Санкт-Петербургский государственный университет

**ПАВЛИКОВА Владислава Станиславовна**

**Выпускная квалификационная работа**

# Автоматическое выявление социолингвистических данных на материале дневников проекта «Прожито»

Уровень образования: магистратура
Направление 45.04.02 «Лингвистика»
Основная образовательная программа ВМ.5805. «Компьютерная и прикладная лингвистика»

Научный руководитель:
доцент, Кафедра математической лингвистики,
Митренина Ольга Владимировна

Рецензент:
доцент, Школа лингвистики, Национальный исследовательский университет «Высшая школа экономики»,
Слюсарь Наталья Анатольевна

Санкт-Петербург
2024

# ОГЛАВЛЕНИЕ

# ВВЕДЕНИЕ

Задача составления профиля автора, социолингвистического портрета, призвана выявлять демографическую информацию об авторах текстов, такую как возраст, пол, уровень образования на основе анализа текста. Важность этой задачи неуклонно возрастает в современном мире, где анализ данных становится центральным элементом в стратегическом планировании и принятии решений. Профилирование авторов позволяет не только углубить понимание личности создателя текста, но и находить применение в самых разных областях: от правоохранительной деятельности и судебно-медицинских экспертиз до маркетинга и персонализированной рекламы. Особенно велика ценность таких анализов при разработке систем, способных определять авторство анонимных или псевдонимных текстов.

**Целью исследования** является подбор методов и архитектур алгоритмов глубинного обучения, с наиболее высокой точностью предсказывающих скрытые атрибуты по тексту, такие как гендер, возраст автора и время создания записи. Объектом является связь особенностей языка и демографических признаков в дневниковых записях. Предмет – анализ связи между особенностями языка в дневниковых записях и такими признаками, как гендер, возраст авторов текстов и время создания записи. Для достижения поставленной цели исследования необходимо учитывать разнообразные факторы, влияющие на структуру и содержание текстов. Например, возраст автора может сказываться на выборе лексики, структуре предложений и общем стиле письма. Пол также может оказывать влияние на языковые особенности, женщины и мужчины могут выражать свои мысли и эмоции по-разному. Особое внимание уделяется изучению того, как особенности использования языка коррелируют с демографическими признаками авторов, и как эта информация может быть интегрирована в алгоритмы машинного обучения для создания точных и эффективных предиктивных систем. Примеры таких исследований включают анализ языковых особенностей в

дневниках, публикациях в социальных сетях и других письменных источниках, что открывает новые возможности для научного сообщества, программ сохранения исторических и культурных текстов и бизнеса в оценке и моделировании поведенческих паттернов различных групп населения.

Исследование проводится на **материале** дневниковых записей проекта «Прожито»[1] – цифрового архива личных документов из частных собраний. Корпус содержит записи более 9 тысяч авторов, начиная с XVIII столетия. Дневниковые записи представляют собой уникальный источник данных, отражающих индивидуальные мысли, эмоции и опыт каждого автора.

Для достижения цели были поставлены следующие **задачи**:

1. Исследовать вопрос социолингвистического портрета как феномен и специфику дневниковых записей как лингвистического источника;
2. Составить характеристику корпуса дневников проекта «Прожито»;
3. Изучить существующие решения в области бинарной и мультиклассовой классификации текстов в машинном и глубинном обучении;
4. Собрать корпус дневниковых записей проекта «Прожито» для обучения моделей машинного обучения;
5. Осуществить предобработку текстов для подачи в модель;
6. Провести сравнительный анализ различных подходов к выделению признаков и архитектур моделей на данных дневниковых записей для определения наилучшего метода;
7. Проанализировать результаты экспериментов и оценить производительность моделей;
8. Подготовить выводы о точности в предсказании скрытых атрибутов по тексту, о связи между языковыми особенностями и

[1] Прожито // Сайт проекта центра. [Б. м.]. URL: https://prozhito.org/page/corpus/ (дата обращения: 15.02.2024).

демографическими признаками и о подходящих алгоритмах для их предсказания.

В процессе достижения поставленной цели использовались следующие **методы**: моделирование для построения алгоритмов глубинного обучения, эксперимент для проверки эффективности моделей, сравнение различных подходов, статистический анализ и социолингвистический анализ. При описании материала использовался метод научного описания, включающий приемы классификации единиц, анализа, количественных подсчетов.

**Новизна** представленного исследования заключается в обращении к комплексному построению моделей предсказания социолингвистических признаков с использованием алгоритмов машинного и глубинного обучения. Дневниковые записи представляют собой новый материал для автоматического анализа лингвистических данных. Также на данный момент не существует модели, способной предсказывать временной период создания текста на основе его содержания.

**Практическая значимость** данного исследования заключается в использование лучших результирующих моделей для предсказания пола, возрастных групп авторов, временного периода создания текста. Эти модели могут быть использованы в качестве предобученных моделей для предсказания демографических признаков по тексту. Знание о языковых особенностях текстов, связанные с гендером, возрастом автора и временем создания, может быть полезно для социологов, психологов, культурологов и других специалистов, изучающих взаимосвязь между языком, социальными атрибутами и историческим контекстом. Исследование может быть полезным для программ сохранения исторических и культурных текстов. Анализ языковых особенностей в дневниках и других письменных источниках позволяет лучше понять контекст и значение этих текстов, а также помогает в их классификации и моделировании. Использованные инструменты при написании моделей, различные комбинации параметров и архитектур могут

способствовать развитию методов машинного обучения и улучшению точности предсказаний на текстовых данных.

Применяя алгоритмы машинного обучения для анализа больших объемов дневниковых данных, можно получить представление о том, как использование языка варьируется в различных социальных контекстах и у разных людей. Такой подход позволяет более систематически и количественно изучать социолингвистические данные, что может способствовать более глубокому пониманию языковых вариаций и изменений.

Теоретической базой исследования стали работы, посвященные:

- социолингвистике, профилированию авторов текстов (W. Labov, J. Fishman, А. Д. Швейцер, Л. Б. Никольский, Белл Р., Л. П. Крысин, В. И. Беликов, Т. М. Николаева, D. Nguyen и др.), дневниковым записям в виде лингвистического источника (А. А. Зализняк, Е. В. Богданова, М. А. Мельниниченко, Н. Б. Тышкевич, А. А. Бызов и др.);

- задачам автоматической классификации текстов (H. Li, D. Jurafsky, J. H. Matrin, Т. В. Батура и др.), моделям реализации классификации и кластеризации текстов и их оценки машинного и глубинного обучения (Ш. Шалев-Шварц, Ш. Бен-Давид, А. А. Казанцева, М. В. Прохоров, П. С. Худякова, Й. Гольдберг, I. Goodfellow, Y. Benigo, A. Courville и др.);

- моделям предсказания гендера (А. В. Кириллина, N. Cheng, R. Chandramouli, А. Г. Сбоев) и возраста авторов текста (D. C. Asogwa, A. A. Morgan-Lopez).

**Структура работы:** исследование состоит из введения, основной части, заключения, списка использованных источников, в том числе электронных ресурсов, и приложений.

В первой главе работы исследуется тема социолингвистического портрета как лингвистического феномена с акцентом на методы и подходы составления авторского профиля. Рассматривается специфика дневниковых

записей как источника лингвистических данных, предметно изучаются особенности текстов корпуса «Прожито».

Во второй главе рассматривается классификация текстов с применением методов машинного и глубинного обучения, изучаются различные подходы к предварительной обработке и векторизации текстов, способы оценки моделей классификации.

В третьей главе описываются эксперименты по классификации демографических атрибутов дневниковых записей, включая сбор корпуса, предобработку текстов, выделение признаков и саму классификацию по таким признакам как пол, возрастные группы автора и временные периоды создания записи.

В заключении приводятся результаты проведенного исследования.

В список использованной литературы включены теоретические источники, статьи, электронные ресурсы и работы, материал которых пересекается с настоящим исследованием. Список включает 70 единиц, в том числе 38 на английском языке и 10 электронных ресурсов.

**Апробация работы** проводилась в рамках открытой международной олимпиады студентов и молодых специалистов Petropolitan Science (Re)Search, был представлен отрывок из работы с основными экспериментами и осуществлена его публичная защита. По результатам всех этапов было занято призерское место. Также эксперименты по предсказанию возраста частично были представлены на семинаре по NLP от лаборатории естественного языка Яндекса и НИУ ВШЭ. Были заданы вопросы про специфичность дневниковых записей как выбранного материала, проблему достоверности данных.

# 1 Социолингвистический анализ текстов: методы и подходы в составлении портрета автора

Социолингвистика играет важную роль в современном лингвистическом исследовании, позволяя углубленно изучать взаимосвязь между языком и обществом. Один из ключевых методов сбора данных в социолингвистике – это анализ текстов, который позволяет выявить различия в языковых проявлениях в зависимости от социальных и демографических характеристик авторов. С развитием технологий и увеличением объемов доступных данных стал использоваться также автоматический анализ в пределах такой области как компьютерная социолингвистика. В данной главе мы рассмотрим методы и подходы к социолингвистическому анализу текстов, понятие социолингвистического портрета и особенность дневниковых записей как источника для его построения.

## 1.1 Введение в социолингвистику и ее значение для анализа текстов

Социолингвистика как научная дисциплина имеет различные интерпретации и определения среди ученых, что отражает ее мультидисциплинарный и развивающийся характер. Например, У. Лабов представляет определение, сосредотачиваясь на изучении структуры языка и языковых изменений на основе анализа ежедневного употребления языка[2]. Это подход подчеркивает важность эмпирических данных и первичного анализа языкового поведения. Влиятельны также работы Дж. Фишмана, который в своих работах видит социолингвистику как науку, изучающую взаимодействие признаков языковых вариантов, их функций и говорящих, подчеркивая динамический характер этих взаимодействий в рамках языковых сообществ[3]. Разнообразие определений социолингвистики отражает её

[2] Labov W. The Study of Language in its Social Context. In «Studium Generale», 23. 1970. P. 30-87.

[3] Fishman J. Preface. Advances in the Sociology of Language. The Hague : Paris, 1971. Vol. 1. P. 8-10.

сложность и многоаспектность, подчеркивая важность диалога между социологическим и лингвистическим подходами в исследовании языка и общества.

Исходя из установленной связи между лингвистикой и социологией, социолингвистика занимает уникальное положение, позволяя глубже исследовать язык как социокультурное явление. Социолингвистика, занимаясь изучением взаимодействия языка и общества, фокусируется на том, как социальные конструкты влияют на языковые практики и наоборот, как изменения в языке могут влиять на социальные структуры и взаимоотношения. Предмет социолингвистики включает изучение языковых вариаций в зависимости от социальных групп и контекстов, многоязычия, языковых идентичностей и политики, языковых стереотипов и предрассудков.

Вопрос о том, что именно должно входить в предмет социолингвистики, остается центральным местом научных дискуссий. Некоторые ученые предлагают сосредоточить внимание на строго определенных аспектах взаимодействия языка и социальных факторов, тогда как другие настаивают на более широком, интегративном подходе, который включает в себя анализ социокультурных, экономических и политических аспектов воздействия на язык.

Определение статуса социолингвистики также является объектом активных обсуждений в академических кругах. С одной стороны, существует тенденция рассматривать социолингвистику как автономную дисциплину, обладающую своими методами, теоретическими подходами и исследовательскими вопросами. С другой стороны, ее часто воспринимают как подраздел лингвистики или социологии, что подчеркивает ее мультидисциплинарный характер и зависимость от основных положений и методов данных наук[4]. Возможно, именно в этом синтезе исходных принципов

[4] Швейцер А. Д., Никольский Л. Б. Введение в социолингвистику. М., 1978. С. 48-60.

и находится уникальная ценность и сила социолингвистики как научной дисциплины.

Связь между лингвистикой и социологией является тесной и важной для понимания языка как социокультурного явления. Лингвистика изучает язык как систему знаков и правил, а социология анализирует социальные структуры и процессы. Объединение этих двух дисциплин позволяет рассматривать язык как инструмент коммуникации, который отражает и воздействует на социокультурную среду. Лингвистические исследования в социологии помогают понять, как язык формирует социальные отношения, идентичность и отдельные социальные группы.

