Московский авиационный институт
(Национальный исследовательский университет)

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

Лабораторная работа № 0

по курсу «Искусственный интеллект»

Студент: Гаврилов М.С.

Группа: 80-306б

Преподаватель:

Оценка:

Москва, 2022

1. Постановка задачи

В данной лабораторной работе, вы выступаете в роли предприимчивого начинающего стартапера в области машинного обучения. Вы заинтересовались этим направлением и хотите предложить миру что-то новое и при этом неплохо заработать. **От вас требуется определить задачу, которую вы хотите решить, и найти под нее соответствующие данные.** Так как вы не очень богаты, вам предстоит руками проанализировать данные, визуализировать зависимости, построить новые признаки и сказать хватит ли вам этих данных, и, если не хватит, найти еще. Вы готовитесь представить отчет ваши партнерам и спонсорам, от которых зависит дальнейшая ваша судьба. Поэтому тщательно работайте:) И главное, день промедления и вас опередит ваш конкурент, да и сплагиаченная работа отразится на репутации

1. Постановка своей задачи

Я хочу решить задачу классификации частей речи в русском языке. Классические словари могут не содержать нужного слова, а также, поиск по таблицам словаря может занимать продолжительное время. Если же обучить классификатор на заранее размеченных данных, можно обойти эти две проблемы. Так как в русском языке часть речи задается морфемами, т.е. разные части речи имеют различные характерные особенности в структуре слова, я считаю, что задачу определения части речи можно решить с достаточно высокой точностью, даже без учета контекста, в котором слово употреблено. Такой классификатор может найти применение в решении различных задач обработки естественного языка (NLP).

Ввиду того, что различные характерные для частей речи морфемы не так многочисленны и разнообразны, в принципе, можно создать такой классификатор с помощью классического программирования. Так что, одна из основных моих задач – сравнить точность классификатора, основанного на методах машинного обучения и вручную запрограммированного классификатора. По возможности – добиться более высокой точности классификатора – нейросети, чем у классификатора – программы.

Перед подачей на вход классификатора, слово представляется массивом символов. Размер массива равен размеру самого длинного слова обучающей выборки. Если слово короче массива, в пустые ячейки записываются символы, означающие отсутствие буквы. Какие – не принципиально. Каждая ячейка массива является индивидуальным признаком слова. Важно – удалить из обучающей выборки слова, сильно длиннее большего числа сильно длинных слов (чтобы не было такого, чтобы в последней ячейке массива буква была только у 1-2 слов). Массив подается на вход анализатору. Результатом работы анализатора будет являться число – номер класса. Классы: имя существительное [1], имя прилагательное [2], имя

числительное [3], местоимение [4], глагол [5], наречие [6], причастие [7], предлог[8], союз [9], частица [10], междометие [11].

Я планирую использовать либо датасет nerus(https://github.com/natasha/nerus), либо анализатор RuWord2Tags (https://github.com/Koziev/ruword2tags), который размечает слова на основе данных русского грамматического словаря. Правда, данная фраза из readme последнего проекта: «Кроме того, для многих новых (out-of-vocabulary) слов пакет обеспечивает распознавание, даже если точно такого слова нет в словаре.» – заставляет меня подозревать, что в нем уже имплементировано нечто, подобное тому, что я планирую разработать. Не знаю, насколько это плохо.

2. Анализ данных

В итоге было решено использовать датасет nerus. Этот массив данных состоит из 2 гб размеченных новостных заметок. При проведении анализа классифицируемых данных я смотрел первые 1000 документов датасета.

Пример размеченного текста в датасете nerus:

```
tokens=[NerusToken(
     id='1',
     text='Вице-премьер',
     pos='NOUN',
     feats={'Animacy': 'Anim',
      'Case': 'Nom',
      'Gender': 'Masc',
      'Number': 'Sing'},
     head_id='7',
     rel='nsubj',
     tag='O'
 ),
 NerusToken(
     id='2',
     text='по',
     pos='ADP',
     feats={},
     head_id='4',
     rel='case',
```

Для начала я попробовал представить данные так, как задумал изначально – «Перед подачей на вход классификатора, слово представляется массивом символов. Размер массива равен размеру самого длинного слова обучающей выборки. Если слово короче массива, в пустые ячейки записываются символы, означающие отсутствие буквы. Какие – не принципиально. Каждая ячейка массива является индивидуальным признаком слова.»

