Московский государственный университет имени М. В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Рысьмятова Анастасия Александровна

Ранжирование текстов литературных произведений

КУРСОВАЯ РАБОТА

Научный руководитель:
д.ф.-м.н., профессор
А.Г. Дьяконов

Содержание

Аннотация

В данной работе методами машинного обучения решается задача ранжирования текстов литературных произведений в порядке их написания. Обучающая и тестовая выборки сформированы из текстов стихотворений русских поэтов XVIII-XX века.

Исследованы основные подходы к решению задачи ранжирования. Для экспериментов был выбран попарный подход, сводящий задачу ранжирования к бинарной классификации.

В работе приведены результаты классификации текстов литературных произведений как нейросетевыми методами машинного обучения, получившими особую популярность в последнее время, так и традиционными методами автоматической обработки текстов.

1 Введение

Автоматическая обработка текстов становится все более востребована в связи с постоянно растущим объемом информации в Интернете и потребностью в ней ориентироваться. Ранжирование текстов — важная задача автоматической обработки текстов, ее исследованием активно занимаются все поисковые системы. Наиболее популярные поисковые системы используют методы машинного обучения для решения данной задачи. [12]

В работе исследованы основные подходы к ранжированию текстов на примере задачи ранжирования литературных произведений. Для этого с сайта [11] выбраны тексты стихотворений различных русских поэтов, и с помощью алгоритмов машинного обучения построен алгоритм, способный ранжировать тексты одного автора в порядке возраста, в котором он написал произведения.

Для решения данной задачи ранжирования был применен попарный подход. Был построен бинарный классификатор, принимающий на вход пары стихотворений одного и того же автора, и определяющий, какое стихотворение было написано раньше. Качество классификации измерялось по метрике Accuracy.

В работе приведены результаты решения задачи ранжирования литературных произведений с использованием как нейросетевых методов машинного обучения [9] [7], получивших особую популярность в последнее время, так и традиционных методов автоматической обработки текстов.

В работе показано, что с точностью 70% можно определить возраст автора, в котором было написано стихотворение, если известна информация о других произведениях данного автора.

1.1 Постановка задачи ранжирования

Ранжирование решает следующую задачу. Имеется множество объектов, для конечного подмножества которых известен их правильный порядок. Это подмножество называется обучающей выборкой. Порядок остальных объектов не известен. Требуется построить алгоритм, способный упорядочивать произвольные объекты из исходного множества.

Задачу ранжирования можно формализовать следующим образом [13]:
$X-$ множество объектов; $X^\ell = \{x_1, ..., x_\ell\}-$ обучающая выборка;
$i \prec j-$ правильный порядок на парах $(i, j) \in \{1, ..., \ell\}^2$;
необходимо построить ранжирующую функцию $a : X \to \mathbb{R}$ такую, что $i \prec j \Rightarrow a(x_i) < a(x_j)$.

В задаче ранжирования текстов объекты — это текстовые документы.

2 Методы ранжировния

Выделяют три основных подхода в задаче ранжирования [5]:

Поточечный подход (pointwise approach). В поточечном подходе предполагается, что каждому объекту поставлена в соответствие численная оценка. Задача обучения ранжированию сводится к построению регрессии: для каждого отдельного объекта необходимо предсказать его оценку. В рамках этого подхода могут применяться многие алгоритмы машинного обучения для задач регрессии. Когда оценки могут принимать лишь несколько значений, также могут использоваться алгоритмы классификации.

Попарный подход (pairwise approach). В данном подходе обучение ранжированию сводится к построению бинарного классификатора, которому на вход поступают два объекта, соответствующих одному и тому же запросу, и требуется определить, какой из них должен стоять выше в упорядоченном списке.

Списочный подход (listwise approach). Списочный подход заключается в построении модели, на вход которой поступают сразу все объекты, а на выходе получается их перестановка.