В социолингвистике одной из ключевых характеристик является уделение внимания изучению использования языка социальными группами. Однако существуют два различных подхода к структуре группы, которые при анализе языка приводят к совершенно разным результатам. Подход может быть сфокусирован либо на индивиде, либо на группе, и лингвистические особенности взаимодействия могут быть рассмотрены с точки зрения индивидуальной или групповой динамики. Такое деление укладывается в понятия микро- и макросоциолингвистики. Первое направление фокусируется на различиях использования языка между индивидами, акцентируя внимание на автономных и несвязных идиолектах, не поддающихся группировке. Большинство лингвистических элементов в таком случае будут относится к вариациям, связанным с относительно постоянными характеристиками самого говорящего, а не определенной группы. Макросоциолингвистика рассматривает языковые различия между группами в соотношении с демографическими категориями – возраста, пола, образования, этнической принадлежности и так далее[5]. В данном исследовании вариации будут рассматриваться в терминах макросоциолингвистики.

[5] Белл Р. Социолингвистика: цели, методы и проблемы. [Пер. с англ.]. М. : Междунар. отношения. С. 45-47.

В последние годы социология и социолингвистика активно интегрируют автоматический интеллектуальный анализ для обогащения методов исследования и анализа данных. В этом контексте действует атомарное восприятие текста, то есть «текст сначала подвергается какой-то форме систематического извлечения элементов (коды, темы, термины, слова и т.д.)»[6], после чего подвергается анализу.

Социолингвистика занимает значимое место в изучении языка в его социальном контексте, объединяя лингвистику с социальными науками для исследования того, как социальные факторы влияют на языковое поведение. Этот раздел языкознания разрабатывает методы и теории, позволяющие анализировать, как различия в возрасте, поле, этнической принадлежности, социальном статусе и уровне образования влияют на способы использования языка. В контексте текстового анализа, социолингвистика предоставляет инструменты для понимания того, как эти социальные различия могут быть воплощены в письменной форме. Это осознание подчеркивает важность социолингвистики в анализе текстов, предоставляя возможность не только интерпретировать лингвистические особенности в контексте культурно-исторических условий, но и использовать эти данные для создания более эффективных коммуникационных стратегий и инструментов автоматического определения характеристик текста.

### 1.2 Социолингвистический портрет как лингвистический феномен

Изначально социолингвистический портрет представлял собой методологический инструмент, позволяющий анализировать и интерпретировать речь отдельных личностей как отражение языковых и социологических особенностей их социальной среды. Первое упоминание

[6] Бызов А. А. Интеллектуальный анализ текстов в социальных науках // Социология: методология, методы, математическое моделирование. 2019. №. 49. С. 134.

термина связано с работами Т. М. Николаевой[7], в то время как фундаментальная идея была заложена М. В. Пановым, который рассматривал обозначенную область в двух направлениях – первое касается эволюции, развития языка; второе – его функционирования в обществе. Им была разработана теория антиномий языкового развития и предложены концепции массового обследования говорящих[8]. Этот подход позволяет не только описывать индивидуальные языковые особенности, но и выявлять типичные для определенного социального слоя или профессии речевые модели.

Социолингвистические портреты находят применение в различных сферах. В частности, исследования такого рода полезны в академическом контексте для глубокого понимания взаимосвязей между языком и социальными факторами. Эти портреты также используются в лингвистической антропологии, социологии и культурологии для анализа того, как языковые практики отражают культурные, социальные и профессиональные идентичности. Интеграция социолингвистических портретов в различные дисциплины позволяет раскрывать новые аспекты в изучении языка и его функций в социальном контексте.

Социолингвистический портрет можно рассматривать как лингвистическое явление, изучающее отношения между языком и обществом. Он изучает, как язык используется и интерпретируется в различных социальных контекстах, как он отражает и формирует социальную идентичность. По сути, социолингвистический портрет представляет собой снимок языковых практик и поведения определенной группы или сообщества, проливая свет на сложные связи между языком, культурой и обществом.

Одним из ключевых аспектов социолингвистического портрета является изучение языковых вариаций и изменений. Язык не является статичной

[7] Николаева Т. М. «Социолингвистический портрет» и методы его описания // Русский язык и современность. Проблемы и перспективы развития русистики. Доклады Всесоюзной научной конференции. Часть 2. М., 1991. С. 73-75.

[8] Крысин Л. П. Очерки по социолингвистике. М. : ФЛИНТА, 2021. С. 68-74.

сущностью, он постоянно развивается и адаптируется к потребностям и желаниям своих пользователей. Социолингвистический портрет фиксирует этот динамизм, изучая различные способы, которыми язык используется в обществе. Это включает в себя анализ вариаций произношения, лексики, грамматики и моделей дискурса. Понимая эти вариации, лингвисты могут получить представление о социальных факторах, влияющих на использование языка: «Социолингвистика делает упор на то, как используют языковой знак люди, – все одинаково или по-разному, в зависимости от своего возраста, пола, социального положения, уровня и характера образования, от уровня общей культуры и т.п.?»[9].

Кроме того, социолингвистический портрет также включает в себя анализ языковых установок и идеологий. Язык – это не только средство общения, он также несет в себе социальные смыслы и коннотации. Различные языки или диалекты могут ассоциироваться с определенными социальными группами или статусами, а разговор на определенном языке может рассматриваться как показатель идентичности или принадлежности. Социолингвистические исследования часто изучают отношение и идеологию к различным языкам или диалектам, отвечая на такие вопросы, как языковая дискриминация, сохранение или изменение языка и языковая политика.

### 1.3 Дневниковые записи как источник лингвистических данных

Материалом для исследования были выбраны дневниковые записи. Дневники могут рассматриваться как многофункциональный источник данных в различных областях: исторические данные – дневники содержат записи о событиях из первых рук и могут служить важными историческими свидетельствами; документальные данные – поскольку дневники часто включают описание ежедневных событий и взаимодействий, они могут

[9] Беликов В. И., Крысин Л. П. Социолингвистика: учебник для бакалавриата и магистратуры. М., 2016. С 3.

использоваться для реконструкции повседневной жизни прошлых эпох; лингвистические данные – изучение дневников представляет ценные сведения о языковом использовании в спонтанных и неофициальных контекстах. Основное отличие дневниковых записей заключается в их способности представить язык в его наиболее естественном, необработанном виде. Дневники относят к эго-текстам – документам личного происхождения, к которым также относятся мемуары и личные письма[10].

Рассмотрим определение понятия «дневники» в контексте источника лингвистической информации и источника записей, в частности. В Малом академическом словаре А.П. Евгеньевой[11] дневники определяются как «ведущиеся изо дня в день записи каких-л. фактов, событий, наблюдений и т. п. во время путешествия, экспедиции или каких-л. занятий, деятельности». Это определение является общим, не отражающим субъективный вид письма, содержащий описание мыслей и чувств автора. В словаре литературных терминов[12] словарная статья содержит конкретизирующее описание понятия «дневники» как записей, ведущихся от первого лица, которые содержат последовательное описание событий, мыслей и чувств автора, оформленные в хронологическом порядке, современных соответствующим событиям. Дневник выделяется в отдельный жанр со следующими присущими признаками: наличие даты записи, нефикциональность текста дневника, совпадение адресата с автором, что определяет интимность письма и

[10] Филатова Н. М. Подходы к изучению эго-документов в современной исторической науке в свете" лингвистического поворота" // Документ и "документальное" в славянских культурах: между подлинным и мнимым. 2018. С. 24-40.

[11] Евгеньева А. П. Малый академический словарь. М. : Институт русского языка Академии наук СССР. 1957-1984. URL: https://rus-academic-dict.slovaronline.com/ (дата обращения: 13.11.2023).

[12] Литературная энциклопедия: Словарь литературных терминов: В 2-х т. / Под ред. Н. Бродского, А. Лаврецкого, Э. Лунина, В. Львова-Рогачевского, М. Розанова, В. Чешихина-Ветринского. М.; Л. : Изд-во Л. Д. Френкель, 1925. URL: https://rus-literary-terms.slovaronline.com/ (дата обращения: 13.11.2023).

отсутствие авторского замысла[13]. Е.В. Богданова подводит дневник под автобиографический жанр – «жанр описания собственной жизни и событий, в ней произошедших»[14]. Если сравнивать дневники с другими источниками этого жанра, например, с воспоминаниями, автобиографиями, то можно отметить, что дневники более субъективны, непосредственны, искренни. Эту специфику они обретают в том числе потому, что являются максимально приближенными к устному изложению мыслей, если рассматривать в плоскости дихотомии языка и речи[15].

Варьирование языковых особенностей зависит от художественного или нехудожественного характера дневников. Художественные дневники используются писателями и поэтами как инструмент авторского замысла, например, «Дневник лишнего человека» И.С. Тургенева, «Дневник чумного доктора» Д. Дефо и др. Нехудожественные языки имеют реального автора и описывают невыдуманную действительность.

Дневниковые записи проекта «Прожито», рассмотренные в исследовании, в большинстве относятся к нехудожественным дневникам. Рассмотрим отличительные языковые черты этого типа подробнее. «Для личных, интимных, не рассчитанных на публику дневников, как правило, не характерно наличие художественных и композиционных приемов, направленных на раскрытие образов системы персонажей и на формирование линии конфликта»[16], – вследствие чего записи содержат больше простых предложений с минимальным количеством различного рода тропов, фигур речи. Также стоит заметить, что записи в большинстве ретроспективны, что делает использование форм прошедшего времени более частотным, чем

[13] Зализняк А. Дневник: к определению жанра // Новое литературное обозрение. 2010. Т. 106. С. 73.

[14] Богданова Е. В. Языковые особенности жанра дневника // Филологические науки. Вопросы теории и практики. 2008. №. 1-1. С. 29.

[15] Elspaß S. The use of private letters and diaries in sociolinguistic investigation // The handbook of historical sociolinguistics. 2012. С. 158.

[16] Богданова Е. В. Языковые особенности жанра дневника // Филологические науки. Вопросы теории и практики. 2008. №. 1-1. С. 29.

ориентированность на формы настоящего и будущего времени. Другой яркой особенностью является использование имен собственных и топонимов в записях. Авторы дневников часто упоминают в своих записях родственников, коллег, друзей, персонажей из просмотренных фильмов и прочитанных книг, артистов и т.д. Названия географических объектов важны для фиксаций мест, где происходило какое-либо события или для сохранения впечатлений о месте. Такая характеристика может оказаться ключевой при построении формальных моделей, предоставляя информацию о частоте использования онимов, позволяющую с большей вероятностью классифицировать разные записи. В дневниках также часто используются лексический повтор и фразеологические обороты, что придает тексту естественность разговорной речи.

Таким образом, язык дневника отражает индивидуальный стиль автора, включая его уникальные стилистические приемы, такие как эллипсы, опущения, использование просторечий, что создает искреннюю и реалистичную атмосферу повествования. Стоит добавить, что многие из приведенных характеристик доступны для автоматизации обработки языка в виде дополнительных модулей в целях усовершенствования качества работы моделей классического машинного и глубинного обучения.

### 1.3.1 Характеристика корпуса дневниковых записей «Прожито»

Проект «Прожито» является уникальным онлайн-архивом дневниковых записей, собранных из различных источников, включая музеи, частные коллекции и личные архивы с помощью сообщества волонтеров и студентов гуманитарных специальностей, проходящих практику в центре. Проект является самой известной российской инициативой по созданию открытых электронных баз данных эго-документов[17]. «Прожито» позволяет сохранить и изучить уникальные истории обычных людей, запечатленные в их дневниках.

[17] Лутошкина В. В. и др. Открытый электронный архив эго-документов "Прожито": сохранение личных историй. 2023. С. 92-97.

