COMP 472
ARTIFICIAL INTELLIGENCE

MINI-PROJECT 3 REPORT

Submitted to
Professor Leila Kossom

Submitted by
Ryan Nichols 29787739
Jiayin Liu 27532628

December 4th, 2018

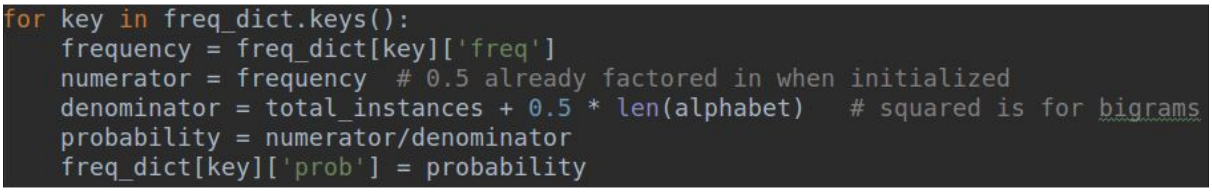
Experimental Setup

The experiment is performed using Python3 with no special machine learning libraries required. The provided training texts along with our own addition materials were parsed and data on term frequencies was recorded in data structures. From that data the probabilities were calculated. From the probabilities the most likely language of both the provided sentences and our own additional sentences was determined.

Unigram

To calculate and store the data for the unigram models nested python dictionaries were used. Each letter of the alphabet was used as a key in the outer dict and the inner dict's keys were 'freq' and 'prob' for frequency and probability. Frequencies were initialized to 0.5 for smoothing and while parsing through the training corpus for a particular language each occurance of each letter caused the frequency for that letter to be incremented.

When this process finished the frequencies for all letters were added up to get the total character frequency of the training corpus. To get the letter's probability the frequency was then divided by the total character frequency with delta smoothing technique[1] being used as picture in the image below.



```
for key in freq_dict.keys():
    frequency = freq_dict[key]['freq']
    numerator = frequency  # 0.5 already factored in when initialized
    denominator = total_instances + 0.5 * len(alphabet)   # squared is for bigrams
    probability = numerator/denominator
    freq_dict[key]['prob'] = probability
```

To calculate the language of highest probability for each sentence the characters were examined in order while the cumulative sum of their logs was calculated yielding a negative number. This is to be expected from knowledge of the shape of the log graph for small values on the x-axis. The highest number for the sum of logs determines the most likely language.

Bigram

The calculation of Bigram model is performed using the tool *itertools.product()* that takes 2 lists of letters and produce all the possible cross product combinations. Then, while iterating through the language corpora (a combination of 2 literature books in said language), the Bigram algorithm outputs the language model by transforming each line to lowercase characters, stripping all punctuations, then breaking down the line of string into pairs of consecutive letters, and finally counting the letter pairs. Then, using this produced language model, the Bigram probability for a random sentence is computed using an similar fashion: first transform the sentence into a string of lower case characters without punctuation, then for every pair of consecutive letters, compute the P(w|c) of of said pair.

```
for language in results:
  P_w_c = math.log10((results[language]['bigram_model'][pair] + DELTA)/results[language]['sum'])
  results[language]['probability'] += P_w_c
```

[1] Slide 50, class notes on NLP

This formula calculates the total log sum of all possible pairs for each language loaded in the bigram model. The calculation is smoothened using a delta of 0.5 and a vocabulary size of 26*26. In the formula above, the vocabulary size is already added into the [‘sum’] value.

Choice of 3rd Language

The 3rd language is chosen to be Spanish, since it deals with the same 26 character set as English and French, also because it is very close to French due to both of them being rooted in Latin, the unigram and especially bigram models for these languages could be very similar. It would be interesting to observe the language estimation for sentences that are highly alike in grammar and orthography.