В данной работе задача ранжирования сводится к задаче классификации текстов с помощью попарного подхода.

3 Традиционные методы решения задачи классификации текстов

Задачу классификации текстов решают с помощью следующих этапов :

1. Предварительная обработка текстов.

   Все тексты на естественном языке имеют большое количество слов, которые не несут информации о данном тексте. К примеру, в английском языке такими словами являются артикли, в русском к ним можно отнести предлоги, союзы, частицы. Данные слова называют шумовыми или стоп-словами. Для достижения лучшего качества классификации текстов обычно удаляют такие слова. По мимо этого на данном этапе приводят каждое слово из текста к основе, одинаковой для всех его грамматических форм. Это необходимо, так как слова несущие один и тот же смысл могут быть записаны в разной форме. Например, одно и то же слово может встретиться в разных склонениях, иметь различные приставки и окончания.

2. Перевод текстов в вещественное пространство признаков, где каждому документу сопоставляется вектор фиксированной длины.

   Наиболее известные способы, позволяющие осуществить перевод текста в пространство признаков, основаны на статистической информации о словах. При использовании этих способов каждый объект переводится в вектор, длина которого равна количеству используемых слов во всех текстах выборки. В данной работе для перевода документов в вещественное пространство признаков используется TF-IDF.

   Определение 3.1. *TF-IDF [6] — это статистическая мера, используемая для оценки важности слова в контексте документа. Вычисляется по формуле:*

$$\text{TF-IDF}(w, d, D) = \text{TF}(w, d) \times \text{IDF}(w, D)$$

   TF — частота слова, оценивает важность слова w_i в пределах отдельного документа.

$$\text{TF}(w, d) = \frac{n_i}{\sum_k n_k}$$

   n_i — число вхождений слова i в документ.

   $\sum_k n_k$ — общее число слов в данном документе.

IDF — обратная частота документа. Учёт IDF уменьшает вес широко употребляемых слов.

$$\text{IDF}(w, D) = \log \frac{|D|}{|(d_i \supset w_i)|}, \text{ где}$$

$| D |$ — количество документов в корпусе.

$| (d_i \supset w_i) |$ — количество документов, в которых встречается слово w_i.

3. Выбор алгоритма классификации

   Как правило, полученное признаковое пространство в данном методе сильно разрежено и имеет высокую размерность за счет того, что различных слов, встречающихся во всей выборке, обычно много. Из-за этого для данной задачи чаще всего используют линейные методы машинного обучения.

4 Нейросетевые методы решения задачи классификации текстов

В связи с успехом применения сверточных нейронных сетей к задаче классификации изображений появилось множество попыток использовать нейронные сети в других задачах. В последнее время их стали активно использовать для задачи классификации текстов.

4.1 Word2Vec для классификации текстов

Word2Vec [2] — нейросетевая технология от компании Google, которая разработана специально для статистической обработки больших массивов текстовой информации. Word2Vec собирает статистику о появлении слов в данных, удаляет наиболее редко встречаемые и часто встречаемые слова, после чего методами нейронных сетей решает задачу снижения размерности и выдает на выходе компактные векторные представления слов заранее определенной длины. При этом Word2Vec максимизирует косинусную меру близости между векторами слов, которые встречаются в близких контекстах и минимизирует косинусную меру между словами которые не встречаются рядом.

В статье [3] представлена функция расстояния между тестовыми документами на основе векторных представлений слов, полученных с помощью Word2Vec. Расстояние вычисляется как минимальная дистанция, на которую слова одного документа должны «переместиться», чтобы достичь соответствующих слов другого документа.

