МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ

им. А. И. ГЕРЦЕНА»

Институт информационных технологий и технологического образования

Кафедра информационных систем

Направление подготовки   
09.04.02 – Информационные системы и технологии

**Отчёт по учебной практике**

магистерской программы

“Анализ и синтез информационных систем”

Период прохождения практики: 02.02.2021-24.02.2021

**Исполнитель**: магистрант 1 курса

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Момот Даниэль Михайлович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Ф.И.О.)*

**Проверяющие:**

Руководитель от профильной структуры организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(должность) (Фамилия И.О.)*

Научный руководитель магистерской работы

\_\_\_\_проф. каф. ИС\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_Фомин В.В.\_\_\_\_\_\_\_\_\_\_

*(должность) (Фамилия И.О.)*

Руководитель практики доц. каф. ИС Кудрявцева И.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(оценка) (подпись)*

Санкт-Петербург

2021

Оглавление

[ВВЕДЕНИЕ 3](#_Toc65251211)

[Описание существующих подходов, техник и методов классификации текста 4](#_Toc65251212)

[Формирование категорий 4](#_Toc65251213)

[Индексация документов 4](#_Toc65251214)

[Обучение классификатора 6](#_Toc65251215)

[Оценка качества классификаторов 11](#_Toc65251216)

[Анализ устройства существующих классификаторов текста 13](#_Toc65251217)

[ЗАКЛЮЧЕНИЕ 14](#_Toc65251218)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 15](#_Toc65251219)

# ВВЕДЕНИЕ

Важной частью магистерской программы “Анализ и синтез информационных систем” 1 курса является учебная практика, проводимая в текущем учебном году в период 02.02.2021 – 24.02.2021. Ее целью является закрепление и углубление теоретической подготовки обучающегося и приобретение им первичных профессиональных умений и навыков. Кроме того, результаты учебной практики в дальнейшем будут использованы при написании магистерской диссертации. Практика проходится на кафедре информационных систем РГПУ им А. И. Герцена.

Программа практики состоит из трех этапов. На организационном этапе создается и согласовывается с научным руководителем план практики, на основном этапе магистрант выполняет зафиксированные в плане задачи практики, на заключительном этапе завершается и предоставляется для проверки научному руководителю и руководителю практики отчет по практике, а научный руководитель оценивает отчет и пишет отзыв.

При составлении плана учебной практики были поставлены и согласованы с научным руководителем следующие задачи.

1. Описание существующих подходов, техник и методов классификации текста. Анализ их преимуществ и недостатков.
2. Анализ устройства существующих классификаторов текста по материалам научных работ и описаний систем.
3. Выполнение выбора в пользу одного (или нескольких) из методов для будущей разработки собственного классификатора.
4. Исследование метрик оценки качества классификаторов текста.
5. Анализ архитектур существующих классификаторов текста с указанием их преимуществ и недостатков.

# Описание существующих подходов, техник и методов классификации текста

Задача классификации текстов формулируется следующим образом. Существуют заранее заданные категории, и необходимо отнести документ к одной из них.

В процессе классификации текстов можно выделить следующие этапы.

1. Формирование категорий.
2. Индексация документов.
3. Обучение классификатора.
4. Оценка качества классификатора.

Осветим подходы к каждому из этапов в отдельности.

## Формирование категорий

Данный этап является одним из наиболее важных, так как неверный выбор категории приведет к бесполезности всех дальнейших этапов. В работе [1] авторы приходят к выводу, что для достижения оптимальных результатов требуется человеческое вмешательство.

Этот этап также можно назвать первичной классификацией. Выбирается маленькое подмножество обучающей выборки, и оно анализируется для оптимального набора категорий.

## Индексация документов

Этап индексации документов состоит в приведении текстов к форме, удобной для дальнейшей обработки. Такими формами являются:

1. Векторная модель. Строится векторное пространство, единое для всех документов коллекции. Каждая размерность соответствует уникальному слову данной коллекции. Документу из коллекции сопоставляется вектор, значение каждой координаты – количество вхождений соответствующего слова в текст [2].
2. Терм-документная матрица. Она является обобщением векторной модели. Ее строки соответствуют документам, столбцы – словам, а в ячейках – некоторая функция «важности» слов (см ниже).
3. Мешок слов (bag of words). Текст представляется как мультимножество уникальных для этого текста слов. Для каждого слова хранится количество его вхождений в текст.
4. Мешок N-грамм (w-shingling). Хранятся уникальные для этого текста последовательности из N подряд идущих слов и количество вхождений каждой такой последовательности. При N=1 подход совпадает с мешком слов. Такой подход позволяет частично компенсировать потерю упорядоченности слов.