Сборка корпуса дневниковых записей проекта представляет собой ценный ресурс для исследований в области лингвистики, истории, социологии, психологии, литературы. Особенностью проекта является не только сохранение текстов, но и разработка инструментов для расширенного поиска по корпусу дневников (возможен поиск по дате написания, ключевым словам, тегам, имени, фильтрация по языку документа), что значительно облегчает работу исследователей. Использование материалов проекта дает возможность работать с большим корпусом текстов одного жанра и проводить автоматическую обработку социолингвистических атрибутов: «Для лингвистов тексты «Прожито» ценны тем, что они представляют собой особый регистр, по дискурсивной структуре близкий к устной речи, а наличие информации о каждом авторе и датировки записей позволяет использовать их для социолингвистических и диахронических исследований»[18].

Дневники содержат повседневные записи о жизни более 9 тысяч авторов по состоянию на март 2023 года. Общий объем корпуса – более 626 тысяч подневных записей XVIII–XXI веков. Корпус мультиязычный: он включает в себя русский, украинский, белорусский и казахский языковые разделы. Изначально фокус проекта был на изучении явления «блокадного дневника», который составляет значительную часть коллекции. Исследователи стремились понять различия между дневниками, веденными в мирное и военное время, а также выяснить значение личных записей для их авторов. Позже проект был расширен по тематике и временным рамкам – помимо дневников блокады, в коллекцию добавились записи от представителей советской молодежи, крестьянства, сотрудников дипломатических миссий и других групп. Сейчас собранные тексты отличаются разнообразием размеров, стилей написания и тематик.

[18] Мельниченко М. А., Тышкевич Н. Б. "Прожито" от рукописи до корпуса: сбор, разметка, анализ дневниковых текстов // Цифровая гуманитаристика: ресурсы, методы, исследования. 2017. С. 137.

Тексты, собранные в корпусе проекта «Прожито», являются достоверными и подлинными документами, что отличает их от текстов, например, полученных из социальных сетей или блогов-платформ, где пользователи в целях сохранения анонимности могут изменять или скрывать свои данные. В проекте «Прожито» под подлинностью дневника имеют в виду, что: «дневник был создан именно тем человеком, которому его приписывают; дневник был веден диахронически, т. е. в моменты описываемых событий; текст дневника не был искажен или изменен»[19].

Использование подлинных и достоверных текстов для построения модели классификации обладает следующими преимуществами:

- **Доверие к данным:** подлинные тексты предоставляют более достоверную информацию, поскольку они обычно несут личный характер и имеют меньше склонности к манипуляциям или искажениям, чем тексты, например, из социальных сетей, где присутствует больше возможностей для фальсификации и манипуляции информацией.
- **Обеспечение надежности результатов:** подтвержденные метаданные при обучении модели с «учителем», то есть со знанием меток классов текстов, могут обеспечить более надежные результаты в анализе и классификации информации и дальнейшем использовании моделей на новых данных.

**Выводы к главе 1**

В данной главе проведен обзор введения в социолингвистику и ее роли в анализе текстов. Социолингвистика как мультидисциплинарная научная дисциплина имеет различные интерпретации и определения, отражающие ее значение в изучении языковых вариаций, идентичностей, политики и

[19] Прожито. О корпусе. // Сайт проекта центра. [Б. м.]. URL: https://prozhito.org/page/corpus/ (дата обращения: 16.01.2024).

стереотипов. Важность социолингвистики заключается в ее способности объединить лингвистические и социологические подходы для понимания языка как социокультурного явления, что предоставляет инструменты для интерпретации социальных различий в письменной форме.

Социолингвистический портрет является лингвистическим явлением, изучающим взаимосвязь между языком и обществом. Изучение языковых вариаций и изменений является ключевым аспектом социолингвистического портрета, поскольку язык постоянно эволюционирует под влиянием потребностей и предпочтений его пользователей. Социолингвистический портрет выступает в роли методологического инструмента в анализе языковых практик и поведения групп и сообществ, находя применение в академическом контексте.

Дневники представляют собой ценный источник данных в различных областях исследования. Они содержат записи о событиях из первых рук, что делает их важными историческими свидетельствами. Дневники также могут быть использованы для реконструкции повседневной жизни прошлых эпох, так как они часто содержат описание ежедневных событий и взаимодействий. Изучение дневников предоставляет ценные сведения о языковом использовании в неофициальных контекстах. Основное отличие дневниковых записей заключается в их способности представить язык в его естественном и необработанном виде. Нехудожественные дневники, как те, которые рассматривались в данном исследовании, обладают определенными языковыми чертами. Они часто содержат простые предложения с минимальным использованием тропов и фигур речи. Записи также часто являются ретроспективными, содержат онимы. Лексический повтор и фразеологические обороты придают тексту естественность разговорной речи. В целом, дневники представляют собой ценный источник информации, который может быть использован для понимания и анализа различных аспектов языка.

Таким образом, представленный обзор подчеркивает важность социолингвистики для анализа текстов и актуальность использования дневниковых записей в научных исследованиях.

## 2 Классификация текстов с помощью методов машинного и глубинного обучения

Машинное обучение и глубинное обучение являются двумя подходами к искусственному интеллекту. Глубинное обучение является подразделом машинного обучения, которое использует искусственные нейронные сети для извлечения иерархических представлений данных. Часто используют общий термин «машинное обучение» для обозначения двух подходов, мы в большинстве контекстов разделяем их на «классические методы машинного обучения» и «методы глубинного обучения».

Предсказание скрытого признака по тексту с помощью машинного обучения является частью задачи классификации. Классификация текста с помощью нейросетей и улучшенных алгоритмов машинного обучения является одним из наиболее перспективных и эффективных подходов в области обработки естественного языка[20]. Модели искусственного интеллекта обладают способностью автоматически извлекать сложные признаки из текстовых данных, что позволяет эффективно разделять и относить тексты к определенным классам на основе их содержания.

Первый этап классификации текста – предварительная обработка. Она проводится для подготовки текста к последующей обработке и для удаления «шума» из входного текста. Это может включать очистку текста от лишних символов, токенизацию. Шаги автоматической предобработки текста помогают улучшить качество анализа текстовых данных, уменьшить их размерность. Выделение признаков или векторизация текста позволяет представить текст в виде числового вектора.

Этап классификации включает создание архитектуры модели. На этом этапе определяется структура и параметры модели. Для нейронных сетей это включает выбор типа слоев (например, рекуррентные, сверточные или

[20] Li H. Deep learning for natural language processing: advantages and challenges // National Science Review. 2018. Т. 5. №. 1. P. 24-26.

полносвязные слои), количество слоев, количество нейронов в каждом слое, функции активации и другие гиперпараметры. Обучение модели состоит в настройке параметров на основе обучающей выборки. Во время обучения нейросеть принимает входные данные, проходит через слои и вычисляет прогнозируемые значения. Затем, сравнивая прогнозы с правильными ответами, вычисляется ошибка и с помощью метода обратного распространения ошибки корректируются веса и параметры нейросети. После завершения обучения модели ее эффективность оценивается на тестовой выборке.

Для успешной классификации текста необходимо правильно выбрать и подготовить обучающую выборку, а также определить оптимальные гиперпараметры модели. Также важным фактором является размер обучающей выборки, так как недостаток данных может привести к переобучению модели.

В этой главе будут рассмотрены необходимые теоретические основы для создания сильных алгоритмов машинного и глубинного обучения в задачи бинарной и мультиклассовой классификации текстов, методы обработки текстов и способы оценки моделей.

## 2.1 Бинарная и мультиклассовая классификация текстов

Классификация в машинном обучении в целом относится к процессу присвоения объектам (например, изображениям, звукам, данным) одной или нескольких категорий на основе их признаков. Это может включать в себя задачи, такие как определение категории изображения, выявление спама в электронной почте или диагностика заболеваний на основе медицинских данных.

Классификация текстов – это конкретный случай классификации, где объектами являются текстовые документы, а признаками могут быть слова, фразы или другие структуры текста. Дополнительной характеристикой

является специфика в методах обработки данных – для получения признакового пространства используются токенизация и векторизация.

Бинарная и мультиклассовая классификация являются основными задачами машинного обучения, где цель состоит в том, чтобы отнести объекты к определенным классам на основе их признаков. В бинарной классификации каждый объект должен быть отнесен к одному из двух классов. Представим, что у нас есть обучающая выборка $\{(x_i, y_i), (x_2,\ y_2), \dots, (x_n, y_n)\}$, где $x_i$ – вектор признаков объекта, а $y_i$ – его метка класса (0 или 1). Задача бинарной классификации заключается в построении модели $f(x)$, которая по входному вектору признаков $x$ предсказывает метку класса $y$.

В случае мультиклассовой классификации объекты могут принадлежать к одному из нескольких классов, то есть $y_i$ – метка класса $(1, 2, \dots, K)$, где $K$ – количество классов. Задача заключается в построении модели, способной различать эти классы. При этом модель должна предсказывать вероятности принадлежности объекта к каждому классу.

Одним из подходов к мультиклассовой классификации является метод «one-vs-all» (один против всех). Для каждого класса строится $K$ бинарных классификаторов, каждый из которых отличает один класс от всех остальных. При предсказании метки класса объекта выбирается класс с наивысшей предсказанной вероятностью. В оппозиции существует класс «one-vs-one» (один против всех): строится $\frac{K(K-1)}{2}$ бинарных классификаторов, каждый из которых отличает пару классов. Предсказание принимается на основе голосования по всем классификаторам[21].

Математически мультиклассовая классификация может быть представлена как задача оптимизации функции потерь, учитывающей предсказанные вероятности для каждого класса. Например, для логистической

[21] Шалев-Шварц Ш., Бен-Давид Ш. Идеи машинного обучения: от теории к алгоритмам / пер. с англ. А. А. Слинкина. М. : ДМК Пресс, 2019. С. 220-222.

регрессии функция потерь может быть кросс-энтропия между предсказанными и истинными вероятностями классов.

В машинном и глубинном обучении настройка бинарной классификации требует определения двух классов, к которым должны относиться данные, и использования алгоритма, способного предсказать принадлежность к одному из этих классов. Для мультиклассовой классификации используются алгоритмы типа мультиклассовой логистической регрессии или глубокие нейросети с функцией активации softmax на последнем слое, который преобразует выходные данные в распределение вероятностей по классам. Это позволяет интерпретировать выходы сети как вероятности принадлежности к каждому из классов.

## 2.2 Методы предварительной обработки и векторизации текстов

В области обработки естественного языка (Natural Language Processing, NLP) предварительная обработка и векторизация текстов являются важными этапами для эффективного анализа и понимания текстовых данных. Эти методы позволяют преобразовать неструктурированный текст в структурированный формат, который может быть использован для различных задач, таких как классификация текстов, извлечение информации и машинный перевод. Эти методы занимают большую часть от всего процесса классификации (Рисунок 1) и отличают работу моделей машинного обучения на текстовом материале.


Рисунок 1 – Процесс классификации текста в глубинном обучении

Одним из ключевых методов предварительной обработки текстов является токенизация. При процессе токенизации текст делится на отдельные слова или токены, которые являются базовыми единицами для анализа. Это может быть достигнуто путем разделения текста по пробелам или использования более сложных алгоритмов, учитывающих особенности языка.

Следующий важный метод предварительной обработки текстов – удаление стоп-слов – распространенных слов, которые не несут смысловой нагрузки и могут быть исключены из анализа. Примерами таких слов могут быть предлоги, союзы, междометия, частицы. Удаление стоп-слов позволяет сократить размер словаря и улучшить качество анализа. Также это напрямую связано с редукцией признаков, описанной в статье А.А. Казанцева, М.В. Прохорова и П.С. Худяковой: «В типичном подходе для классификации текстовых статей частота слов используется в качестве основной части вектора признаков. Это обычно приводит к созданию векторов признаков с размерностью порядка десятков тысяч измерений. Вычислительная сложность любых операций с такими векторами признаков будет пропорциональна размеру вектора признаков, поэтому любые методы, которые уменьшают размер вектора признаков, не оказывая существенного влияния на производительность классификации, очень приветствуются в любом практическом применении»[22].