Расстояние между текстами можно вычислять как расстояние между векторами, полученными с помощью обычного представления мешка слов (BOW), но тогда фразы, которые означают одно и то же могут иметь большое расстояние. Рассмотрим пример:

"Trump speaks to the media in Illinois"

"The President greets the press in Chicago"

Две эти фразы имеют один и тот же смысл, но не имеют общих слов, поэтому расстояние между BOW векторами будет большим. Авторы статьи используют нейросетевую технологию word2vec для получения векторного представления слов, учитывающее семантическое сходство между отдельными словами, и предлагают новую формулу вычисления расстояния между текстами, на основе которой можно классифицировать тексты. Опишем подробно, как вычисляется расстояние между текстами в статье [3].

Пусть $X \in \mathbb{R}^{n \times m}$ — матрица векторных представлений слов. $m-$ размерность вектора, $n-$ число слов в документах. Пусть $d \in \mathbb{R}^n$ - вектор документа, полученный с помощью представления BOW. Пусть $c(i,j) = \|x_i - x_j\|_2$ - стоимость «путешествия» одного слова к другому. Пусть имеется два текста, которые имеют векторное представление d и d', для них определим разреженную матрицу $T \in \mathbb{R}^{n \times n}$ такую, что $T_{ij} \geqslant 0$ означает «сколько» слов i в d перемещается в слово j в d' При этом гарантируется, что $\sum_j T_{ij} = d_i$ и $\sum_i T_{ij} = d'_j$.

Расстояние между текстами определяется как минимальная (взвешенная) совокупная стоимость, необходимая для перемещения слов из d в d'. Для нахождения расстояния между текстами необходимо решить следующую оптимизационную задачу.

$$\min_{T \geqslant 0} \sum_{i,j=1}^{n} T_{ij} c(i,j)$$
$$\text{subject to: } \sum_{j=1}^{n} T_{ij} = d_i,\ \forall i \in \{1,2,..,n\}$$
$$\sum_{i=1}^{n} T_{ij} = d_j,\ \forall j \in \{1,2,..,n\}$$

В статье [3] было показано, что, используя построенную функцию расстояния при решении задачи классификации текстов метрическим алгоритмом, можно достичь лучшего качества, чем при использовании других функций расстояния.

4.2 LSTM сеть

Рекуррентные нейронные сети [7] — вид нейронных сетей, в которых имеется обратная связь. Это значит, что нейроны элементов последующих слоев имеют соединения с нейронами предшествующих слоев. Такая архитектура приводит к возможности учета результатов преобразования нейронной сетью информации на предыдущем этапе для обработки входного вектора на следующем этапе функционирования сети. Рекуррентные нейронные сети активно применяются для решения задачах автоматической обработки текстов, таких как моделирование языка, распознавание речи, перевод. На Рис.1 представлена схема работы рекуррентных нейронных сетей. Разные типы рекуррентных нейронных сетей отличаются архитектурой ячейки A.

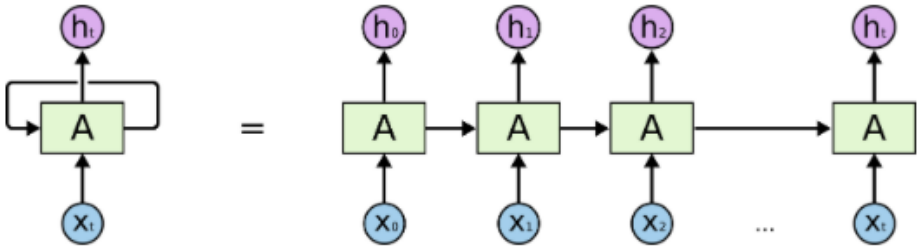

Рис. 1: Рекуррентная нейронная сеть

LSTM [7] (long short-term memory, долговременно-кратковременная память) сеть — особый тип рекуррентных нейронных сетей. На Рис.2 представлена архитектура ячейки A в LSTM сети.