Results Analysis

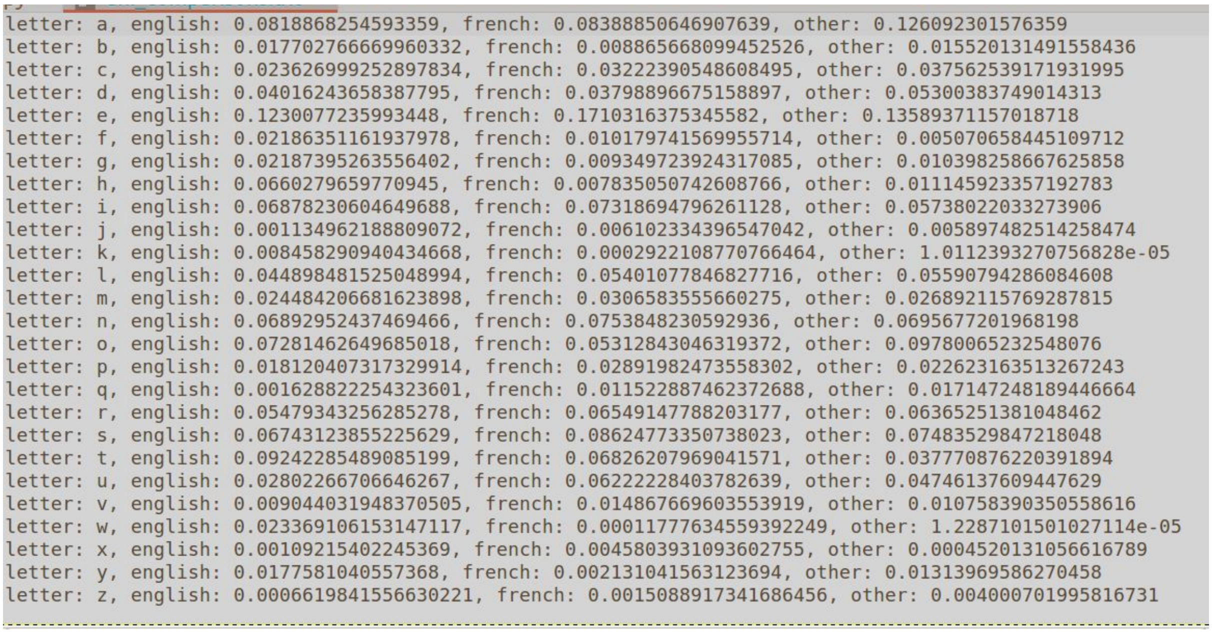
Most Probable Language

	Sentence	Unigram	Bigram
1	What will the Japanese economy be like next year?	ENGLISH	ENGLISH
2	She asked him if he was a student at this school.	ENGLISH	ENGLISH
3	I'm OK.	ENGLISH	ENGLISH
4	Birds build nests.	FRENCH	ENGLISH
5	I hate AI.	ENGLISH	ENGLISH
6	L'oiseau vole.	OTHER	FRENCH
7	Woody Allen parle.	ENGLISH	ENGLISH
8	Est-ce que l'arbitre est la?	FRENCH	OTHER
9	Cette phrase est en anglais.	ENGLISH	FRENCH
10	Where is she going?	ENGLISH	ENGLISH
12	What is the cause?	ENGLISH	ENGLISH
12	How does that work?	ENGLISH	ENGLISH
13	What is the point?	ENGLISH	ENGLISH
14	We're going to the restaurant.	ENGLISH	ENGLISH
15	I'm happy to help.	ENGLISH	ENGLISH
16	This is a good project.	ENGLISH	ENGLISH
17	There has to be a way.	ENGLISH	ENGLISH
18	The job is not done.	ENGLISH	ENGLISH
19	The pay is too little.	ENGLISH	ENGLISH
20	Let's meet at noon.	FRENCH	ENGLISH
21	It's ten.	FRENCH	FRENCH
22	It's more.	FRENCH	FRENCH
23	It's over.	FRENCH	ENGLISH

24	Sell it.	FRENCH	FRENCH
25	Sell it for me.	FRENCH	ENGLISH
26	zebra's earn jack.	FRENCH	ENGLISH
27	Jill earns a lot.	OTHER	ENGLISH
28	Jill is a zenith.	FRENCH	ENGLISH
29	Jump for joy.	OTHER	OTHER
30	J'aime l'IA.	FRENCH	FRENCH
30	Claro que si.	OTHER	OTHER
31	Tengo hambre vamos a por una hamburguesa	OTHER	OTHER
32	C'est pour faire parler les curieux.	FRENCH	FRENCH

We can see that the bigram models are more accurate than the unigram models which is to be expected since there is more information available when considering character pairs rather than just characters. When you have word pairs you also have order associated with them which again is more information than just singular characters.