Дополнительно удаляются лишние символы: числа, знаки препинания, отдельного стоящие буквы и т.д. Также токены проходят через процесс лемматизации – приведения слов к начальной форме (например, *писал –> писать, красивому –> красивый*).

После предварительной обработки текста необходимо преобразовать его в числовую форму, чтобы его можно было использовать в алгоритмах

[22] Казанцев А. А., Прохоров М. В., Худякова П. С. Обзор подходов к классификации текстов актуальными методами // Экономика и качество систем связи. 2021. №. 1 (19). С. 62.

машинного обучения. Для этого используются методы векторизации текстов. Рассмотри самые распространенные из них.

Метод мешка слов (Bag-of-Words) заключается в следующем: каждому слову из словаря присваивается уникальный индекс, а затем текст представляется в виде вектора, где каждая компонента соответствует количеству вхождений соответствующего слова в тексте[23]. Этот метод позволяет сохранить информацию о наличии или отсутствии слов в тексте, но не учитывает порядок слов, что может привести к потере контекста и сути текста. Стоит заметить, что при таком подходе неэффективно используется память и вычислительные ресурсы.

Допустим, у нас есть следующие три текста:

1. Кошка сидит на крыше;

2. Собака сидит в доме;

3. Крыша мокрая после дождя.

Словарь (слова и их индексы): *{кошка: 1, сидит: 2, на: 3, крыше: 4, собака: 5, в: 6, доме: 7, крыша: 8, мокрая: 9, после: 10, дождя: 11}*. Тогда векторизация будет выглядеть как в таблице 1.

Таблица 1 – Пример векторизации текста с помощью Bag of Words

| **Текст** | кошка | сидит | на | крыше | собака | в | доме | крыша | мокрая | после | дождя |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Следующий метод к рассмотрению – TF-IDF (Term Frequency-Inverse Document Frequency). Он работает таким образом: каждому слову присваивается значение, основанное на частоте его встречаемости в тексте (TF) (1) и обратной частоте его встречаемости во всем корпусе текстов (IDF) (2). Этот подход учитывает не только наличие слов, но и их значимость для

[23] Hobson L., Dyshel M., Hapke H. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python. 2019. P. 70-73.

конкретного текста. Из недостатков – требуется предварительная обработка корпуса текстов для расчета IDF.

$$tf_{xy} = \frac{n_{xy}}{\Sigma_k n_{xy}}, \quad (1)$$

где $n_{xy}$ – количество слов $x$ в документе $y$.

$$w_{xy} = tf_{xy} * \log\left(\frac{N}{df_x}\right), \quad (2)$$

где $tf_{xy}$ – частота встречаемости слова $x$ в документе $y$;

$df_x$ – число документов, содержащих слово $x$;

$N$ – общее число документов.

Рассчитаем для примера TF-IDF для слова «*сидит*» в первом тексте («*Кошка сидит на крыше*»): $(1/4) * log(3/2) = 0.25 * 0.176 \approx 0.044$.

Плотные векторные представления слов (Word Embeddings): метод представления слов в виде плотных векторов в многомерном пространстве с возможностью не хранить информацию о нулях. Эти векторы учитывают семантические отношения между словами, что позволяет модели улавливать смысловые связи между ними[24]. Слова, близкие по смыслу, располагаются рядом, что улучшает понимание контекста и отношений между словами. Но такой метод требует больших объемов данных для обучения. Обучение плотных векторных представлений может происходить в составе нейронной сети или в составе дистрибутивно-семантической модели (Word2Vec, GloVe, FastText), для которой не требуется размеченный набор данных.

### 2.3 Основы машинного обучения для классификации текстов

Машинное обучение представляет собой раздел искусственного интеллекта, который фокусируется на разработке алгоритмов и моделей, способных извлекать паттерны и закономерности из данных с целью принятия

[24] Jurafsky D., Martin J. H. Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. New Jersey, 2008. P. 121-126.

решений без явного программирования. В области классификации текстов модели машинного обучения используются для автоматической категоризации текстовых документов по заранее определенным классам или категориям. Различают обучение с учителем (Supervised Learning) и без учителя (Unsupervised Learning). Обучение с учителем в задачах классификации текстов предполагает наличие размеченного обучающего набора данных, где каждый текст имеет соответствующую метку класса. Модели обучения с учителем, такие как логистическая регрессия или наивный Байесовский классификатор, обучаются на этих данных для того, чтобы впоследствии классифицировать новые тексты. С другой стороны, обучение без учителя в задачах классификации текстов не требует размеченных данных. Здесь модели, такие как кластеризация или методы понижения размерности, используются для выявления внутренних структур и паттернов в текстовых данных без предварительного знания о категориях.

Рассмотрим наиболее часто используемые модели двух типов, которые использовались для проведения экспериментов в целях предсказать скрытые атрибуты авторов по тексту.

### 2.3.1 Модели обучения с учителем

Логистическая регрессия является одним из самых простых и широко используемых методов для бинарной классификации. Ее цель состоит в прогнозировании вероятности принадлежности объекта к одному из двух классов. Основное предположение логистической регрессии заключается в том, что логарифм отношения вероятностей (логарифм odds) линейно зависит от независимых переменных.

В основе модели лежит сигмоидная функция (Рисунок 2), которая преобразует линейную комбинацию входных переменных в прогнозируемую вероятность[25]. Математически это выражается следующим образом:

[25] Yaser S. et al. Learning from Data: A Short Course. AMLBook, 2012. P. 90.

$$P(y = 1|X) = \frac{1}{1 + e^{b_0 + \Sigma b_i}}, \quad (3)$$

где $P(y = 1|X)$ – вероятность того, что $X$ принадлежит классу 1;

$b_0$ – свободный член (интерсепт);

$b_i$ – коэффициенты модели.

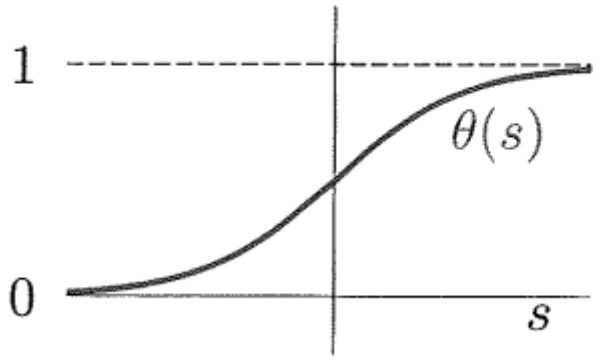


Рисунок 2 – Сигмоидная функция

Основное преимущество логистической регрессии заключается в простоте интерпретации – коэффициенты модели легко интерпретируются как влияние независимых переменных на вероятность события. Также логистическая регрессия относительно быстро обучается на больших наборах данных и требует малых вычислительных ресурсов, имеет относительно простую программную реализацию алгоритма[26]. Из недостатков можно назвать допущение линейности – логистическая регрессия предполагает линейную зависимость между независимыми переменными и логарифмом вероятностей, что может не всегда соответствовать действительности. Более того, модель чувствительна к аномальным значениям в данных, что может ухудшать ее производительность.

Другим распространенным методом является наивный байесовский классификатор, который основывается на принципах байесовской статистики и представляет собой простой, но мощный метод предсказания категории

[26] Батура Т. В. Методы автоматической классификации текстов // Программные продукты и системы. 2017. Т. 30. № 1. С. 85-99.

текста. Основой классификатора служит теорема Байеса, которая выражается следующей формулой:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}, \quad (4)$$

где $P(A|B)$ – апостериорная вероятность класса $A$ при условии признака $B$;

$P(B|A)$ – вероятность признака $B$ при условии $A$;

$P(A)$ – априорная вероятность класса;

$P(B)$ – вероятность признака.

В контексте классификации текстов, признаки часто представляют собой слова или фразы из текста, а классы – категории, к которым эти тексты нужно отнести. Модель вычисляет вероятность каждого класса на основе встречаемости слов в тексте, предполагая независимость признаков внутри классов.

Наивный байесовский классификатор особенно эффективен при работе с текстами благодаря своей способности обрабатывать большие объемы данных, и давать результаты в режиме реального времени. С другой стороны, нехватка данных может привести к потере точности, поскольку модель основана на предположении, что любые два признака независимы от выходного класса[27].

### 2.3.2 Модели обучения без учителя

Обучение без учителя является ключевым методом в машинном обучении, позволяющим анализировать и кластеризовать неструктурированные данные без предварительно заданных меток. Кластеризация – это процесс разбиения набора объектов на подгруппы, так что

[27] Dogra V. et al. A complete process of text classification system using state-of-the-art NLP models // Computational Intelligence and Neuroscience. 2022. Т. 2022. P. 19.

объекты в одной группе (кластере) более похожи друг на друга, чем на объекты в других группах.

Среди самых популярных алгоритмов кластеризации выделяют:

- K-средних (K-means), где K обозначает количество кластеров, заданное априори;
- Иерархическую кластеризацию, организующую данные в древовидную структуру;
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise), который выделяет кластеры на основе плотности объектов.

Остановимся подробнее на алгоритме K-means[28], который будет использоваться далее в эксперименте по предсказанию уровня образования авторов текста. Алгоритм K-means минимизирует сумму квадратов расстояний между точками и соответствующими центрами их кластеров. Формула для вычисления этих расстояний:

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 , \quad (5)$$

где $k$ – количество кластеров;

$C_i$ – i-й кластер;

$x$ – точка внутри кластера;

$\mu_i$ – центроид i-го кластера.

Алгоритм стремится минимизировать суммарное внутрикластерное расстояние. В начале работы алгоритма инициализируются $K$ начальных центров кластеров (центроидов), что может быть сделано случайным образом или с помощью более продвинутых методов, таких как K-means++. После для

[28] Ahmed M., Seraj R., Islam S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation // Electronics. 2020. Т. 9. №. 8.

каждого объекта в данных вычисляется расстояние до каждого центроида, и объект присваивается кластеру с ближайшим центроидом:

$$C_i = \{x: \|x - \mu_i\| \leq \|x - \mu_i\|, \forall_j, 1 \leq j \leq K\}, \quad (6)$$

Где:

$C_i$ – набор точек, наиболее близких к центроиду $\mu_i$.

Центроиды кластеров обновляются путем вычисления среднего арифметического всех точек, присвоенных каждому кластеру:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x, \quad (7)$$

Этот шаг влечет пересчет центров масс для каждого кластера. Последние два шага повторяются до тех пор, пока присвоение кластеров не перестанет изменяться, или до достижения заданного числа итераций, что означает устойчивость кластеров. Алгоритм считается сходящимся, когда изменения в центроидах или принадлежности элементов к кластерам становятся минимальными или достигают определенного порога изменений.

### 2.4 Основы глубинного обучения для классификации текстов

Традиционные модели машинного обучения, рассмотренные выше, долгое время были основой для систем обработки текста. Однако с появлением глубинного / глубокого обучения (Deep Learning, DL) возможности NLP значительно расширились. Глубокое обучение является одной из самых мощных и перспективных технологий в области машинного обучения. В частности, во многих исследованиях по классификации текстов подтверждается преимущество моделей глубокого обучения[29] [30]. Благодаря

[29] Kamath C. N., Bukhari S. S., Dengel A. A Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification. Proceedings of the ACM Symposium on Document Engineering 2018. DocEng'18.

[30] Suneera C. M., Prakash J. Performance analysis of machine learning and deep learning models for text classification // 2020 IEEE 17th India council international conference (INDICON). IEEE, 2020. P. 1-6.

улучшению технологий и доступа к большим массивам данных, глубокое обучение становится доминирующей технологией в NLP.

Глубинные нейронные сети предлагают значительные преимущества в понимании контекста и семантики текста благодаря использованию сверточных и рекуррентных нейронных сетей, которые будут подробнее рассмотрены далее. Также модели DL отлично справляются с задачами, где требуется анализировать большие объемы данных благодаря их способности извлекать ключевые особенности из данных автоматически. Трансферное обучение и модели, например, BERT или GPT, позволяют разрабатывать мощные системы, используя предварительно обученные модели, которые можно настраивать для конкретных задач и т.д.