Опишем работу LSTM сети. Введем обозначения:

$\sigma(s) = \dfrac{1}{1+e^{-s}}$ — сигмоидная функция активации

$\tanh(s) = \dfrac{e^{s}-e^{-s}}{e^{s}+e^{-s}}$ — гиперболический тангенс

x_t — входной вектор в текущую ячейку LSTM сети

h_{t-1} — выход предыдущей ячейки LSTM сети

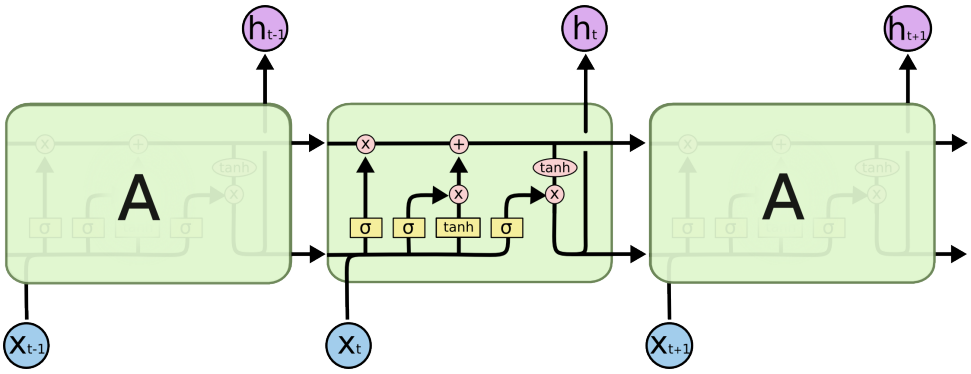

Рис. 2: Рекуррентная нейронная сеть

$W_f, W_C, W_i, W_o, b_f, b_C, b_i, b_o$ — матрицы и векторы, параметры нейронной сети

Оператор [,] — конкатенация векторов

Тогда h_t — выход текущей ячейки LSTM сети вычисляется следующим образом

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\widehat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widehat{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

4.3 Сверточная нейронная сеть

В данной работе используется сверточная нейронная сеть с посимвольным подходом. В статье [9] описана архитектура нейронной сети, а также показано, что данная нейронная сеть справляется с задачей классификации текстов, лучше чем рекуррентные нейронные сети. Опишем метод классификации текстов из статьи [9] подробнее.

Назовем алфавитом упорядоченный набор символов. Пусть выбранный алфавит состоит из m символов. Каждый символ алфавита в тексте закодирован с помощью $1 - m$ − кодировки. (т. е. каждому символу сопоставлен вектор длины m, элемент которого равен единице в позиции равной порядковому номеру символа в алфавите, а нулю во всех остальных позициях.) Если в тексте встретится символ, который не

вошел в алфавит, то необходимо закодировать его вектором длины m, состоящим из одних нулей.

Из текста выбираются первые ℓ символов. Параметр ℓ должен быть большим, чтобы в первых ℓ символах содержалось достаточно информации для определения класса всего текста.

Каждому из выбранных ℓ символов текста сопоставляется вектор длины m.