Для эффективной индексации коллекции документов необходимо научиться решать ряд более простых задач. А именно:

1. Токенизация – разбиение текста на составные части: абзацы, предложения, слова.
2. Нормализация – приведение слов текста к начальной форме. Это позволяет распознать, что две формы слова соответствуют одной лемме – начальной форме – и отнести их к одной размерности вектора или мешка слов. Данная задача является весьма трудной. В зависимости от желаемого качества, она решается с помощью одного из следующих подходов.
   1. Стемминг – отбрасывание окончания слова. Это наименее интеллектуальный, но наиболее простой способ нормализации.
   2. Лемматизация – приведение слова к начальной форме. Необходимо уметь определять часть речи (PoS-tagging). Этот способ нормализации лучше, чем стемминг, но не устойчив к омонимии.
   3. Устранение омонимии – выбор правильной начальной формы (из нескольких возможных) для приведения к ней.
3. Чистка шума (стоп-слов) – удаление слов, не несущих информации (предлоги, артикли). Этим методом необходимо пользоваться с осторожностью, так как он может ухудшить результаты в случае работы с мешком N-грамм, а также исказить стилевой окрас предложения [3].

На этапе индексирования документов обычно выполняется выкидывание лишних признаков (features) – не дающих большого количества информации, но усложняющих вычисления. Это выполняется с помощью следующих методов:

1. Чистка слишком редких слов. Такие слова встречаются в очень небольшом количестве текстов из коллекции, а значит не слишком помогут в классификации остальных текстов.
2. Чистка «неважных» слов. Для каждого слова вычисляется метрика важности, и наименее важные слова отбрасываются. Обычно используется TF-IDF метрика, но также могут использоваться CHI и другие метрики [4].
3. Стандартные техники понижения размерности, такие как метод главных компонент, анализ независимых компонент.

## Обучение классификатора

Данный этап является классической задачей классификации. На вход подается текст в удобном для классификации виде (векторная модель, мешок слов, N-граммы), необходимо определить, к какому из заранее заданных классов относится входной текст. При проектировании и обучении классификатора можно выделить следующие группы подходов.

1. Метрические методы. К этой группе относится метод ближайших соседей и его модификации. Основная идея состоит в том, что на первом этапе создаются кластеры документов, соответствующие категориям, а затем классификатор относит очередной классифицируемый документ к той категории, к которой принадлежат ближайшие к нему документы обучающей выборки.
2. Вероятностные методы. К этой группе относится наивный байесовский классификатор и его модификации. Результатом работы классификатора являются вероятности принадлежности документа к категориям. Он основан на предположении о том, что признаки вносят независимый вклад в вероятности. По обучающей выборке вычисляется вклад значений признаков в итоговые вероятности, и с учетом значений признаков на конкретном документе дается прогноз принадлежности категориям.
3. Линейные методы классификации. К этой группе относится метод опорных векторов (SVM) и его модификации. Строится гиперповерхность, разделяющая объекты выборки (представляемые точками) на категории оптимальным образом.
4. Методы деревьев решений. Дерево решений – это ориентированное дерево, в узлах которого признаки, а в исходящих ветвях – значения этих признаков. В листьях находятся результаты. Классификация происходит путем спуска по дереву вплоть до листьев. Деревья решений могут действовать по одному или в ансамбле. Также к этой группе можно отнести экспертные системы, работающие на основе правил.
5. Нейросетевые методы. Нейросети предоставляют огромное разнообразие вариаций использования, обеспечивая гибкость настройки. Наиболее часто в классификации текстов используются архитектуры сверточная, рекурсивная, рекуррентная и ее разновидность LSTM.

Изначальные допущения и обусловленные ими достоинства и недостатки описанных методов представлены в Таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Метод классификации | Изначальное допущение | Преимущества | Недостатки |
| Метрический | Категориям соответствуют кластеры документов | Простота  Оптимален, если документы хорошо кластеризуются | Неопределенность количества и расположения кластеров |
| Вероятностный | Независимое влияние признаков | Простота  Высокая скорость  Оптимален, если признаки независимы | Часто низкое качество на реальных данных |
| Линейный | Существование разделяющей гиперплоскости |  |  |
| Дерево решений | Существует набор правил, позволяющих высоковероятно классифицировать | Простота  Не требуется много признаков | NP-полнота построения оптимального варианта требует эвристик  Переобучение (чувствительность к шуму, нестабильность) |
| Нейросетевой | Существование нелинейного закона |  |  |

Таблица 1. Изначальные допущения, преимущества и недостатки методов классификации.