Глубинное обучение позволяет создавать модели способные автоматически извлекать высокоуровневые признаки из сложных данных, к которым можно отнести тексты. Главный компонент, способствующий качественной работе с текстовыми данными, – слой погружения. Гольдберг определяет этот термин так: «отображение дискретных символов на непрерывные векторы в пространстве сравнительно небольшой размерности»[31].

Нейронная сеть – это сложная дифференцируемая функция, задающая отображение из исходного признакового пространства в пространство ответов, все параметры которой могут настраиваться одновременно и взаимосвязанно (то есть сеть может обучаться end-to-end). В частном случае представляет собой последовательность (дифференцируемых) параметрических преобразований. Слои могут быть двух видов: линейные слои (Dense Layer), которые преобразовывают входящие данные путем преобразования $d$-мерных векторов в $k$-мерные, и слои с функцией активации – нелинейным преобразованием, поэлементно применяемым к пришедшим на

[31] Гольдберг Й. Нейросетевые методы в обработке естественного языка / пер. с англ. А. А. Слинкина. М. : ДМК Пресс, 2019. С. 22.

вход данным. Существует множество различных функций активаций, самые популярные из них – ReLU (Rectified Linear Unit), Sigmoid, Tanh и др.[32] Краткое описание для каждой из перечисленных функций активации (Рисунок 3) можно представить следующим образом:

- ReLU: все отрицательные значения входа обнуляются и положительные значения передаются без изменений. Преимущества: простота вычисления, отсутствие проблемы исчезающего градиента, ускорение сходимости обучения. Недостатки: может вызвать проблему «мертвых нейронов» (нейроны, которые не активируются), не центрирована относительно нуля.
- Sigmoid: преобразует любое входное значение в диапазоне от 0 до 1. Преимущества: простота интерпретации как вероятности, гладкая производная. Недостатки: проблема затухающего градиента при глубоких сетях, не центрирована относительно нуля.
- Tanh (гиперболический тангенс): входное значение преобразуется в диапазоне от -1 до 1. Эта функция подобна сигмоиде, но центрирована относительно нуля. Преимущества: подходит для задач, где данные должны быть нормализованы. Недостатки: может вызвать проблему затухающего градиента при глубоких сетях.

[32] Nwankpa C. et al. Activation functions: Comparison of trends in practice and research for deep learning // arXiv preprint arXiv:1811.03378. 2018.

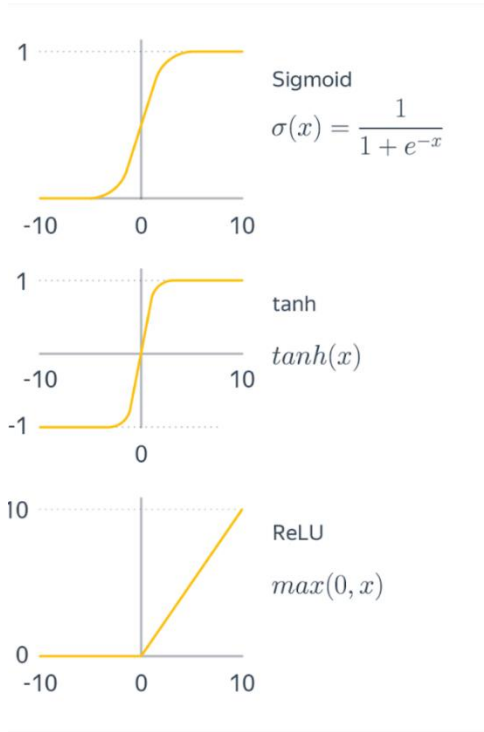


Рисунок 3 – Сравнение функций активации в нейронных сетях

Нейросеть можно представить как вычислительный граф, где вершинам соответствуют преобразования. Нейросеть, в которой есть только линейные слои и различные функции активации, называю **полносвязной** (Fully Connected) **нейронной сетью** или **многослойным перцептроном** (Multilayer Perceptron, MLP) (Рисунок 4). Многослойный перцептрон (8) – простейшая архитектура нейронной сети, в которой каждый слой нейронов связан со всеми нейронами с предыдущего слоя.

$$y = f(W_n \cdot f(W_n - 1 \cdot f(\dots f(W_1 \cdot x + b_1) \dots) + b_n - 1) + b_n), \quad (8)$$

где $x$ – входные данные;

$W_i$ – матрица весов для $i$-го слоя;

$b_i$ – вектор смещения для $i$-го слоя;

$f$ – функция активации;

$y$ – выходные данные.

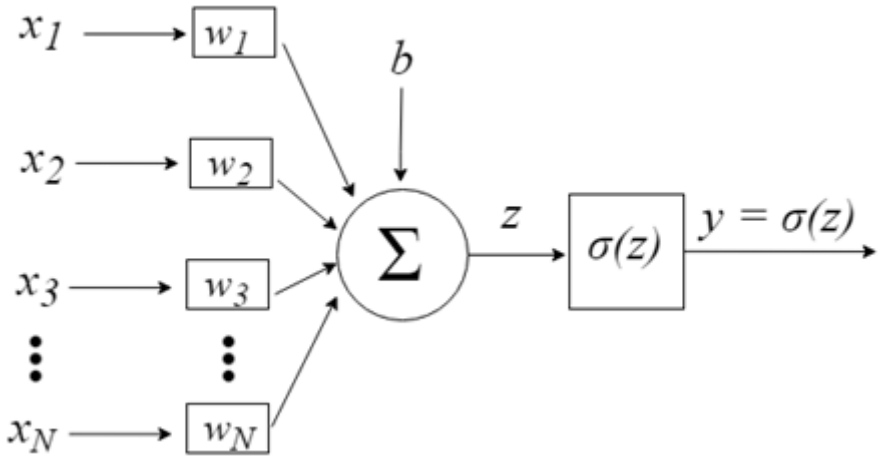


Рисунок 4 – Устройство многослойного перцептрона

Применение нейронной сети к данным (вычисление выхода по заданному входу) часто называют прямым проходом или **Forward Propagation** (forward pass). На этом этапе происходит преобразование исходного представления данных в целевое и последовательно строятся промежуточные (внутренние) представления данных – результаты применения слоев к предыдущим представлениям. То есть сначала вычисляются активации каждого нейрона в слое (9), после вычисляется взвешенная сумма входов для каждого нейрона в следующем слое (10). Таким образом повторяется для каждого слоя до достижения последнего слоя.

$$a^l = f(z^l), \tag{9}$$

где $a^l$ – активация в слое $l$; $z^l$ – взвешенная сумма входов для слоя $l$, $f$ – функция активации.

$$z^{l+1} = W^l \cdot a^l + b^l, \tag{10}$$

где $W^l$ – матрица весов между слоями $l$ и $l+1$, $b^l$ – вектор смещений для слоя $l+1$.

При обратном проходе или **Backward Propagation** (backward pass), информация (обычно об ошибке предсказания целевого представления) движется от финального представления (а чаще даже от функции потерь) к исходному через все преобразования. Сначала происходит вычисление градиента функции потерь по активациям последнего слоя (11), после

вычисляются градиенты функции потерь по взвешенным суммам последнего слоя (12) и градиенты функции потерь по матрице весов и вектору смещения последнего слоя (13.1-13.2). Далее рассчитывается распространение градиента обратно через предыдущие слои (14.1-14.2). На последнем шаге вычисляются градиенты функции потерь по матрице весов и вектору смещений каждого слоя (15.1-15.2).

$$\frac{\partial L}{\partial a^L}, \tag{11}$$

$$\frac{\partial L}{\partial z^L} = \frac{\partial L}{\partial a^L} \odot f'(z^L), \tag{12}$$

где $f'(z^L)$ – производная функции активации для последнего слоя.

$$\frac{\partial L}{\partial W^L} = \frac{\partial L}{\partial z^L} \cdot (a^{L-1})^T, \tag{13.1}$$

$$\frac{\partial L}{\partial b^L} = \frac{\partial L}{\partial z^L}, \tag{13.2}$$

$$\frac{\partial L}{\partial a^l} = (W^{l+1})^T \cdot \frac{\partial L}{\partial z^{l+1}}, \tag{14.1}$$

$$\frac{\partial L}{\partial z^l} = \frac{\partial L}{\partial a^l} \odot f'(z^l), \tag{14.2}$$

где $f'(z^l)$ – производная функции активации для слоя $l$.

$$\frac{\partial L}{\partial W^l} = \frac{\partial L}{\partial z^l} \cdot (a^{l-1})^T, \tag{15.1}$$

$$\frac{\partial L}{\partial b^l} = \frac{\partial L}{\partial z^l}, \tag{15.2}$$

Основные проблемы, которые могут возникать при обучении нейронных сетей – это переобучение и нестабильность процесса обучения. Для борьбы с этими проблемами используются методы нормализации и регуляризации. Эти методы не только способствуют улучшению производительности моделей, но и обеспечивают более быструю и стабильную сходимость обучения.

**Нормализация** – это процесс предварительной обработки данных, который помогает стандартизировать диапазон значений входных данных. Существует несколько методов нормализации: масштабирование по

минимаксу, при котором все числовые признаки масштабируются в диапазон от 0 до 1; стандартизация Z-оценки, преобразующая признаки таким образом, чтобы их средние значения равнялись 0, а стандартные отклонения – 1; нормализация пакета (Batch Normalization). Остановимся на последнем методе, так как он наиболее часто используется при создании нейросетей. Batch Normalization – это техника, которая нормализует входы каждого слоя внутри сети. Это повышает стабильность обучения и ускоряет сходимость[33]. Процесс можно описать следующими формулами (16.1-16.4).

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i\,, \tag{16.1}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2\,, \tag{16.2}$$

$$\hat{x}_i = \frac{x_i^1 - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{16.3}$$

$$yi = \gamma\hat{x}_i + \beta, \tag{16.4}$$

где $x_i$ – входы для мини-пакета;

$\mu_B$ и $\sigma_B^2$ – среднее и дисперсия пакета;

$\gamma$ и $\beta$ параметры масштабирования и сдвига, настраиваемые во время обучения.

**Регуляризация** – это методы, которые помогают снижать переобучение, ограничивая сложность модели. Основные методы включают L1-регуляризацию (Lasso) с добавлением штрафа, равному абсолютной величине коэффициентов (17), и L2-регуляризацию (Ridge) с учетом штрафа, равному квадрату величины коэффициентов (18).

$$L1 = Loss\ Function + \ \lambda\sum_{i=1}^{n} |w_i|\,, \tag{17}$$

[33] Huang L. et al. Normalization techniques in training dnns: Methodology, analysis and application // IEEE transactions on pattern analysis and machine intelligence. 2023. Т. 45. №. 8. P. 10173-10196.

$$L2 = Loss\ Function + \lambda \sum_{i=1}^{n} w_i^2, \quad (18)$$

где $\lambda$ – параметр регуляризации, а $w_i$ – веса модели.

Dropout является еще одним эффективным методом регуляризации, применяемым в нейронных сетях. Этот метод заключается в случайном исключении некоторых нейронов из процесса обучения во время каждой итерации, что помогает предотвратить переобучение, так как модель не может полагаться на любой конкретный нейрон.

Нормализация и регуляризация являются критически важными техниками в разработке и обучении моделей нейронных сетей. Эффективное применение этих методов требует понимания и опыта, а их взаимодействие должно быть также тщательно настроено, чтобы соответствовать конкретной архитектуре и данным.

Также важно оптимизировать модели для увеличения скорости и способности к обобщению. Эффективная оптимизация помогает достигать высокой точности при минимальных временных затратах и ресурсах. Основа большинства алгоритмов оптимизации в нейронных сетях – это градиентный спуск (19). Его суть заключается в минимизации функции потерь, которая оценивает разницу между предсказанным и истинным значениями.

$$\theta = \theta - \alpha \nabla J(\theta), \quad (19)$$

где $\theta$ – параметры модели, которые мы обновляем;

$\alpha$ – скорость обучения (learning rate), коэффициент, определяющий размер шага при обновлении параметров;

$J(\theta)$ – функция потерь, который минимизируем;

$\nabla J(\theta)$ – градиент функции потерь по параметрам $\theta$, указывающий направление наискорейшего убывания функции потерь.