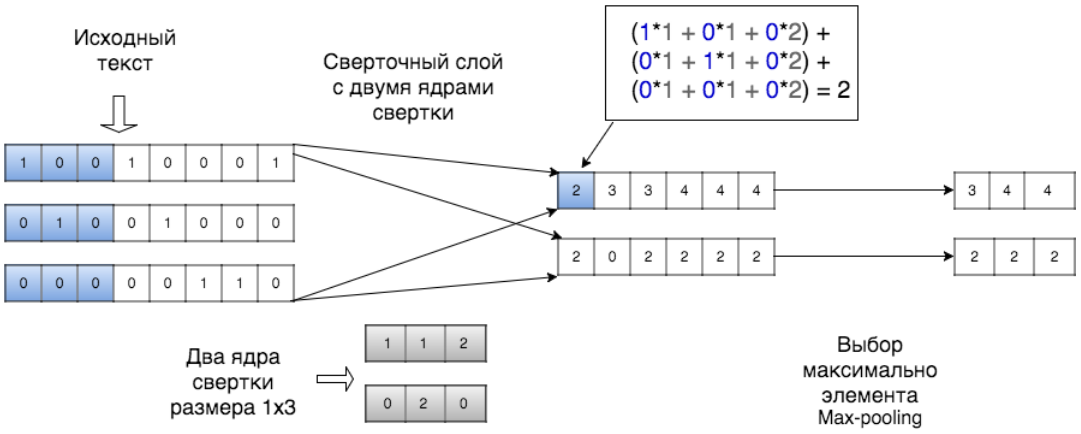

Рис. 3: Посимвольный подход

Далее полученные векторы составляются в матрицу размера $m \times \ell$, в которой каждый столбец будет иметь не более одной единицы. Каждая строка полученной матрицы используется как отдельный канал. Ядро свертки применяется к каждому каналу в отдельности, а полученные результаты на выходе для каждого канала суммируются между собой. На вход сверточной нейронной сети подается m векторов размера $1 \times \ell$ аналогично изображениям. Архитектуру сети необходимо выбирать, исходя из задачи. На Рис. 1 приведен пример посимвольного подхода для $\ell = 8$, $m = 3$. В примере показан один сверточный и один субдискретизирующий (max-pooling) слой с выбором максимального элемента и разбиением на ячейки размера 1×2 .

В статье [9] были приведены эксперименты, которые показали, что описанный подход с высокой точностью классифицирует тексты, по сравнению с большинством других известных на данный момент методов классификации текстов, если размеры выборки достаточно велики. На выборке размером 1400000 объектов сверточная нейронная сеть с посимвольным подходом дала качество классификации по метрике

accuracy — 0.712, а традиционным методом классификации текстов удалось достичь лишь — 0.689.

5 Ранжирование текстов литературных произведений

Для исследования задачи ранжирования текстов были выбраны тексты литературных произведений, так как это естественным образом сформировавшийся набор данных, упорядоченный во времени. Не удалось найти статей, где бы решалась задача упорядочивания текстов литературных произведений методами машинного обучения.

Возникает вопрос, существует ли закономерность между тем, о чем, и как пишет автор, и его возрастом. А главное, смогут ли алгоритмы машинного обучения найти такую закономерность.

В данной работе исследованы три различных задачи ранжирования:

- По имеющимся литературным произведениям с известным порядком, различных авторов, необходимо упорядочить тексты нового автора.
- По части текстов с известным порядком одного автора, необходимо восстановить порядок остальных произведений данного автора.
- По имеющимся литературным произведениям с известным порядком во времени различных авторов, а так же по части текстов одного автора с известным порядком , необходимо восстановить порядок остальных произведений данного автора.

6 Вычислительные эксперименты

6.1 Исходные данные

Для решения задачи ранжирования текстов литературных произведений использовались тексты стихотворений русских поэтов. Необходимые данные были выкачены с сайта [11], используя язык Python и библиотеку scrapy [8]. Из полученной выборки использовались лишь тексты, содержащие не более 1014 символов. Полученные данные содержат 8871 стихотворений 112-ти русских поэтов. (Код с проведенными экспериментами [1])

На Рис. 1 и Рис. 2 показано распределение следующих признаков: год рождения авторов и возраст авторов, в котором было написано произведение. Видно, что данные содержат стихи, написанные после 18 века.