Из всей разметки я выделил части речи, а слова поместил в массивы размером с длину наиболее длинного слова.

```
['п', 'о', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
                :       ['ADP']
['с', 'о', 'ц', 'и', 'а', 'л', 'ь', 'н', 'ы', 'м', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
                :       ['ADJ']
['в', 'о', 'п', 'р', 'о', 'с', 'а', 'м', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
                :       ['NOUN']
['Т', 'а', 'т', 'ь', 'я', 'н', 'а', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']
                :       ['PROPN']
```

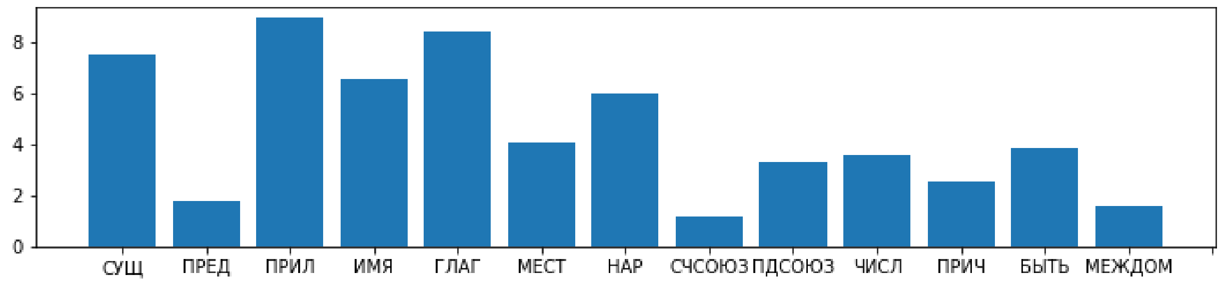
Несмотря на то, что эти идея изначально показалась мне неплохой, я вскоре понял, что этот вариант представления данных не очень хорош. Он кое-как пригоден для классификации нейросетью, и то, желательно, сверточной, но не алгоритмом машинного обучения. Как минимум, здесь нет никакой возможности анализировать последнюю букву слова: в зависимости от длины она находится в разных ячейках.

Так что я решил, что лучше будет выделить ограниченное количество признаков.

В качестве предполагаемых признаков рассматривались:

- Длина слова
- Значения букв на определенных позициях (первая, последняя, и т.д)
- Наличие тех или иных букв в слове (может быть так, что в словах одной части речи часто встречается определенная буква, а в словах другой она редка).

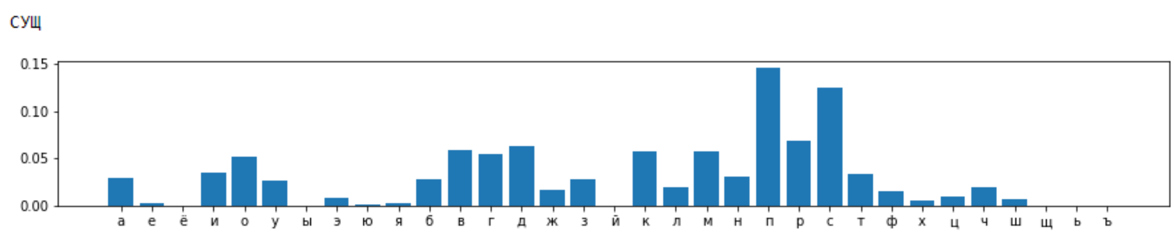
Сравнение средних длин слова по классам



Видно, что длина слов существенно отличается между некоторыми классами. Данный признак пригоден для классификации, но, определенно, его одного недостаточно.