Также существуют методы адаптивной оптимизации: AdaGrad, RMSProp, Adam. Последний метод, как указывают авторы[34], сочетает преимущества двух первых. Ключевой особенностью является расчет адаптивных скоростей обучения для каждого параметра. Adam также сохраняет экспоненциально затухающие средние прошлых градиентов и квадратов градиентов.

С точки зрения глубокого обучения, профилирование авторов включает в себя обучающие модели для прогнозирования одного или нескольких предопределенных атрибутов автора на основе его стиля письма. Обычно это предполагает использование архитектур нейронных сетей, таких как сверточные нейронные сети (CNNs) или рекуррентные нейронные сети (RNNs) для изучения сложных закономерностей в текстовых данных и извлечения признаков, которые имеют отношение к задаче прогнозирования. Например, в задаче гендерной классификации модель глубокого обучения будет обучена предсказывать пол автора на основе его стиля письма. Модель будет обучена на большом наборе данных текстовых образцов с известными гендерными метками (задача бинарной классификации) и научится распознавать шаблоны в текстовых данных, которые указывают на мужской или женский стиль письма. Аналогично, в задаче многоклассовой классификации по возрасту модель обучается предсказывать возраст автора на основе его стиля письма и находить закономерности в текстовых данных, которые связаны с различными возрастными группами и т.д.

### 2.4.1 Сверточные нейронные сети

Сверточная нейронная сеть (CNN) – класс нейронных сетей, специально разработанных для обработки и анализа структурированных данных, таких как изображения. Архитектура сверточной нейросети состоит из нескольких

[34] Kingma D. P., Ba J. Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. 2014.

типов слоев, каждый из которых выполняет определенную функцию в процесс обработки данных[35]:

1. Сверточные слои являются основным строительным блоком нейросети. В них происходит операция свертки, где фильтры (ядра) применяются к входным данным для извлечения признаков. Каждый фильтр выделяет определенные характеристики изображения или других данных. Затем результаты свертки объединяются для создания карт признаков.
2. Пулинговые слои, следующие после сверточных, уменьшают размерность карт признаков, сохраняя важные информационные характеристики. Популярные методы пулинга включают Max Pooling (операция выбора максимума из подвыборки) (Рисунок 5) и Average Pooling (операция выбора среднего из подвыборки).

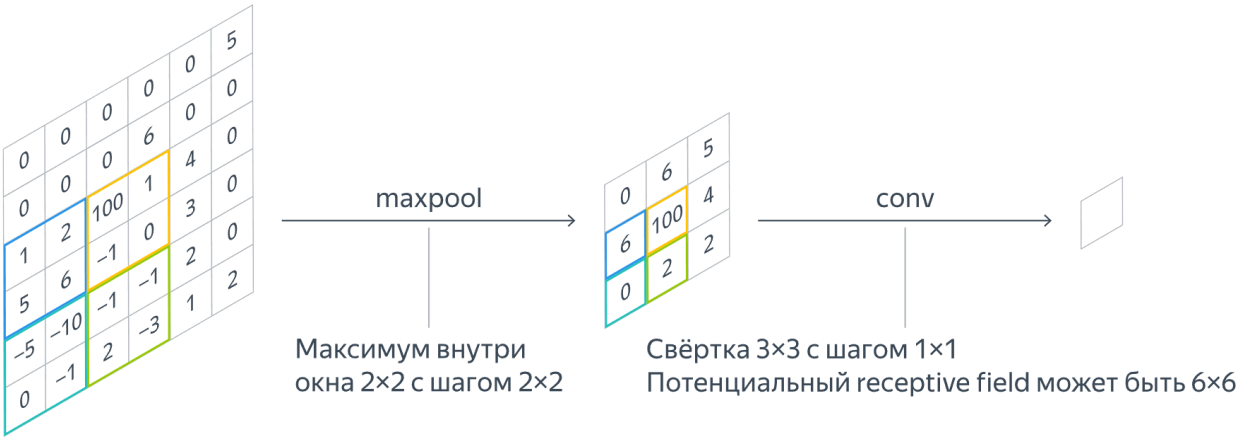


Рисунок 5 – Пример работы слоя Max Pooling

Можно использовать также другой способ уменьшения размера карт – Strided Convolution, при котором ядро свертки сдвигается на каждом шаге на некоторое большее единицы число пикселей.

3. Полносвязанные слои соединяют все признаки, извлеченные из предыдущих слоев, и используют их для классификации.

[35] Goodfellow I., Bengio Y., Courville A. Deep Learning. 2016. P. 330-360.

4. Нормализация (Normalization Layers) служит для стабилизации процесса обучения и ускорения сходимости модели.
5. Слои регуляризации (Regularization Layers) помогают предотвратить переобучение модели путем ограничения ее сложности. Примерами классических методов регуляризации являются Dropout и L1/L2-регуляризация. Отдельно для сверточных нейросетей можно использовать аугментацию данных, включающую сдвиги и повороты по осям матрицы, добавление случайного гауссовского шума и др. Также возможно использование Label Smoothing, при котором модель штрафуется за крайние значения предсказаний, то есть слишком уверенные, что помогает бороться с шумом в разметке данных.

Обратное распространение ошибки в сверточных нейронных сетях работает путем передачи ошибки от выходного слоя к входному через последовательность матричных операций, таких как свертка, пулинг и активация. Этот процесс позволяет сети корректировать веса (параметры) каждого слоя, чтобы минимизировать ошибку предсказания. В сверточных нейронных сетях обратное распространение ошибки учитывает пространственную структуру входных данных, что делает его эффективным для обработки изображений и других типов данных, где важно учитывать пространственные зависимости. Математический аспект обратного распространения ошибки включает в себя использование цепного правила для вычисления градиентов. Для сверточных слоев это также включает операцию обратной свертки (кросс-корреляции) для вычисления градиентов по фильтрам.

Хотя сверточные нейронные сети изначально были разработаны для обработки изображений, они также могут быть применены для обработки текстов. Для этого необходимо представить их в виде матрицы слов или

символов, которая затем подается на вход сети[36]. Часто в таких задачах используют **одномерные сверточные сети** (CNN1D) (Рисунок 6).

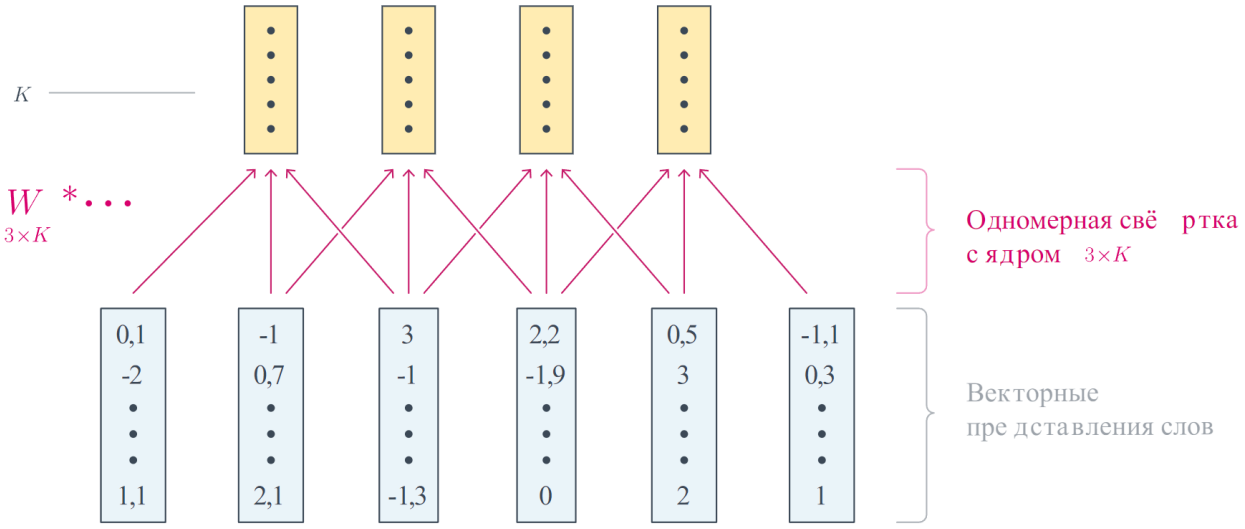


Рисунок 6 – Пример работы одномерной свертки

В отличие от классических сверточных сетей, которые работают с двумерными данными (изображениями), CNN1D используется для анализа последовательностей данных. Архитектура CNN1D включает в себя сверточные слои, пулинговые слои и полносвязанные слои, аналогично классическим CNNs, однако свертки в CNN1D перемещаются только в одном направлении по временной или пространственной оси данных, извлекая признаки из последовательности.

### 2.4.2 Рекуррентные нейронные сети

Рекуррентные нейронные сети (RNNs) представляют собой мощный класс глубоких нейронных сетей, способных обрабатывать последовательные данные с учетом контекста и зависимостей во времени. Их уникальная способность запоминать предыдущие состояния и использовать эту информацию для прогнозирования следующих шагов делает их особенно эффективными в задачах обработки текста, аудио, временных рядов и других

[36] Jacovi A., Shalom O.S., Goldberg Y. Understanding convolutional neural networks for text classification // arXiv preprint arXiv:1809.08037. 2018. 10 p.

данных, где важно учитывать последовательную природу информации. В этом смысле рекуррентные сети подобны людям, они также «читают» текст не моментально, а слово за словом в строго определенном порядке.

Алгоритм работы сети на рисунке 7 можно описать следующим образом: на каждом временном шаге $t$ RNN принимает вход $x_t$ и скрытое состояние $h_{t-1}$ с предыдущего шага; скрытое состояние на текущем шаге $h_t$ вычисляется как функция от входа $x_t$ и предыдущего скрытого состояния $h_{t-1}$:

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h), \quad (20)$$

где $W_{hx}$ – матрица весов для входа;

$W_{hh}$ – матрица весов для скрытого состояния;

$b_h$ – вектор смещения;

$f$ – функция активации, например, гиперболический тангенс или ReLU.

На выходе RNN может быть добавлен слой для прогнозирования или классификации (21).

$$y_t = g(W_{hy}h_t + b_y), \quad (21)$$

где:

$W_{hy}$ – матрица весов для выходного слоя;

$b_y$ – вектор смещения;

$g$ – функция активации для выходного слоя.