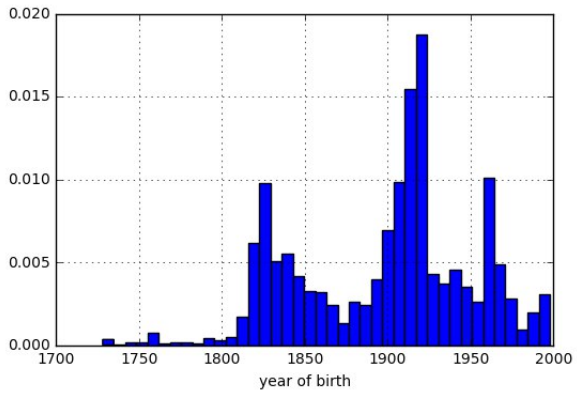

Рис. 4: Год рождения авторов

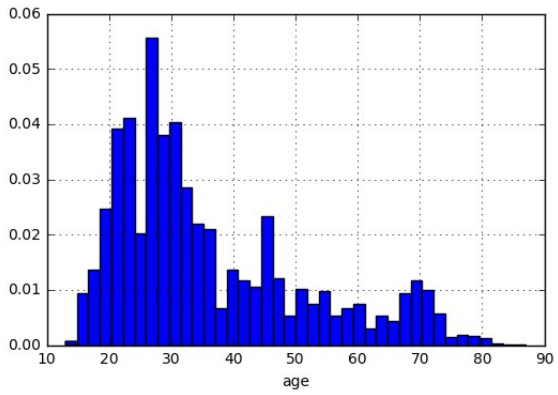

Рис. 5: Возраст написания произведения

6.2 Подходы к решению

При решении задачи ранжирования текстов использовался попарный подход, в нем каждый объект представляет собой тексты двух произведений одного автора. Ответ для данного объекта 1, если текст первого произведения был написан раньше, чем текст второго, и 0, если иначе. При таком подходе доля объектов класса 1 равна доле объектов класса 0.

Разбиение на обучение и контроль происходило следующими способами:

1. Выбиралось несколько случайных авторов, и из их стихотворений формировалась тестовая выборка вышеописанным способом. Из оставшихся произведений формировалась обучающая выборка. Т. е. при таком способе в обучающую и тестовую выборку попадают произведения разных авторов.

2. Случайно выбирались произведения и из них формировалась тестовая выборка вышеописанным способом (путем сопоставления каждого произведения одного автора с каждым). Из оставшихся произведений формировалась обучающая выборка. При таком способе в обучающую и тестовую выборку могут попадать произведения одних и тех же авторов.

3. Для каждого автора в отдельности случайно выбиралось 20% его произведений и из них формировалась тестовая выборка. Из остальных 80% произведений данного автора формировалась обучающая выборка. В таком способе в обучающую и тестовую выборку попадают тесты одного поэта. Итоговое качество усреднялось по всем авторам.

6.3 Традиционные методы

Для получения простого базового качества решения данной задачи тексты были переведены в матрицу признаков путем использования TF-IDF и буквенных Ngram. Перед формированием матрицы признаков текст был предобработан: символы приведены к нижнему регистру, удалены знаки препинания, проведена лемматизация. Далее, используя полученную разряженную матрицу, решалась задача бинарной классификации с помощью логистической регрессии. Качество оценивалось по метрике Accuracy. Тем самым мы оцениваем долю верно упорядоченных пар стихотворений внутри автора.

Другой способ решения данной задачи основан на эвристических признаках. В ходе работы удалось придумать следующие признаки:

- количество слов в тексте
- количество слов в заголовке

- количество точек, запятых, восклицательных и вопросительных знаков
- среднее количество слов в строке
- среднее количество точек, запятых, восклицательных и вопросительных знаков в строке
- количество существительных, прилагательных и глаголов в тексте
- среднее количество существительных, прилагательных и глаголов в строке

С помощью полученной признаковой матрицы решалась задача бинарной классификации с использованием алгоритмов логистической регрессии и случайного леса.

В Таблице 1 приведено качество базового решения задачи, усредненное по трем запускам (это необходимо, так как выборка на обучение и контроль разбивалась случайным образом) для первого и второго способа разбиения на обучение и контроль.

Таблица 1: Данные

	TFIDF+Ngram 4 букв	TFIDF+мешок слов	Эвристич. признаки
Способ №1	0.55	0.54	0.51
Способ №2	0.59	0.57	0.53

В способе №3 изменен принцип разбиения на обучающую и тестовую выборку. Теперь в обучение и в тестирование попадают тексты лишь одного из авторов. Т. е. для каждого автора задача решается отдельно, независимо от остальных. В данном способе использовались тексты авторов, написавших более 100 произведений.

