QUESTION 1:

Intrinsic Evaluation: Intrinsic evaluations measure the performance of an NLP component on its defined subtask

Extrinsic Evaluation: Extrinsic evaluations focus on the component's contribution to the performance of a complete application, which often involves the participation of a human in the loop.

- How do they perform intrinsic evaluation? Spell out the task (specifically, the input and output: what problem are they solving?)

Input:Tweet it appears in as context.

Output:To explicitly predict the function of a vulgar word.

What problem are they solving? : To identify one of its six functions**(**Aggression, Express Emotion, Emphasis, Auxiliary, Signal Group Identity, Non-Vulgar)

Method: Logistic regression is used to binary classifiers for each of the six functions using information from the immediate lexical and syntactic context surrounding the word and general usage of the word in training data.

Results: Our predictive model vastly outperforms the most frequent baseline, which uniformly selects the most frequent class overall (emphasis) and scores very low due to the very even distribution over functions. Our best model achieves a macro- averaged **F1 score of 67.4** across the six classes.

This is **Intrinsic** because it achieves a score against a defined standard.

- How do they perform extrinsic evaluation? Spell out the task (specifically, the input and output: what problem are they solving?). Why is it an extrinsic evaluation?

Input: Dataset with hate speech, offensive, and neither as three classes containing tweets collected using vulgar words.

Output:To predict tweets containing hate speech.

What problem are they solving?: To explicitly model the function a vulgar word has in context which benefits the hate speech prediction task, by differentiating between aggression and other usages.

Method: We thus train three one-vs-all logistic regression classifiers with L2 regularization.
We directly and explicitly include the function of the vulgar word present in the tweet by introducing six new features to the hate speech detection model which represent the scores with

which the vulgar word is associated with the six functions. If multiple vulgar words exist in a tweet, we use the average predictions over the six functions.

Results: The addition of the six vulgar function features improves the F1 score for each of three classes to 6.1 F1 for the hate speech class, which had the lowest performance. This results in an improvement of the macro-F1 score for the entire classification task of **3.7 in F1.** This demonstrates the importance of the proposed vulgar function modeling task in detecting hate speech.

This is **Extrinsic** because we directly and explicitly include the function of the vulgar word present in the tweet

QUESTION 2:

3.2 N-gram Frequency Cutoff

> The counts in Section 5 of the README file, however, state that the count of 4-grams is larger than the count of 5-grams.
>
> This is because of the N-gram Frequency Cutoff. Considering N-grams that are appearing more than 40 times were kept.Only those appear in the n-gram tables.
>
> N-grams that have less frequency were discarded.

QUESTION 3:

- P(glory) is much lower than the probability of most words in plays of the period.

Answer: False

According to the passage, it is given that “the word glory is not all that unusual in plays of the period”. This means that the word glory is usual in plays.It is one of the most words in plays of the period.Therefore, P(glory) is not much lower than the probability of most words in the plays of the period.

- There is only one author for which P(droopeth) is not zero.

Answer:False

According to the passage, it is given that “And the verb droopeth, you know, it occurs in a number of different writers.” This means that the word droopeth is used by more than one writer and probability is not zero. Therefore there is not only one author for which P(droopeth) is not Zero.

- P(droopeth | glory) is higher in the corpus of Shakespeare’s plays than in the corpus of Marlowe’s plays.

Answer: False

The word droopeth after glory(glory droopeth) occurs in one of these disputed passages in “Henry VI, Part 1 and in a play by Marlowe. So the P(droopeth | glory) is not higher in Shakespeare's corpus than in Marlowe’s corpus.

QUESTION 4 REPORT:

Challenges:

1.Calclation of Bigram probability

Had to use a dictionary instead of Array as arrays take much computational time for searching a character using index. By dictionary we can use key value pairs.

2.Prediction using test data.

Checked probabilities for every bigram characters data and compared with existing models that were created using train data.

3.Smoothing.

Smoothing is done for both test(only unseen data) and training data to achieve good performance.

- Do you think it makes sense to create a language model at the character level instead of at the word level for this task? Why?

Yes.
Creating a character level language model rather than word level saves a lot of storage with minimum processing.Words made of vocabulary are large in number but characters are not. Word generative models have thousands of inputs but character models have approximately 40 potential units which have letters,punctuations, spaces and numbers.

Since we need to learn a parameter for every one of those inputs, this makes the model much larger and hard to train.