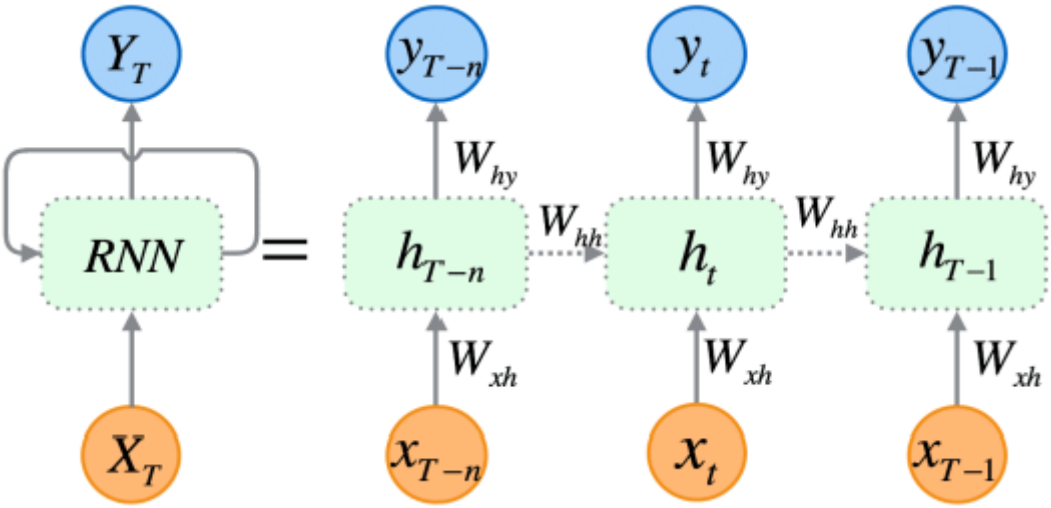


Рисунок 7 – Архитектура RNN

Таким образом, RNN позволяют учитывать контекст и последовательность данных, сохраняя информацию о предыдущих состояниях. Однако у классических RNNs есть проблемы с затуханием и взрывом градиентов – ситуация, при которой градиент, передаваемый от выходного слоя к входному, с каждым шагом уменьшается до нуля или становится очень маленьким. Это происходит из-за того, что в процессе обратного распространения ошибки градиент умножается на матрицу весов между двумя последовательными слоями. Если эта матрица имеет собственные значения, близкие к нулю, то градиент также будет очень маленьким. Эта проблема может привести к тому, что RNN не сможет обучаться на длинных последовательностях, так как градиент будет слишком маленьким, чтобы обновить веса входного слоя. В результате модель не сможет корректно запомнить долгосрочные зависимости в последовательности. Для решения этой проблемы были разработаны более продвинутые архитектуры, такие как LSTM (Long Short-Term Memory), BiRNNs (Bidirectional Recurrent Neural Networks) и GRU (Gated Recurrent Unit) (Рисунок 8), которые помогают сохранять и обрабатывать информацию на длинных временных интервалах.

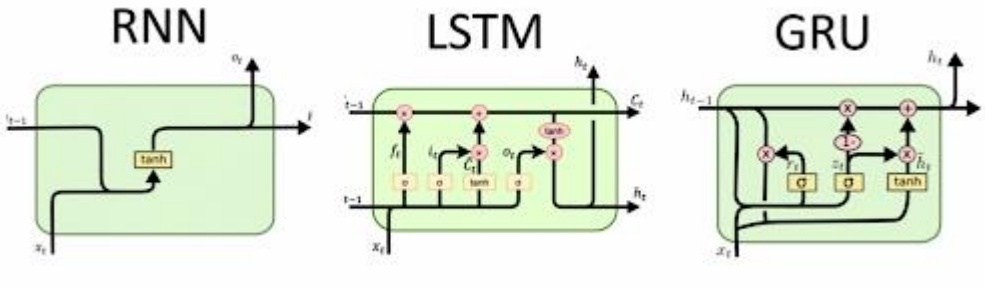


Рисунок 8 – Отличие разных архитектур рекуррентных нейросетей

Рассмотрим наиболее популярные архитектуры.

**LSTM** (Long Short-Term Memory) – это тип рекуррентной нейронной сети, который способен эффективно учитывать долгосрочные зависимости в последовательных данных. Процесс работы LSTM можно описать следующим образом:

1) На каждом временном шаге модели подается входной вектор данных;
2) Вычисление входного ворота (Input Gate): модель вычисляет, какую информацию из входных данных нужно сохранить;
3) Вычисление ворота забывания (Forget Gate): модель определяет, какую информацию из предыдущего состояния следует забыть;
4) Обновление клеточного состояния (Cell State): информация из входного ворота и предыдущего состояния клетки комбинируется для обновления текущего состояния клетки;
5) Вычисление выходного ворота (Output Gate): LSTM определяет, какую информацию из обновленного состояния клетки нужно передать на выход;
6) На основе выходного ворота и обновленного состояния клетки LSTM генерирует выходные данные и скрытое состояние, которые могут быть использованы на следующем временном шаге.

Этот процесс позволяет LSTM эффективно учитывать долгосрочные зависимости в последовательных данных и принимать решения о том, какую информацию сохранить и передать дальше[37].

**BiRNNs / BiLSTM** (Bidirectional LSTM) – тип рекуррентных сетей, которые могут обрабатывать последовательности данных в двух направлениях.

Они состоят из двух рекуррентных слоев: прямого и обратного. Прямой рекуррентный слой обрабатывает входные данные в прямом направлении, начиная с первого элемента последовательности и заканчивая последним. Обратный рекуррентный слой обрабатывает входные данные в обратном направлении, начиная с последнего элемента последовательности и заканчивая первым. Результаты обоих слоев объединяются для получения окончательного выхода.

[37] Goodfellow I., Bengio Y., Courville A. Deep Learning. 2016. P. 408-411.

Одним из преимуществ двунаправленных рекуррентных нейросетей является то, что они могут учитывать контекст как слева, так и справа от текущего элемента последовательности. Это позволяет им лучше понимать смысл текстов и улучшает качество предсказаний.

### 2.5 Оценка моделей классификации

Модели классификации, будучи одним из ключевых инструментов в различных прикладных областях, нуждаются в тщательной оценке их эффективности и точности.

Оценка производительности модели чаще всего определяется по ее точности (Accuracy) – метрики, которая определяется как отношение количества правильно предсказанных признаков ($Nc$) к общему количеству предсказанных признаков ($Nt$):

$$Accuracy = \frac{Nc}{Nt} \tag{22}$$

Также для оценки используется матрица ошибок (Рисунок 9), позволяющая оценить не только общую точность, но также специфичность и чувствительность модели для каждого класса. Она включает в себя значения истинных положительных (True Positives, TP), истинных отрицательных (True Negatives, TN), ложных положительных (False Positives, FP) и ложных отрицательных (False Negatives, FN) судейств.

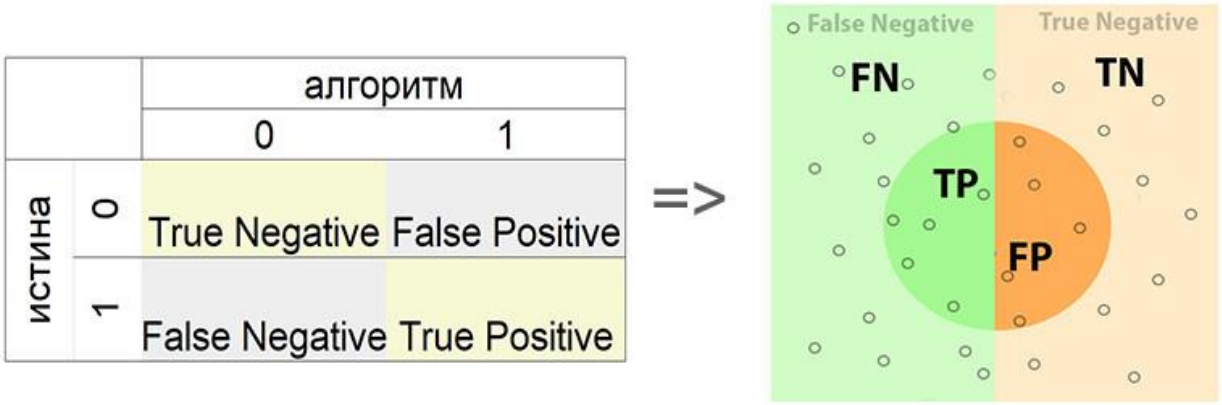


Рисунок 9 – Матрица ошибок

Полнота (Recall) показывает (23), какую долю объектов реального положительного класса модель смогла обнаружить:

$$Recall = \frac{TP}{TP + FN} \quad (23)$$

Точность (Precision) отражает (24), какая доля объектов, отнесенных моделью к положительному классу, действительно является положительными:

$$Precision = \frac{TP}{TP + FP} \quad (24)$$

F1-Score – гармоническое среднее полноты и точности (25). Оно полезно, когда необходимо учесть оба эти показателя одновременно:

$$F1 = 2 * \frac{Precision * Recall}{Precission + Recall} \quad (25)$$

Кривая ROC (Receiver Operating Characteristics) иллюстрирует отношение между чувствительностью и специфичностью для разных порогов классификации. Площадь под кривой ROC (AUC) оценивает, насколько хорошо модель может различать классы. Значение 1 означает идеальную классификацию.

Оценка моделей классификации следует широкому спектру методов, каждый из которых имеет свои преимущества и случаи предпочтительного применения[38]. Осознание особенностей каждого из методов позволяет более грамотно подходить к анализу результатов классификации и способствует разработке более точных и надежных моделей. Важно учитывать, что идеального метода оценки не существует: выбор метода должен опираться на специфику данных и поставленные задачи.

**Выводы к главе 2**

В главе были изложены различия между бинарной и мультиклассовой классификацией, ключевые техники предварительной обработки текста;

[38] Kowsari K. et al. Text Classification Algorithms: A Survey // arXiv e-prints. 2019. P. 45-47.

описаны основные модели машинного обучения, такие как логистическая регрессия и наивный байесовский классификатор для обучения с учителем, а также K-means для обучения без учителя; изложены принципы построения нейронных сетей, включая важность выбора активационных функций, методов нормализации, стратегий оптимизации, и эффективные архитектуры в задачах, связанных с пониманием контекста и последовательностей в тексте – одномерные сверточные и рекуррентные сети; описаны ключевые подходы и метрики для оценки эффективности моделей классификации текстов.

Бинарная классификация применяется в задачах с двумя классами, а мультиклассовая – в задачах с тремя и более классами. Классификация текстов в рамках первой задачи осуществляется при помощи размеченного корпуса с метками для каждого текста. При обучении моделей для классификации текстов важно определить классы данных в датасете, выбрать подходящий алгоритм, в случае нейронных сетей учесть функцию softmax с указанием количества классов на выходном слое, и правильно разделить данные для обучения и тестирования качества работы модели. Некоторые модели (как логистическая регрессия и наивный байесовский классификатор) лучше подходят для определенных типов текстов, в то время как методы кластеризации могут быть полезны для исследовательских задач, где неопределенные классы текстов необходимо группировать предварительно.

Специфика классификации текста как задачи машинного обучения заключается в необходимости превращать текстовые данные после очистки их от шума, токенизации и лемматизации в вещественные признаки. Существуют различные подходы к векторизации текстов, в главе подробно были рассмотрены одни из самых популярных: Bag of Words, TF-IDF, Word Embeddings. Важно заметить, что качество первичной предобработки и выбор метода векторизации могут значительно повлиять на точность классификации.

Отмечена растущая эффективность глубокого обучения в задачах классификации текстов, особенно с использованием CNN1D и рекуррентных

сетей (LSTM, BiLSTM), которые способны улавливать контекстуальные зависимости в данных. Главное отличие нейросетей от алгоритмов машинного обучения – способность автоматически извлекать и обобщать сложные признаки из данных.

Вводные данные о значении архитектуры нейросети: нормализации, регуляризации и оптимизации для улучшения результатов классификации текстов демонстрируют, что внедрение современных подходов в конфигурацию нейронных сетей (например, применение регуляризации для борьбы с переобучением и выбора эффективных оптимизаторов) являются решающими для достижения высоких результатов.

Метрики для оценки эффективности моделей классификации текстов, такие как точность, полнота и F1-мера помогают анализировать степень «успешности» модели в решении поставленных задач.

# 3 Описание экспериментов по классификации социолингвистических атрибутов

Код, написанный на этапах по сбору, предобработке корпуса, и эксперименты, проведенные с использованием моделей машинного и глубинного обучения, загружены в репозиторий на GitHub[39].

## 3.1 Сбор корпуса дневниковых записей

Дневниковые записи «Прожито» были выгружены командой проекта 08.01.2023 года по запросу для более детальной работы с текстами дневников. Выгрузка представила записи всех дневников корпуса «Прожито» не закрытого доступа, а также ограниченные сведения об авторах.