Помимо базового решения с использованием мешка слов и TFIDF, и решения, основанного на эвристических признаках, применялся метод LSA, для снижения размерности.

В Таблице 2 приведено качество работы полученных методов.

Таблица 2: Данные

	Среднее Accuracy по всем авторам
TFIDF + мешок слов + log regression	0.68
LSA + log regression	0.68
LSA + log regression + эвристич. признаки	0.69

Для каждого из авторов, имеющих в выборке более 100 стихотворений, в Таблице 3 приведен результат работы полученного алгоритма.

Таблица 3: Данные

Размер тестовой выборки	Accuracy	Автор
(1892, 50)	0.70	Александр Блок
(16002, 50)	0.63	Александр Пушкин
(420, 50)	0.45	Алексей Апухтин
(21170, 50)	0.701	Анна Ахматова
(462, 50)	0.755	Аполлон Майков
(2756, 50)	0.745	Афанасий Фет
(506, 50)	0.616	Булат Окуджава
(1722, 50)	0.687	Валерий Брюсов
(2550, 50)	0.605	Владимир Высоцкий
(420, 50)	0.652	Владислав Ходасевич
(1056, 50)	0.646	Илья Эренбург
(3906, 50)	0.754	Иосиф Бродский
(3540, 50)	0.722	Константин Бальмонт
(15006, 50)	0.73	Марина Цветаева
(420, 50)	0.633	Михаил Лермонтов
(2162, 50)	0.677	Николай Гумилев
(420, 50)	0.7190	Осип Мандельштам
(2652, 50)	0.719	Федор Сологуб
(2970, 50)	0.673	Федор Тютчев
(2162, 50)	0.544	Эдуард Асадов

6.4 Нейронные сети

6.5 Сверточная нейронная сеть

Нейронные сети на данный момент активно применяются в задачах автоматической обработки текстов. Для данной задачи использовалась сверточная нейронная сеть с посимвольным подходом. Архитектура данной сети описана в статье [3]. Из каждого стихотворения было выбрано первые 507 символов, затем каждые два укороченных стихотворения одного автора объединялись в 1 текст длиной 1014 симво-

лов. В данном подходе использовался алфавит из символа перевода строки, цифр, русских букв и знаков препинания.

Известно, что использование нейронной сети с архитектурой, описанной в статье, дает лучшее качество по сравнению с традиционными методами, если порядок числа объектов в выборке не менее 10^6. В способе 1 и 2 в обучающей выборке более 10^6 объектов. В способе 3 для каждого автора обучение происходит отдельно и количество объектов может быть от 10^2 до 10^4.

Качество по метрике Accuracy приведено в Таблице 4. В способе №3 качество усреднялось по всем авторам. Из таблицы видно, что данная нейронная сеть не показала результат лучший, чем традиционные методы ни для одного из способов.

Таблица 4: Данные

	Accuracy
Способ №1	0.56
Способ №2	0.58
Способ №3	0.50

6.6 LSTM сеть

Архитектура LSTM сети для решения данной задачи была взята с сайта [4]. Перед использованием нейронной сети каждый текст был преобразован в матрицу размером 140×300. Из каждого текста выбрано 140 первых слов и для каждого слова получен вектор длины 300, с помощью предобученной модели Word2Vec. Предобученная модель была скачана с сайта [10].

Каждая ячейка LSTM сети принимает на вход матрицу размером 140×300, а возвращает вектор размером 1×64. Качество по метрике Accuracy приведено в Таблице 4. Из таблицы видно, что данная нейронная сеть показала результат лучший, чем традиционные методы. В способе №3 качество усреднялось по всем авторам.