Creating a word language model requires a lot of memory which results in higher computational cost than the character level model.
Character level models can generate unusual words with some probability when compared with word level considering misspelled words and tokens that aren’t in corpus.Word level model cannot generate new text as the character level model does.

Word generation models have less training data but character models have each character as a separate unit. This helps the machine to get trained well and perform accurately with testing data.

- Take a look at the test documents for English and Spanish. Are documents written only in one language?

No. The Spanish corpus has English words. English words like Author, Title , release date are present in given spanish test corpus data. This saves a lot of storage with minimum processing.

- What is the minimum number of tokens you need to process to always make the right prediction during testing? You can try with 100 tokens, 200 tokens, 300, etc. instead of the whole document. You do not need an exact number. (Just like humans, your software doesn't need to read the whole book to figure out whether it is written in English or Spanish, assuming books are written in one language).

Minimum number of tokens to predict english language are 400 tokens and spanish are 300 tokens.

- If you create several models for English and Spanish (using different training data, using different preprocessing, etc.), how can you compare them? Does your answer change if all the models correctly predict the right language? In other words, how would you declare a winner if two models always predict the correct language?

Creating and training a model with different and large data results in improving accuracy and performance. For example, if we create a bigram model and trigram model, the bigram model has more training data when compared with the trigram model as the corpus is tokenized into two words. Trigram has less data for training and effects in accuracy and performance.

Comparison is made by measuring the performance of two models.

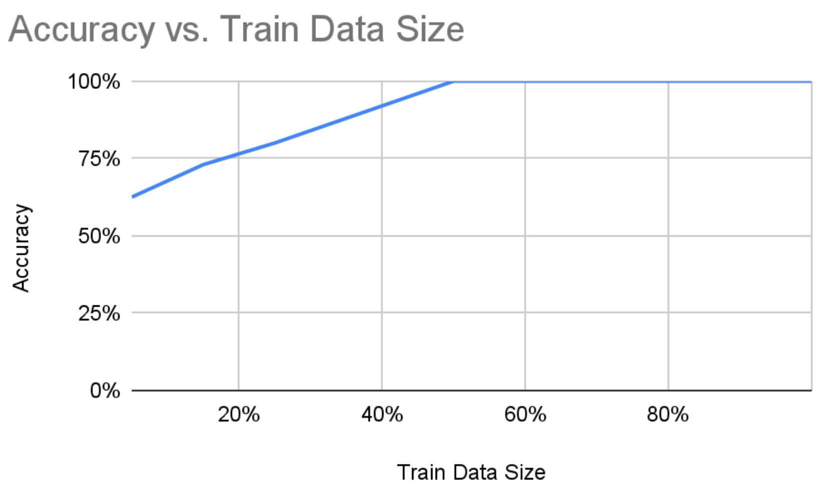
If two models always predict the correct language I would declare the winner by measuring the performance and computational time of the models.

- Can you train with less training data and still get the right predictions? How does training size affect predictions during testing? A graph showing the amount of training data and accuracy of your language models is probably the best way to answer these questions (and a short interpretation of the graph).

  Yes we can train with less training data but there may be a difference in the predictions.This can be achieved through repeatedly training the model with different subsets of training data. It may be helpful to achieve good performance.

  But in this case spanish corpus has few english words too. In order to train the model it might require an average training data where the model can predict language by calculating and comparing probabilities.

Accuracy Vs Train Data Size GRAPH:

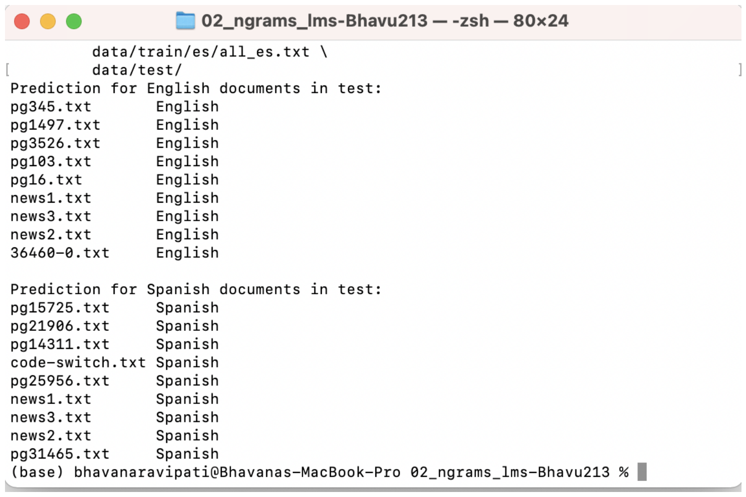


- GetsomedocumentwritteninFrench(e.g.,http://www.gutenberg.org/files/36460/36460-0. txt) and use your models to predict the language. Can you justify the prediction?