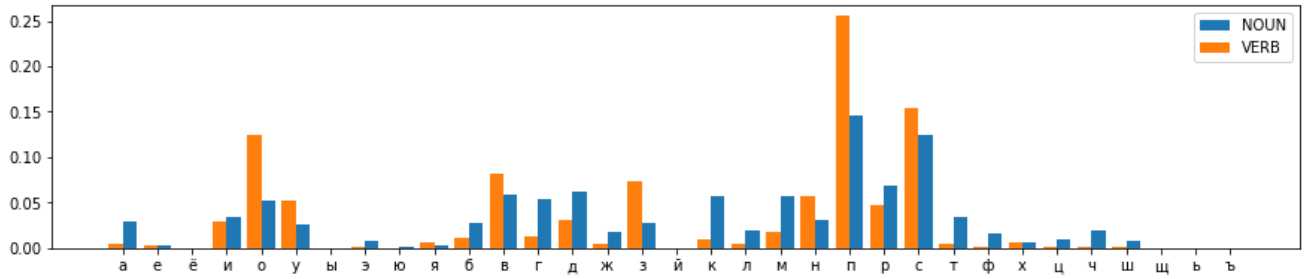
Сравнение первых букв слова по классам

Для каждой части речи я составил такую диаграмму:

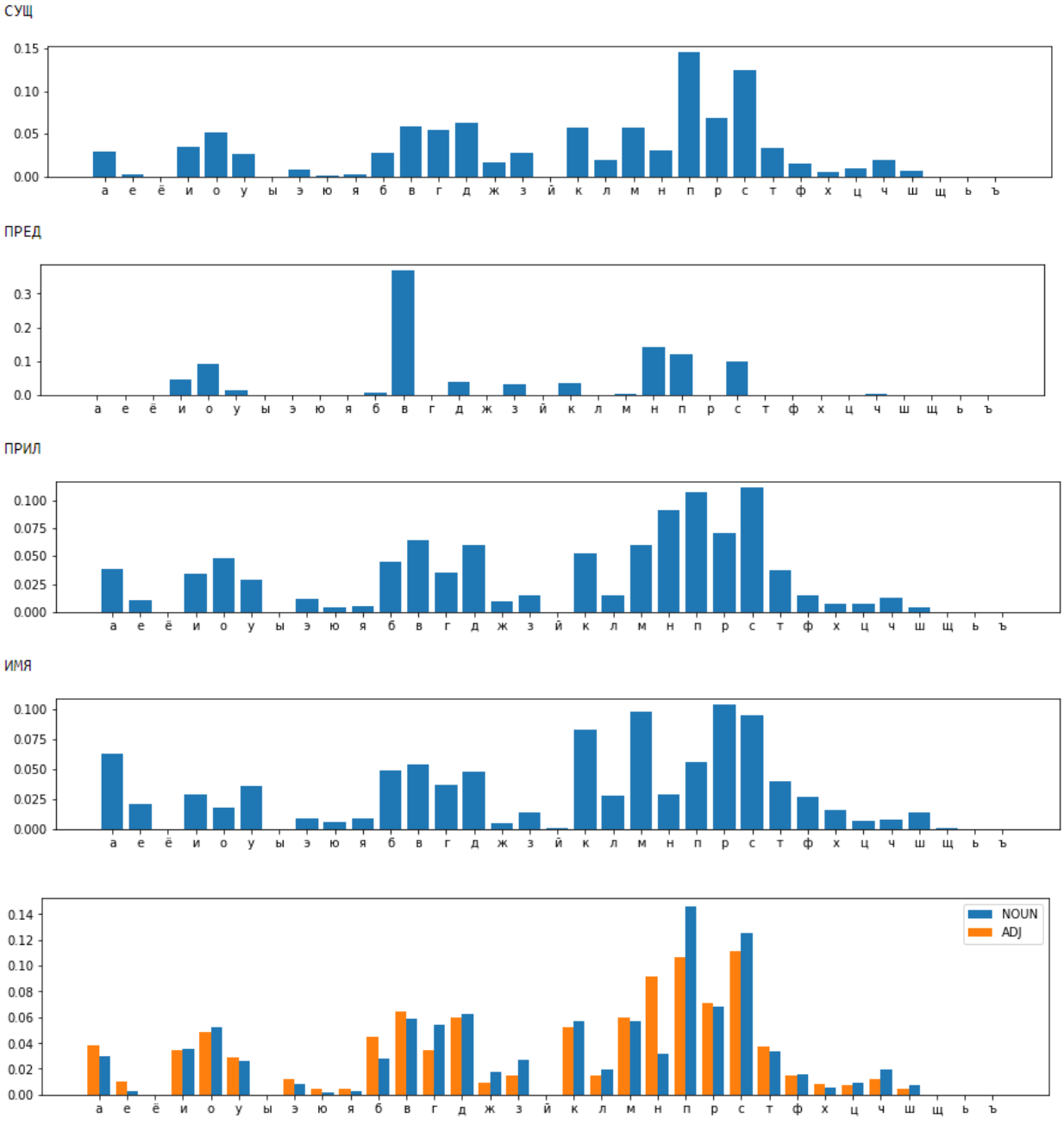


Сопоставляя две диаграммы для разных классов, можно понять, какие буквы на первой позиции для какого из них более характерны

```
overlay(ClassConvToEN("СУЩ"),ClassConvToEN("ГЛАГ"), ClassFreqFST)
```