В обработке языка также используется ряд техник, упрощающих задачу классификации текстов. Рассмотрим их подробнее.

**Латентно-семантический анализ (ЛСА).** На вход классификатор принимает терм-документную матрицу. Этот метод позволяет извлечь взаимные зависимости между словами и документами, при этом «испортив» изначальную матрицу для максимального уменьшения ее размерности.

Чаще всего используется SVD-разложение матрицы. Оно позволяет разложить матрицу на множество ортогональных матриц, линейная комбинация которых является хорошим приближением изначальной матрицы.

Согласно теореме о сингулярном разложении [5], любая вещественнозначная прямоугольная матрица может быть представлена произведением матриц:

Где матрицы и – ортогональные, S – диагональная матрица, значения на ее диагонали называются сингулярными числами матрицы .

Такое разложение обладает важным свойством: если в матрице оставить наибольших сингулярных чисел, а в матрицах и – столбцы, соответствующие этим значениях, то матрица , построенная как произведение полученных , , , будет наилучшим приближением исходной матрицы среди всех матриц ранга .

Основная идея ЛСА в том, чтобы применить SVD-разложение к терм-документной матрице. Это позволяет оставить линейно-независимых компонент, отражающих основные зависимости исходной матрицы. В результате, каждый терм и каждый документ представляются некоторой точкой в -мерном пространстве, где очень невелико по сравнению с исходным количеством документов и слов.

ЛСА идеально подходит для кластеризации (отличается от классификации тем, что категории заранее неизвестны). Однако в классификации он также применяется, что описано в работах [6][7]. Он относится к метрическим подходам.

Существуют также его улучшение: вероятностный ЛСА (лучше учитывающий случайное распределение входных значений), для которого тоже есть оптимизация LDA (Latent Dirichlet allocation), которая исходит из того, что тематики распределены по распределению Дирихле.

Как сам ЛСА, так и его модификации известны давно. ЛСА изобретен в 1988, ВЛСА – в 1999, LDA – в 2003.

# Оценка качества классификаторов

Оценка качества классификаторов во многом схожа таковой в других областях машинного обучения, но имеет несколько нюансов.

Существует 4 возможных исходов применения классификатора по конкретному входу: два верных результата и два ошибочных результата. Они приводятся в таблице 3, называемой матрицей несоответствия (confusion matrix). Исходы обозначаются TP, FP, FN, TN, и так же обозначается количество входов с соответствующим результатом.

|  |  |  |
| --- | --- | --- |
|  | Да (P) | Нет (N) |
| Предсказано Да | TP (true positive) | FP (false positive) – ошибка I рода |
| Предсказано Нет | FN (false negative) – ошибка II рода | TN (true negative) |

Таблица 2. Матрица несоответствия.

Две основные метрики оценки классификаторов – это точность (precision), полнота (recall). Для их объединения обычно берется F-мера (F-measure), иногда взвешенная F-мера. Они вычисляются по следующим формулам.

Для корректного сравнения результатов классификаторов с использованием одной из метрик необходимо обеспечить выполнение условий:

* Сравнение производится на одинаковых коллекциях (одинаковые множества документов и категорий).
* Учебная, тестовая и проверочная выборки выделены одинаково (либо – что рекомендуется – кросс-валидация с одинаковыми параметрами).
* Документы проиндексированы одинаково.

# Анализ устройства существующих классификаторов текста

В предыдущих разделах описаны подходы к классификации текста и способы оценки классификаторов. В данном разделе рассмотрим примеры устройства реальных классификаторов и результаты, которые они демонстрируют. Рассматриваются решения, созданные не ранее 2010-го года, так как ввиду быстрых темпов развития области более старые реализации не отражают современных тенденций.

Большинство современных решений используют нейросетевые подходы.

