Алина Шаймарданова, БКЛ-152

Для выполнения этого задания, я решила не останавливаться на четырёх языках, которые были изначально предложены в тексте, а расширить список до 10 языков. Я это сделала, для того, чтобы более объективно оценить эффективность приведенных алгоритмов по распознаванию языка, т.к. в случае с 4 языками ('be', 'kk', 'uk', 'fr') будет очень лего опередить французский, т.к. он единственный не содержит кириллических знаков, и белорусский, казахский и украинский отличить друг от друга достаточно легко, т.к. они содержат в себе уникальные друг для друга буквы.
Таким образом, я взяла следующие языки и выкачала для каждого из них по 100 статей: 'bg', 'be', 'kk', 'uk', 'fr', 'ru', 'en', 'mk', 'la', 'de'

**Первый метод: частотные слова**

В начале, был сформирован частотный словарь вида:
{‘ru’:
    {‘слово’: его частотность для текстов этого языка, …}
…}

То есть ключ – это язык, значение – это словарь, где ключ – это слово, а значение – это частотность этого слова в текстах этого языка (sum( freq_dict[lang].values())).

Например, первые 4 значения для казахского:

```
0.24532825706742278     жалғастырғышты
0.12463899494783258     жүріс
0.11222148334254944     тежегішті
0.10194716438342984     және
```

После этого, опираясь на этот словарь, довольно распространённым методом мы определяли язык текста. Получаемый на вход текст мы токенизируем, очищая от мусора и цифр. После этого, мы для каждого языка проверяем, наличие каждого слова из анализируемого текста в языковом-частотном словаре. Если слово встречается в определенном языке, то мы приплюсовываем частотность этого слова в текстах этого языка (из словаря) в отдельный словарь, где ключ – это язык, а значение – это сумма частот слов текста. После этого, мы смотрим, у какого языка получилась наибольшая сумма и выбираем его.

**Второй метод: частотные символьные n-граммы**

В начале, мы разделяем весь текст на n-граммы, где n=3.
После этого, аналогично первому методу, мы формируем частотный словарь, где ключ – это язык текстов, а значение – это частотность трёх символов в текстах этого языка.

Первые 4 значения для казахского:

```
0.12213749127777604     жиі
0.10104254400129953     жег
0.09255603943632926     теж
0.08533199900519951     үйл
```

После этого, так же как и в первом методе определяем принадлежность текста к тому или иному языку.

Сравним результаты:

С частотным словарём:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| be | 1.00 | 0.91 | 0.95 | 95 |
| bg | 0.69 | 0.98 | 0.81 | 92 |
| de | 0.99 | 0.98 | 0.98 | 93 |
| en | 0.91 | 0.99 | 0.95 | 96 |
| fr | 0.95 | 0.99 | 0.97 | 96 |
| kk | 0.99 | 1.00 | 1.00 | 100 |
| la | 1.00 | 0.85 | 0.92 | 97 |
| mk | 0.96 | 0.95 | 0.95 | 97 |
| ru | 0.83 | 0.71 | 0.76 | 89 |
| uk | 0.99 | 0.85 | 0.92 | 96 |
| avg / total | 0.93 | 0.92 | 0.92 | 951 |

С n-граммами:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| be | 1.00 | 0.79 | 0.88 | 95 |
| bg | 0.65 | 0.33 | 0.43 | 92 |
| de | 0.95 | 0.99 | 0.97 | 93 |
| en | 0.97 | 0.97 | 0.97 | 96 |
| fr | 0.98 | 1.00 | 0.99 | 96 |
| kk | 0.95 | 0.99 | 0.97 | 100 |
| la | 0.97 | 0.97 | 0.97 | 97 |
| mk | 0.41 | 0.99 | 0.58 | 97 |
| ru | 0.96 | 0.28 | 0.43 | 89 |
| uk | 0.82 | 0.66 | 0.73 | 96 |
| avg / total | 0.87 | 0.80 | 0.80 | 951 |

Дополнительно посмотрим на матрицу ошибок:

С частотным словарём:

```
[[ 86   3   0   0   0   0   0   2   3   1]
 [  0  90   0   1   0   0   0   1   0   0]
 [  0   0  91   0   0   0   0   0   2   0]
 [  0   0   0  95   0   1   0   0   0   0]
 [  0   0   0   0  95   0   0   0   1   0]
 [  0   0   0   0   0 100   0   0   0   0]
 [  0   0   1   6   5   0  82   0   3   0]
 [  0   4   0   0   0   0   0  92   1   0]
 [  0  26   0   0   0   0   0   0  63   0]
 [  0   8   0   2   0   0   0   1   3  82]]
```

Алина Шаймарданова, БКЛ-152

С n-граммами:

```
[[75  1  1  0  0  1  0  5  0 12]
 [ 0 30  0  1  0  0  0 61  0  0]
 [ 0  0 92  0  0  0  1  0  0  0]
 [ 0  0  0 93  2  0  0  1  0  0]
 [ 0  0  0  0 96  0  0  0  0  0]
 [ 0  1  0  0  0 99  0  0  0  0]
 [ 0  0  3  0  0  0 94  0  0  0]
 [ 0  0  0  0  0  0  0 96  1  0]
 [ 0  7  0  0  0  2  2 51 25  2]
 [ 0  7  1  2  0  2  0 21  0 63]]
```

Как видно из результатов алгоритм с частотным словарём показывает лучший результат, нежели с буквенными n-граммами.