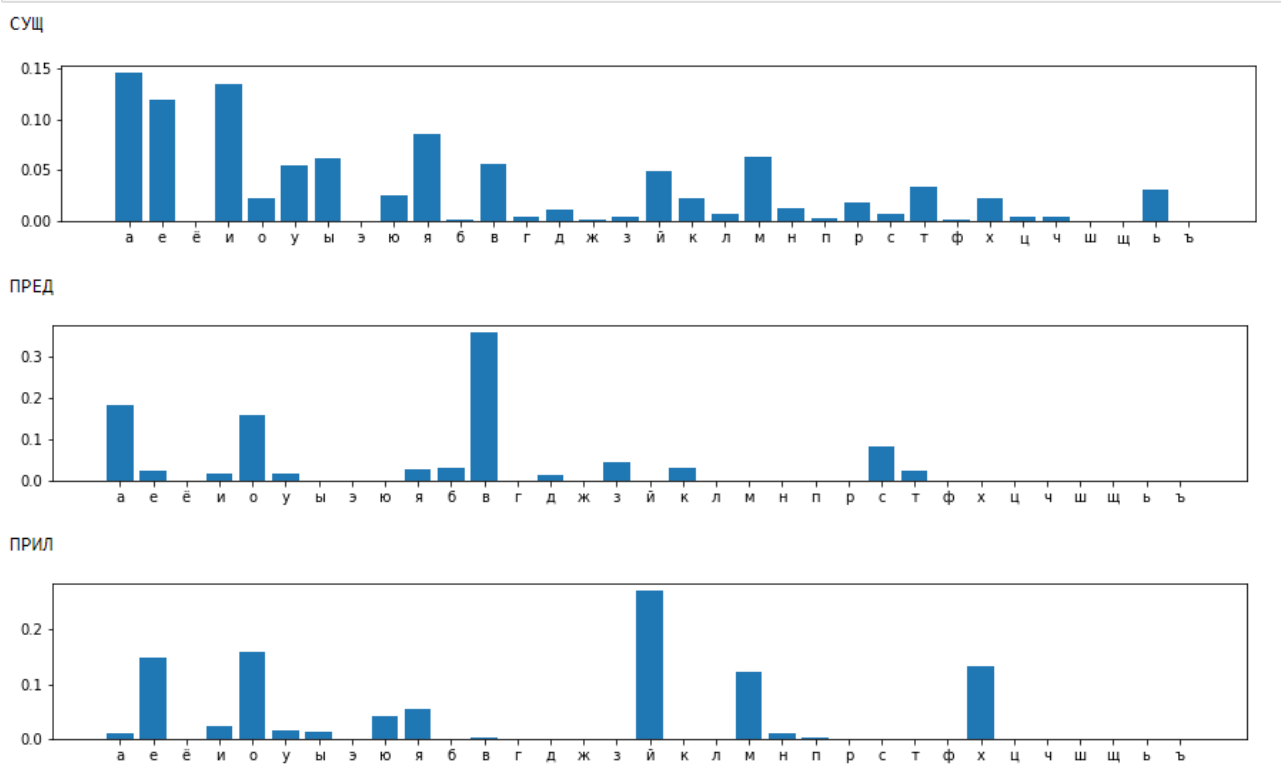


Всего получилось 14 диаграмм, по которым видно, что, хоть для некоторых классов первые буквы различаются существенно, для иных они почти одинаковы (пример – существительное и прилагательное на картинке ниже)