The model predicted the test data **36460.txt** (French file) as English.

This is due to the model getting trained by the English corpus. Since the file has english words, the model has predicted the french file as English.

English and French have 40-50% lexical similarity.



```
         data/train/es/all_es.txt \
[        data/test/                                                             ]
Prediction for English documents in test:
pg345.txt       English
pg1497.txt      English
pg3526.txt      English
pg103.txt       English
pg16.txt        English
news1.txt       English
news3.txt       English
news2.txt       English
36460-0.txt     English

Prediction for Spanish documents in test:
pg15725.txt     Spanish
pg21906.txt     Spanish
pg14311.txt     Spanish
code-switch.txt Spanish
pg25956.txt     Spanish
news1.txt       Spanish
news3.txt       Spanish
news2.txt       Spanish
pg31465.txt     Spanish
(base) bhavanaravipati@Bhavanas-MacBook-Pro 02_ngrams_lms-Bhavu213 %
```

QUESTION 5

- What is code switching?

Code-switching (CS) is a linguistic phenomenon defined as “the alternation of two languages within a single dis- course, sentence or constituent.” In this phenomenon,multilingual speakers switch back and forth between their common languages in written or spoken communication.

- Why does code switching pose an issue for language models?

  Code switching pose an issue for language models is challenging for three Reasons

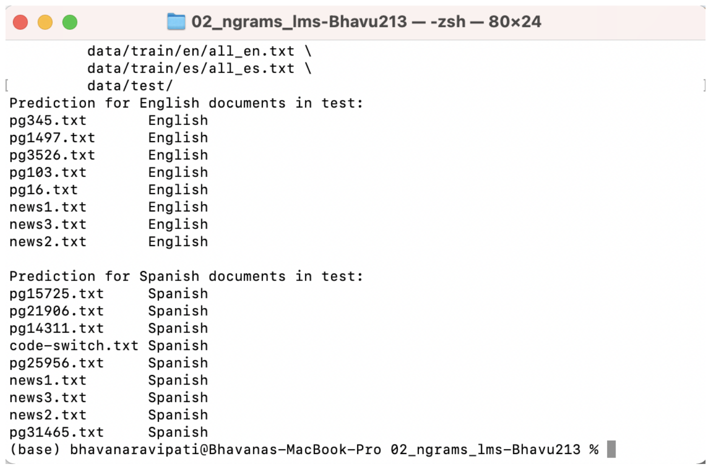
  1.Lack of available large scale code-switched data for training.

  2.Lack of replicable evaluation setup.

  3.The reliance on generative modeling.

  In the context of Automated Speech Recognition(ASR), these challenges can be tackled by proposing an ASR- motivated evaluation setup which is decoupled from an ASR system and the choice of vo-cabulary, and provide an evaluation dataset for English-Spanish code-switching. This setup lends itself to a discriminative training ap- proach, which we demonstrate to work better than generative language modeling. Finally, we explore a variety of training protocols and verify the effectiveness of training with large amounts of monolingual data followed by fine- tuning with small amounts of code-switched data, for both the generative and discriminative cases.

- What would your language detector predict if you give it the sentences in the first column of Table 1?
  Language detector has predicted the sentences in the first column of Table 1 as Spanish. **File name(code-switch.txt)**


02_ngrams_lms-Bhavu213 — -zsh — 80×24

```
        data/train/en/all_en.txt \
        data/train/es/all_es.txt \
        data/test/
Prediction for English documents in test:
pg345.txt       English
pg1497.txt      English
pg3526.txt      English
pg103.txt       English
pg16.txt        English
news1.txt       English
news3.txt       English
news2.txt       English

Prediction for Spanish documents in test:
pg15725.txt     Spanish
pg21906.txt     Spanish
pg14311.txt     Spanish
code-switch.txt Spanish
pg25956.txt     Spanish
news1.txt       Spanish
news3.txt       Spanish
news2.txt       Spanish
pg31465.txt     Spanish
(base) bhavanaravipati@Bhavanas-MacBook-Pro 02_ngrams_lms-Bhavu213 %
```