# Ling 473 Project 5
## Due 11:59pm on Thursday, September 9, 2021

For this project you will build a naïve Bayesian classifier which is able to classify text fragments according to their language. Determining the language of a document may seem trivial, since the most common few words in each language—or the character set it uses—could be thought of as an identifying signature. But we would like to use a more principled approach that *quantitatively scores* the probability of some fragment of text being written in one language, relative to others.

The object of this task is to determine the language $\hat{L} \in \mathcal{L}$ ($\mathcal{L}$ is the set of all languages) that maximize $P(L|W)$, where $L \in \mathcal{L}$ is a language and $W = w_1 \cdots w_n$ (there are $n$ words in the text) is the given text (i.e., word sequence). Therefore, the objective function of this task is

$$\hat{L} = \arg\max_{L \in \mathcal{L}} P(L|W)$$

To compute $P(L|W)$, naturally, Bayes' theorem comes to mind:

$$P(L|W) = \frac{P(W|L) * P(L)}{P(W)}$$

As we did in the POS-tagging lecture, we can ignore the denominator of the right side. This is because we will be interested in comparing the result of this calculation for different languages—but always for the same text. Since we will only compare results for the same text fragment, $P(W)$ can be excluded from the *argmax* calculation.

$$\hat{L} = \arg\max_{L \in \mathcal{L}} P(W|L)\, P(L)$$

**Next, we will assume that each of the 15 languages in this project are equally likely**. Again, we could always multiply the right side by $\frac{1}{15}$, but since we will only use the results for comparison, this term can also be dropped.

$$\hat{L} = \arg\max_{L \in \mathcal{L}} P(W|L)$$

Finally, we apply the naïve Bayesian assumption: all words in the text are *conditionally independent* of each other. Therefore, the probability that a text fragment is written in a language will be product of the probabilities of all of its words, according to a unigram language model of that language.

$$P(W|L) = \prod_{i=1}^{n} P(w_i|L)$$

Because this product may approach zero, the calculation will actually be done by adding log-probabilities.

$$\lg P(W|L) = \sum_{i=1}^{n} \lg P(w_i|L)$$

**In this project, we use 10-base log for all log-probabilities** and 10-base log (i.e., $\log_{10}$) is denoted by lg in this specification**.**

Therefore, the best predicted language $\hat{L}$ should be

$$\hat{L} = \arg\max_{L \in \mathcal{L}} \sum_{i=1}^{n} \lg P(w_i|L)$$

Herein, $P(w_i|L)$ can be estimated through

$$P(w_i|L) = \frac{the\ count\ of\ w_i\ in\ language\ L}{the\ total\ number\ of\ words\ in\ languege\ L}$$

**To find out the best language $\hat{L}$, we need to compute the value of $\sum_{i=1}^{n} \lg P(w_i|L)$ for all 15 languages and predict the language with the highest value.**

## Input

In this project, your code will have two input. The first is the directory containing the 15 unigram language models and the second is the text file containing the text to be identified.

All files for this project use the **UTF-8** encoding. Because some languages use lexical capitalization (for example, nouns are always capitalized in German), and because some programming languages may lack support for correctly changing the case of letters with diacritics, this project will use **case-sensitive comparison**. ***Do not ignore case*.** For example, "*the*" and "*The*" are two different words in English.

The details of the language model files and the input text file are illustrated in the following content.

### Unigram Language Models

The directory **/dropbox/20-21/473/project5/language_models** contains 15 unigram language model files. The format of the file name follows *{xxx}.unigram-lm*, where *xxx* is the **three-letter language code**. For example, for English unigram language model, whose file name is *eng.unigram-lm*, its three-letter language code is *eng*. In the language models, each line is a (*word*, *count*) pair that is similar to what you built for Project 2. For example, the unigram language model for English looks like

| | |
|---|---|
| the | 914934 |
| of | 455662 |
| to | 417411 |
| and | 333525 |
| in | 267940 |
| that | 217468 |
| is | 214041 |
| a | 210407 |
| I | 141077 |
| for | 137073 |
| … | |

All unigram models are truncated, containing only the most common 1,500 general words plus a special "<UNK>" token representing the unknown/unseen words.

In this project, the count of "<UNK>" for each language is given to you. You can use it to estimate $P(w_i|L)$ when $w_i$ is not in the general 1,500 words in the given language $L$. That is, if $w_i$ is not in the unigram language model, estimate it by $P(< UNK >|L)$.

**Hint:** when you compute the total number of words in a language, the number of "<UNK>" should also be counted. This will make sure the sum of all word probabilities is equal to one. That is

$$\sum_{w \in V_L \cup \{'<UNK>'\}} P(w|L) = 1$$

where $V_L$ is the vocabulary of general words in the language $L$.