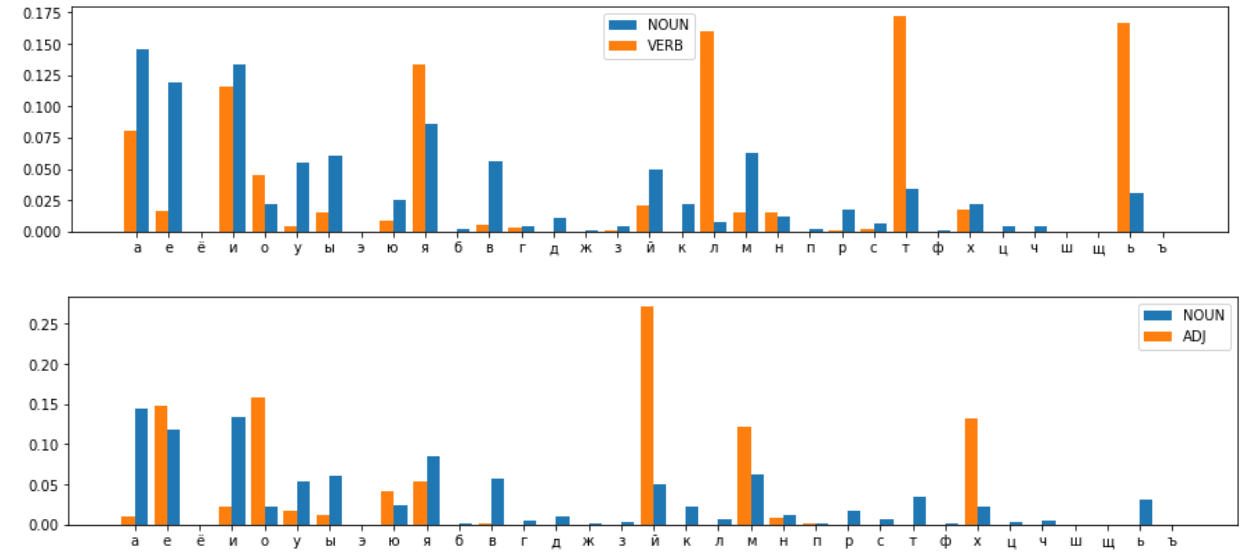


Сравнение последних букв слова по классам

Диаграммы по последним буквам:



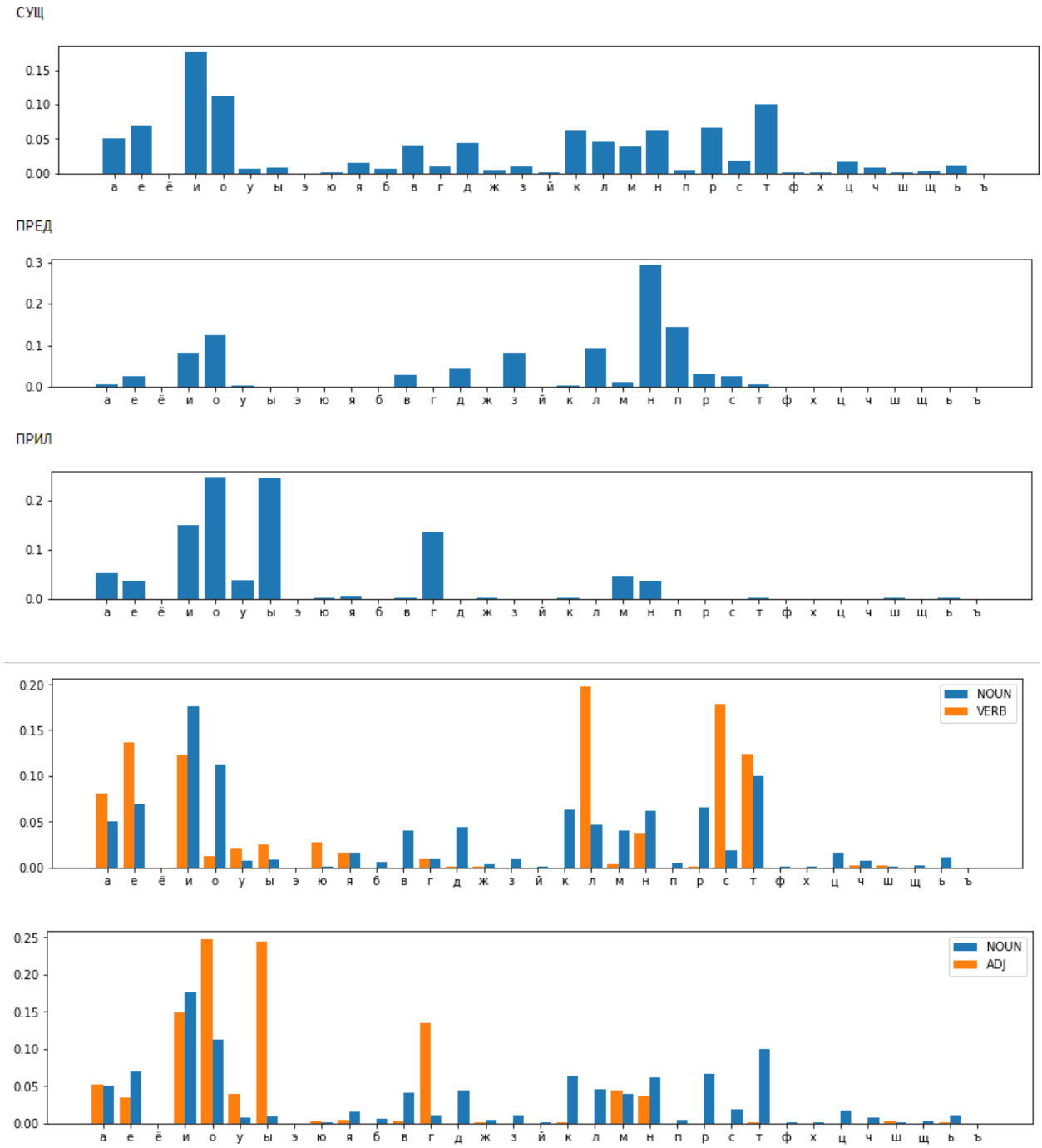
Видно, что последние буквы, что неудивительно, на самом деле, отличаются для разных частей речи куда сильнее