В работе [8] использована сверточная нейросеть для классификации сообщений, которые пользователь посылает чат-боту (беседа, запрос, контакт, информация по доставке). Нейросеть имеет 5 слоев (входной, свертки, субдискретизации, полносвязный скрытый, выходной). Реализация на Python с помощью библиотеки Keras. Результаты сравнивались с построенными для этого классификаторами на LSTM и на SVM (метод опорных векторов). Оказалось, что сверточная нейросеть верно классифицировала 54-62% сообщений, LSTM – 48-59%, SVM – 43-55%.

В работе [9] использована LSTM-нейросеть для классификации тональности англоязычных и русскоязычных текстов. Нейросеть имеет 6 слоев (входной, векторного преобразования слов, регуляризации, сверточная сеть, LSTM-слой, выходной). Результаты оценки качества в статье не приводятся.

В работе [10] проведено сравнение различных классификаторов для классификации корпуса текстов Reuters-21578. Сравнивались классификаторы: наивный Байесовский (NB), k ближайших соседей (KNN), SVM, логистическая регрессия, а также нейросети с архитектурами FFBP, RNN, DAN2, CNN. Наилучшие результаты по соотношению точности, полноты и F-меры показали SVM и CNN (сверточная нейросеть). Наиболее быстрым (ожидаемо) оказался NB.

Сверточные нейросети считаются одной из лучших архитектур. Проблемой является большое количество варьируемых параметров, влияющих на работу сети.

В то же время, SVM, NB, KNN имеют значительно меньше вариаций и изменяемых параметров. Ввиду этого, представляется разумным «идти от простого к сложному»: реализовать классификаторы на SVM и других простых методах и использовать их как базу для сравнения с ними более сложных нейросетевых моделей.

В целом, критерии выбора конкретной архитектуры зависят от решаемой задачи. Принято пробовать несколько архитектур и выбирать из них ту, которая лучше работает на конкретных данных. Наиболее перспективными выглядят сверточные нейросети, но это – поле для экспериментов.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения учебной практики выполнен обзор существующих подходов и методов классификации текстов на естественном языке. Проведен анализ их преимуществ и недостатков. Рассмотрены метрики оценки качества классификаторов. Рассмотрены примеры архитектур нейронных сетей из научной литературы и сделан вывод о том, что наилучшие результаты часто показывают сверточные нейросети, но архитектура выбирается исходя из конкретных данных.

Таким образом, план выполнен в полном объеме.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Oliveira E. Automatic classification of journalistic documents on the Internet. / E. Oliveira, D. B. Filho // Transinformacao – 2017. – V. 29, № 3 – P. 245–255.
2. G. Sidorov. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model / G. Sidorov, A. Gelbukh, H. Gomez-Adorno, and D. Pinto // Computacion y Sistemas – 2014 – Vol. 18, No. 3 – pp. 491-504.
3. Yu B. An evaluation of text classification methods for literary study / B. Yu // Literary and Linguistic Computing – 2008. – V. 23, № 3 – P. 327–343.
4. Deng Z.H. A comparative study on feature weight in text categorization / Z. H. Deng, S.W. Tang, D. Q. Yang, M. Zhang, L. Y. Li, K. Q. Xie // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) – 2004. – V. 3007. – P. 588–597.
5. Голуб Дж., Ван Лоун Ч. Матричные вычисления. М.: «Мир», 1999.
6. Kou, G. An Application of Latent Semantic Analysis for Text Categorization. / G. Kou, Y. Peng // International Journal of Computers Communications & Control. – 2015 – 10 (3) – P. 357 – 369.
7. Li S. Latent Semantic Analysis & Sentiment Classification with Python [Электронный ресурс] // Towards data science: A Medium publication sharing concepts, ideas and codes. 8.09.2018. URL: https://towardsdatascience.com/latent-semantic-analysis-sentiment-classification-with-python-5f657346f6a3 (дата обращения: 23.02.2021).
8. Воробьев Н. В. Классификация текстов с помощью сверточных нейронных сетей. / Н. В. Воробьев, Е. В. Пучков // Молодой исследователь Дона. – 2017 – № 6 (9) – стр. 2 – 7.
9. Лыченко Н. М. Классификатор тональности текста на основе LSTM-нейронной сети. / Н. М. Лыченко, А. В. Сороковая // Вестник Кыргызско-Российского славянского университета. 2019 – № 12 (19) – стр. 87-92.
10. Батура Т. В. Методы автоматической классификации текстов. / Т. В. Батура // Программные продукты и системы. 2017 - № 1 (30) – стр. 85-99.