**input.txt**

**/dropbox/20-21/473/project5/input.txt** contains 24 examples of text to be identified. Each example consists of an identifier, a tab character, and the text. The identifier is the correct three-letter language code for the text (you can compare the model prediction with the correct language to see whether the model prediction is correct). For example, in the following line, "*eng*" is the language code for English and "I voted in favour of the report The …" is the text.

eng I voted in favour of the report The …

In this project, the punctuations in the input text have already been removed for you. So, you can just **split the text into words by white spaces** and compute the value of $\sum_{i=1}^{n} \lg P(w_i|L)$ for each language $L$, where $P(w_i|L)$ should be estimated from the language model.

**Output Format**

Write your results to the output file (not to the standard output) with UTF-8 encoding (this is similar to project 3). For each example you process, write the following lines:

- The identifier (i.e., the correct language code), a tab, and the text
- One line per language model, consisting of the three-letter language code, a tab character and the value of $\sum_{i=1}^{n} \lg P(w_i|L)$. Herein, you don't need to do anything special for the value (e.g., round it to a particular number); just write the value as it is. Also, the order of different language models does not matter.
- The word "result", a tab character, and the language three-letter code for the "best" result (i.e., the language with the highest value) from your classifier.
- An empty line which serves as the breaker of different examples.

The format of the output file should look like the following (the order of the language-value pair does not matter):

dan Jeg håber at medlemmerne vil gøre de kolleger som måske går rundt udenfor og leder efter den opmærksom på dette
dan -53.980461
deu -76.945737
dut -78.708899
eng -79.362169

fin -76.724482
fra -77.948135
gla -73.641607
ita -78.338126
nob -60.749495
pol -73.780847
por -77.385501
spa -77.991172
swe -68.614655
swh -75.589044
tgl -77.349553
result dan

deu Herr Präsident das Grünbuch der Kommission und der Bericht Martens enthalten viele wichtige Reflexionen und Analysen
dan -60.327345
deu -43.572167
dut -65.405682
eng -64.878867
fin -61.419921
fra -64.660841
gla -61.316054
ita -63.600907
nob -59.426191
pol -59.024677
por -62.309188
spa -64.778018
swe -63.523012
swh -60.471235
tgl -64.858473
result deu

## Running the code on Patas

The program should accept exactly **three argument**s where

- the **first** is the absolute path to the directory that contains the unigram language models (i.e., /dropbox/20-21/473/project5/language_models) that contains the categories and the members;
- the **second** is the absolute path to the **input** file (i.e., /dropbox/20-21/473/project5/input.txt);
- the **third** is the path to the output file. Note that, in this project, **your program needs to write the results into a file**, rather than printing it to the standard output. This is similar to project 3.

Run your code on Patas with the following command line:

**./run.sh /dropbox/20-21/473/project5/language_models /dropbox/20-21/473/project5/input.txt ./output**

The results for the first two examples in the output file should be the same as the above figure. You can download the output file and open it on your local machine. If you open it on Patas, you may not be able to see the correct content due to the encoding issues.

## Submission

### hw.tar.gz

Include the following files in your submission:

| | |
|---|---|
| `run.sh` | The command(s) that run your program. Your code should take **three arguments**. The first is the directory to the language models file (i.e., /dropbox/20-21/473/project5/language_models); the second is the input text file (i.e., /dropbox/20-21/473/project5/input.txt); the third is the output file with your results following the formats illustrated above. |
| `output` | The output file which is produced by **./run.sh /dropbox/20-21/473/project5/language_models /dropbox/20-21/473/project5/input.txt ./output** |
| `(source code and binary files)` | All source code and binary files (jar, a.out, etc., if any) required to run and compile your program |

Similar to previous projects, gather together all the required files and use the following command to package your files for submission.

```
tar -czf hw.tar.gz ./*
```

Notice that this command packages all files in the current directory; do not include any top-level directories.

Check whether your submission contains all required file using

```
/dropbox/20-21/473/project5/check_project5.sh
```

**`readme.{pdf, txt}`**

Your write-up of the project. Describe your approach, any problems or special features, or anything else you'd like me to review. If you could not complete some or all of the project's goals, please explain what you were able to complete.

Submit **hw.tar.gz** and **readme.{pdf, txt}** to Canvas.

**Citation**

Philipp Koehn. *Europarl: A Parallel Corpus for Statistical Machine Translation*. MT Summit 2005. http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/europarl-mtsummit05.pdf