Последняя буква слова куда больше подходит на роль признака для классификации, чем первая, но отказываться от первой я не буду, думаю, она тоже может привнести точности.

Сравнение предпоследних букв слова

Часть диаграмм:



Как видно, предпоследняя буква также существенно отличатся в зависимости от части речи.

Пока что я ограничусь первой, последней и предпоследней буквой в качестве «буквенных» признаков, однако, если точности будет не хватать, можно еще больше углубить анализ, беря вторую, третью, и третью с конца букву. Согласно моему предположению, пока эти выбираемые буквы будут приходиться на приставки, окончания и суффиксы, они должны продолжать прибавлять точность, но, когда большая их часть станет принадлежать основам, смысла их учитывать больше не будет.

Сравнение частоты встречаемости в словах различных букв

Здесь проводился подсчет числа букв в словах каждой части речи, а затем результат делился на сумму длин всех пройденных слов этой части речи.

Часть диаграмм:

Видно, что различия между частотами в разных частях речи хоть и есть, но не особо велики. Данный признак я использовать не буду.

3. Вывод

О других возможных признаках

Пока что я попробую построить классификатор на следующих трех признаках:

- Длина слова
- Первая буква
- Последняя буква
- Предпоследняя буква

Если точность будет неудовлетворительной, попробую ввести такие признаки, как:

- Вторая буква
- Третья с конца буква

Также я рассматриваю возможность рассматривать в качестве признаков морфемы слов, однако разбитое на морфемы слово вроде бы неплохо классифицируется и без искусственного интеллекта.

О дискретности признаков

Разница между буквой «В» на позиции n и буквой «Г» на той же позиции – совсем не то же, что и яркость пикселя x = 210 или 220. Не знаю, насколько серьезной проблемой это будет. Как минимум я сгруппировал гласные и согласные буквы в алфавите, это хоть немного должно улучшить ситуацию. Но я боюсь, что эффект, похожий на тот, что возникает при попытке классифицировать данные аппроксимирующей нейросетью, все же будет проявляться, вопрос только, насколько сильно.

О необходимости такого классификатора

Я планирую оценивать полезность анализатора, сравнивая его работу с классификатором, который определяет часть речи программно, с помощью анализа морфем.

Если точность моего классификатора будет не сильно ниже (на 2-5%), то уже неплохо, если разница будет еще меньше, то, возможно при небольших доработках от него можно будет добиться точности не хуже, чем у программных классификаторов, а если же мой классификатор окажется точнее, то его можно будет считать применимым в качестве метода определения части речи слова.