements

I used insights from ChatGPT (OpenAI, 2024), NLTK libraries (Toolkit, 2024b), (Toolkit, 2024a) and other resources referenced below. I also discussed some basic questions about this homework,

Sentence	Predicted Author	True Author
cast for the girls' parts, and when *As You Like It* was produced he	wilde	wilde
"If you are refusing for my sake, I am afraid that I..."	tolstoy	tolstoy
of respectability, unaffected by the east wind of January, and not	dickens	dickens
and the interesting employment had followed, of reckoning up exactly	austen	austen

Table 6: Success cases for NLTK Author Models

Sentence	Predicted Author	True Author
money matter? Love is more than money."	wilde	wilde
"Where are you off to? Stay a little longer," he said to Varenka.	tolstoy	tolstoy
night sky, concentrated into a faint hair-breadth line. So does a whole	dickens	dickens
indeed, though my mother's eyes are not so good as they were, she can	austen	austen

Table 7: Success cases for Discriminative Model

with my classmate, *Sharan Rajamanoharan*, to better understand the project requirements.

References

OpenAI. 2024. Chatgpt: Artificial intelligence language model.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Natural Language Toolkit. 2024a. nltk api documentation. `https://www.nltk.org/api/nltk.html`. Accessed: 2024-10-08.

Natural Language Toolkit. 2024b. nltk.lm.smoothing documentation. `https://www.nltk.org/api/nltk.lm.smoothing.html`. Accessed: 2024-10-08.