7 Выводы

В ходе работы удалось показать, что:

Таблица 5: Данные

Способ №1	0.56
Способ №2	0.60
Способ №3	0.71

- Существует закономерность между тестом стихотворений одного автора и возрастом этого автора.
- Алгоритмы машинного обучения способны распознавать такую закономерность с качеством 0.7 по метрике Accuracy.
- Алгоритмам машинного обучения удалось найти закономерности между текстом стихотворений всех поэтов и их возрастом с качеством около 0.5 по метрике Accuracy.
- Сверточные нейронные сети с посимвольным подходом показали худшее качество, чем LSTM сеть. Возможно, это связано с тем, что для глубокой сверточной нейронной сети необходимо больше данных, или с тем, что LSTM сеть использовала предобученное с помощью Word2Vec векторное представление слов.

8 Заключение

В данной работе была рассмотрена задача ранжирования текстов на примере задачи ранжирования текстов литературных произведений, так как это естественным образом сформировавшийся набор данных, упорядоченный во времени. Рассмотренная задача оказалась задачей, которую ранее никто не решал. При решении задача ранжирования была сведена к задачи классификации текстов с помощью попарного подхода.

В работе проведено исследование основных методов классификации текстов, в том числе и методов классификации текстов с использованием нейронных сетей.

В ходе экспериментов было рассмотрено несколько постановок задачи:

1. По имеющимся литературным произведениям с известным порядком, различных авторов, необходимо упорядочить тексты нового автора.

2. По части текстов с известным порядком одного автора, необходимо восстановить порядок остальных произведений данного автора.

3. По имеющимся литературным произведениям с известным порядком во времени различных авторов, а так же по части текстов одного автора с известным порядком , необходимо восстановить порядок остальных произведений данного автора.

По имеющимся литературным произведениям с известным порядком, различных авторов, упорядочить тексты нового автора удалось лишь с качеством 0.56 по метрике Accuracy. По части текстов с известным порядком одного автора, восстановить порядок остальных произведений данного автора удалось лишь с качеством 0.6 по метрике Accuracy. По имеющимся литературным произведениям с известным порядком во времени различных авторов, а так же по части текстов одного автора с известным порядком , восстановить порядок остальных произведений данного автора удалось с качеством более 0.7 по метрике Accuracy.

Было произведено сравнение качества классификации традиционными и нейросетевыми методами. Выявлено, что LSTM сеть справляется с задачей лучше, чем все остальные рассмотренные методы.

Список литературы

[1] Code. — url https://github.com/AnastasiaRysmyatova/research/tree/master/code.

[2] Efficient estimation of word representations in vector space / Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean // ICLR. — 2013.

[3] From word embeddings to document distances / M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger // ICML. — 2015.

[4] Keras: Deep learning library for theano and tensorflow. — url https://keras.io/getting-started/functional-api-guide//.

[5] Liu, T.-Y. Learning to rank for information retrieval / Tie-Yan Liu // WWW. — 2009.

[6] S, J. K. A statistical interpretation of term specificity and its application in retrieval / Jones K. S. — 1972.

[7] Schmidhuber, S. H. J. Long short-term memory / Sepp Hochreiter; Jьrgen Schmidhuber // Neural Computation. — 1997.

[8] Scrapy. an open source and collaborative framework for extracting the data you need from websites. — url https://scrapy.org/ .

[9] Zhang, X. Character-level convolutional networks for text classification / Xiang Zhang, Junbo Zhao, Yann LeCun // In Advances in Neural Information Processing Systems. — 2015. — Feb. — 649 - 657 p.

[10] «pre-trained word vectors on wikipedia». — url https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md.

[11] «world-art». — url http://www.world-art.ru/ lyric/.

[12] «yandex matrixnet». — url https://yandex.ru/company/technologies/matrixnet.

[13] Воронцов, К. В. Курс лекций по машиному обучению / К. В. Воронцов. — 2015.