Whether or not the unigram model can correctly identify a sentence depends on the letters making up that sentence. If there are many letters with a relatively high probability for the correct language when compared with an incorrect language the sentence will be correctly identified. It was difficult to find sentences that the unigram model would incorrectly identify so we made a comparison of the unigram models to find individual characters that may have a positive impact on whether or not the sentence was identified right or wrong as needed. The comparison is pictured in the image below.



```
letter: a, english: 0.0818868254593359, french: 0.08388850646907639, other: 0.126092301576359
letter: b, english: 0.017702766669960332, french: 0.008865668099452526, other: 0.015520131491558436
letter: c, english: 0.023626999252897834, french: 0.03222390548608495, other: 0.037562539171931995
letter: d, english: 0.04016243658387795, french: 0.03798896675158897, other: 0.05300383749014313
letter: e, english: 0.1230077235993448, french: 0.1710316375345582, other: 0.13589371157018718
letter: f, english: 0.02186351161937978, french: 0.010179741569955714, other: 0.005070658445109712
letter: g, english: 0.02187395263556402, french: 0.009349723924317085, other: 0.010398258667625858
letter: h, english: 0.0660279659770945, french: 0.007835050742608766, other: 0.011145923357192783
letter: i, english: 0.06878230604649688, french: 0.07318694796261128, other: 0.05738022033273906
letter: j, english: 0.001134962188809072, french: 0.006102334396547042, other: 0.005897482514258474
letter: k, english: 0.008458290940434668, french: 0.0002922108770766464, other: 1.0112393270756828e-05
letter: l, english: 0.044898481525048994, french: 0.05401077846827716, other: 0.05590794286084608
letter: m, english: 0.024484206681623898, french: 0.0306583555660275, other: 0.026892115769287815
letter: n, english: 0.06892952437469466, french: 0.0753848230592936, other: 0.0695677201968198
letter: o, english: 0.07281462649685018, french: 0.05312843046319372, other: 0.09780065232548076
letter: p, english: 0.018120407317329914, french: 0.02891982473558302, other: 0.022623163513267243
letter: q, english: 0.001628822254323601, french: 0.011522887462372688, other: 0.017147248189446664
letter: r, english: 0.05479343256285278, french: 0.06549147788203177, other: 0.06365251381048462
letter: s, english: 0.06743123855225629, french: 0.08624773350738023, other: 0.07483529847218048
letter: t, english: 0.09242285489085199, french: 0.06826207969041571, other: 0.037770876220391894
letter: u, english: 0.02802266706646267, french: 0.06222228403782639, other: 0.04746137609447629
letter: v, english: 0.009044031948370505, french: 0.014867669603553919, other: 0.010758390350558616
letter: w, english: 0.023369106153147117, french: 0.00011777634559392249, other: 1.2287101501027114e-05
letter: x, english: 0.00109215402245369, french: 0.0045803931093602755, other: 0.0004520131056616789
letter: y, english: 0.0177581040557368, french: 0.002131041563123694, other: 0.01313969586270458
letter: z, english: 0.0006619841556630221, french: 0.0015088917341686456, other: 0.004000701995816731
```

If you look to the letter j it's contributes to the likelihood of giving a French classification 6 times more so than English. Therefore when coming up with sentences that would be incorrectly identified written in English, many of them contain a j. In this way each character in the sentence contributes to a correct or incorrect classification.

Experiments

The experiment Bigram model takes into consideration all possible combinations of 2 alphabets, including the instance *<s>*|*character* since the appearance of a single letter as word is specific to the language. For example, in English, the possibility of a single letter 'a' appearing is much higher than that in French due to its vast usage as a pronoun. Conversely, the instance of *character*|*<s>* is not taken into account, firstly due to duplication with the instance *character*+*space*, secondly because this combination mainly checks for single letter trailing at the end of a sentence, which is a situation found in none of the languages of study in the scope of this project.

The result of the above experiment is similar to the result of the basic setup, with a few exceptions:

Sentence	Basic setup result	Experimental unigram result	Correct?
I hate AI.	EN	Best guess: FR	No
Est-ce que l'arbitre est la?	OT	Best guess: FR	Yes
Sell it for me.	EN	Best guess: FR	No
Jump for joy.	OT	Best guess: EN	Yes

From the above results, it can be deduced that the Bigram model is not enhanced when the occurrence of single letter is taken into consideration.

Another experiment has been conducted using French and Spanish to confirm the hypothesis of ease of confusion between languages from the same root, which in this case is latin.

Sentence	**Basic Bigram result**	**Correct?**
tu comprends	FR	YES
comprendes	FR	NO
pourriez vous repeter	FR	YES
puede repetir por favor	FR	NO
ca ne m'importe pas	FR	YES
no importa	OT	YES

The results show that when Spanish sentences contain words that are from the same latin root as the equivalent French words, the Bigram model rules in favor of French. This can be due to the limited content provided by the corpora, since the French corpora is much larger than the Spanish one, it creates a slightly more accurate Bigram model.

Conclusion

In conclusion we see that the bigram model is more accurate when trained on the same amount of data though both models were fairly accurate. The unigram is essentially just a bag of words model whereas the bigram model improves upon this by taking order of characters into account. If we had more data, more training corpus’ to add to the training set our models would be more accurate still.