Работа велась с тремя файлами в формате csv – persons (авторы дневников), notes (записи в дневниках), diaries (дневники). Примеры данных см. в Приложении А. Таблица файла persons (Таблица А.1) включала в себя идентификатор автора, дату рождения, дату смерти, показатель, устанавливающий, если даты жизни установлены приблизительно, пол, наличие биографической справки в HTML. Файл diaries (Таблица А.2) содержал идентификатор дневника, идентификатор автора дневника и показатель, устанавливающий, опубликован ли дневник впервые в «Прожито». В большинстве случаев одной персоне соответствовал один дневник, но было несколько десятков случаев, когда одной персоне принадлежало несколько дневников. Файл notes (Таблица А.3) хранил идентификатор записи, идентификатор дневника, текст записи в HTML, дату записи, верхнюю дату записи в случае охватывания периода, отметку, если запись не датирована.

[39] Prediction-Sociolinguistic-Data-Based-on-the-Diaries-Texts-of-the-Prozhito-Project // Репозиторий с кодом для исследования [Б. м.]. URL: https://github.com/vlada-pv/Prediction-Sociolinguistic-Data-Based-on-the-Diaries-Texts-of-the-Prozhito-Project

Так как необходимые для обучения признаки находились в разных таблицах, была написана вспомогательная программа на языке Python (см. Приложение Б) с использованием модуля для работы с табличными данными Pandas, которая обходила все файлы в определенной последовательности и собирала в единую таблицу значения следующих столбцов: идентификатор записи, идентификатор дневника, текст записи, дата записи, пол автора, дата рождения. Отдельно был создан столбец со значениями возраста автора, в котором была сделана запись. Он вычислялся для каждой из строк путем вычитания даты рождения из даты написания.

Из итогового датасета дополнительно были удалены строки с пустыми значениями хотя бы по одному из признаков. Датасет notes_final, таким образом, содержал 785 548 строки и 7 колонок: id, diary, text, date, sex, birth, age (Таблица 2).

Таблица 2 – Часть итогового датасета notes_final

| id | diary | text | date | sex | birth | age |
|---|---|---|---|---|---|---|
| 693 | 3 | <p>Сейчас бьет на башенных часах 12. Полночь. ... | 1932-09-28 | 1.0 | 1890-05-25 | 42.0 |
| 694 | 3 | <p>Читал Шаляпина — воспоминания. Оказывается,... | 1932-10-06 | 1.0 | 1890-05-25 | 42.0 |
| 695 | 3 | <p>При социализме заводы суть храмы и центры ч... | 1932-10-20 | 1.0 | 1890-05-25 | 42.0 |
| 696 | 3 | <p>Завтра с утра — в Москву. Что-то меня там ж... | 1932-10-30 | 1.0 | 1890-05-25 | 42.0 |

### 3.2 Предобработка текстов

Для обеспечения качественной подготовки текстовых данных для последующей работы с алгоритмами машинного и глубинного обучения их необходимо подвергнуть комплексной предобработке. Были написаны функции (см. Приложение В) на языке Python для автоматического осуществления предобработки. Первый этап включал следующие шаги:

- **Удаление HTML тегов:** они не несут в себе полезной информации для анализа текста. Для их удаления использовалась библиотека BeautifulSoup[40];
- **Удаление специальных символов и чисел**: они не релевантны для задач обработки естественного языка, зависящих от контекста и семантики слов;
- **Приведение к нижнему регистру**: алгоритм может воспринимать слова в разных регистрах как разные, поэтому приведение всех слов к нижнему регистру помогает уменьшить размер словаря и упростить обучение;
- **Удаление лишних пробелов:** лишние пробелы и переносы строк могут усложнить обработку текста, поэтому текст нормализуется к формату одного пробела между словами.

Второй этап обработки включал **удаление стоп-слов** – таких слов, которые встречаются чрезвычайно часто и мало добавляют к смыслу текста с точки зрения моделирования (например, предлоги, союзы, частицы и т.д.); **лемматизацию** – процесс приведения слова к его начальной форме, что уменьшает колебания и разнообразия словоформ в текстах, делая данные более последовательными для анализа (см. Таблица 3). Второй этап включал использование таких библиотек и инструментов для обработки естественного языка как:

- **NLTK**[41]: библиотека, которая включает в себя множество утилит для текстовой обработки, включая поддержку множества корпусов, инструменты для токенизации, стемминга, тэгирования, парсинга и машинного обучения;

[40] Beautifulsoup [Б. м.]. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (дата обращения: 05.02.2024).

[41] NLTK. [Б. м.]. URL: https://www.nltk.org/ (дата обращения: 05.02.2024).

- **PyMystem3**[42]**:** обертка для mystem от Яндекса, применяемая для лемматизации слов в русском языке; mystem – это морфологический анализатор для русского языка, способный лемматизировать текст с высокой точностью, PyMystem3 предоставляет Python API для использования Mystem.

Таблица 3 – Пример данных из датасета data до и после этапов предобработки

| **text** | **text_preprocessing** |
|---|---|
| * Завтра с утра — в Москву. Что-то меня там ждет, каково-то там настроение… Как широко пользуются свободой клеветы пакостники… Работа литературная мною будет поставлена на первое место, ведь жить осталось немного. Каждый человек, в том числе и я,\xa0— талант, а талант — хрустальная чаша с нектаром. Нужно, чтобы чаша была в движении и расплескивала бы свой нектар… * | *завтра утро москва ждать каково настроение широко пользоваться свобода клевета пакостник работа литературный поставлять первый место жить оставаться немного каждый человек число талант хрустальный чаша нектар нужно чаша движение расплескивать свой нектар* |

Так как корпус мультиязычный, была также проведена детекция языков с помощью библиотеки Langdetect[43]. Было найдено и удалено 562 текста с перевесом слов на отличном от русского языках, в частности на немецком, английском, французском, польском, молдавском, итальянском, каталанском, вьетнамском, норвежском и др. Также были удалены строки с пустыми значениями и детектируемыми значениями как «error». Более того, были удалены отдельные слова с буквами, не входящими в кириллицу, и слова с буквой «і» (например, *лір, пісьмо*), которые отличают белорусский и украинский языки. Дополнительно были удалены 60 строк с аномальными

[42] MyStem. [Б. м.]. URL: https://yandex.ru/dev/mystem/ (дата обращения: 05.02.2024).

[43] Langdetect. [Б. м.]. URL: https://pypi.org/project/langdetect/ (дата обращения: 06.02.2024).

значениям (отрицательные значения и значения выше 100) в столбце «age», содержащем информацию о возрасте авторов дневников.

Итоговой корпус после предобработки содержал **2139 дневников**, **437 230 текстов**, **39 544 413 токенов**. Текст с максимальным количеством токенов содержал 9855 единицы. **Средняя длина предобработанных текстов составила 90 токенов**. Стоит заметить, что данные даже после обработки могут включать шум, поскольку эта обработка является автоматической.

### 3.3 Эксперименты по выделению признаков для моделей глубинного обучения

Основным компонентом глубинного обучения для языка является использование уровня встраивания, отображение дискретных символов в непрерывные векторы в пространстве с относительно низкой размерностью[44]. При встраивании слова превращаются из изолированных отдельных символов в математические объекты, с которыми уже может работать компьютер. В частности, расстояние между векторами может быть приравнено к расстоянию между словами, что упрощает обобщение отношения одного слова к другому. Представление слов в виде векторов воспринимается нейронной сетью как часть процесса обучения. Поднимаясь вверх по иерархии, сеть также учится комбинировать векторы слов таким образом, чтобы это было информативно для прогнозирования[45].

Для подбора лучшего алгоритма в моделях глубинного обучения векторизации для наших данных были использованы методы мешка слов Bag of Words) и Word Embeddings, представляющие наиболее популярные методы. Список извлеченных признаков использовался для определения метки класса гендера автора текста с использованием модели классификации.

---

[44] Aggarwal Charu C., Zhai ChengXiang. Mining Text Data // A Survey of Text Classification Algorithms (Chapter 6). Boston, 2012. P. 163-168.

[45] Гольдберг Й. Нейросетевые методы в обработке естественного языка / пер. с англ. А. А. Слинкина. М. : ДМК Пресс, 2019. С. 22.

Для создания внутреннего словаря был на языке программирования Python был написан код, который использует объект Tokenizer из библиотеки Keras, являющейся высокоуровневой надстройкой над TensorFlow и предоставляющей модули для создания нейронных сетей. С помощью объекта класса Tokenizer обозначались максимальное количество слов, учитываемое на следующем этапе при обучении текстов, и параметры для обработки текста, описанные выше. Метод fit_on_texts() использовался для сбора словаря частотности на основе обучающих текстов. Метод texts_to_sequences() преобразовывал тексты из переменной trainText, содержащей обучающую выборку, в последовательности индексов согласно собранному частотному словарю, сохраняя результаты в переменных trainWordIndexes и testWordIndexes. Таким образом, был создан внутренний словарь, тексты были преобразованы для последующего обучения нейронной сети.

В процессе формирования обучающей выборки происходило разбиение последовательности индексов слов на окна фиксированной длины с заданным шагом. Функция getSetFromIndexes (см. Приложение Г) используется для формирования обучающей выборки на основе списка индексов слов, а также разделения этой выборки на короткие векторы. Входные параметры функции включают последовательность индексов слов wordIndexes, длину окна xLen и шаг окна step. Функция проходит по списку индексов слов и формирует короткие векторы длиной xLen, начиная с начального индекса index и смещаясь вперед на значение step. При этом проверяется, что индексы слов находятся в допустимом диапазоне и ограничиваются до значения 1000. Полученные короткие векторы добавляются в обучающую выборку xSample, которая затем возвращается из функции. Далее с помощью функция createSetsMultiClasses (см. Приложение Г) формируем обучающую выборку для множества классов на основе последовательностей индексов слов. Для каждого класса происходит разделение текста на окна и создание общего

набора данных xSamples, содержащего окна текста, и ySamples, содержащего соответствующие векторы классов.

Далее происходит формирование обучающей и тестовой выборки для последующего использования в моделях нейросетей. Длина отрезка текста xLen была установлена равной 500 словам, а шаг разбиения текста на обучающие векторы step равен 100, поскольку эксперименты показали (Таблицы 4-5), что уменьшение длины отрезка анализа вело к снижению значений точности, в случае шага разбиения исходного текста на обучающие векторы уменьшение повышало значение точности.

Таблица 4 – Исследование влияния отрезка анализа xLen на значение точности

| | **model** | **xLen** | **accuracy** | **val_accuracy** |
|---|---|---|---|---|
| 0 | model_1 | 100 | 0.699 | 0.674 |
| 1 | model_2 | 300 | 0.790 | 0.735 |
| 2 | model_3 | 500 | 0.827 | 0.783 |

Таблица 5 – Исследование влияния шага разбиения step на значение точности

| | **model** | **step** | **accuracy** | **val_accuracy** |
|---|---|---|---|---|
| 0 | model_1 | 1000 | 0.744 | 0.701 |
| 1 | model_2 | 500 | 0.791 | 0.752 |
| 2 | model_3 | 100 | 0.827 | 0.783 |

С помощью функции createSetsMultiClasses извлекается обучающая и тестовая выборки из последовательностей индексов слов trainWordIndexes и testWordIndexes. Для каждого из классов создается тестовая выборка из индексов слов. В результате получаем готовые данные для дальнейшего использования в моделях, где текст представлен в виде последовательности индексов слов.

В ходе эксперимента с обучением моделей для предсказания пола с общей архитектурой (Рисунок 10) был проведен сравнительный анализ между использованием Bag of Words и Embeddings. В модели был использован Sequential API библиотеки Keras. Архитектура состояла из следующих слоев: полносвязный слой (Dense) с 200 нейронами и функцией активации ReLU, этот

слой принимает входные данные размерности maxWordsCount и выполняет линейное преобразование, за которым следует применение функции активации ReLU для введения нелинейности; слой Dropout с коэффициентом отсева 0.25, который помогает предотвратить переобучение модели путем случайного обнуления выходов нейронов во время обучения; BatchNormalization слой, который нормализует активации предыдущего слоя по мини-пакетам данных для ускорения обучение и улучшения стабильности модели; финальный слой Dense с 2 нейронами (по количеству классов – автор мужчина / автор женщина) и функцией активации softmax, которая предсказывает вероятности принадлежности к каждому из двух классов.

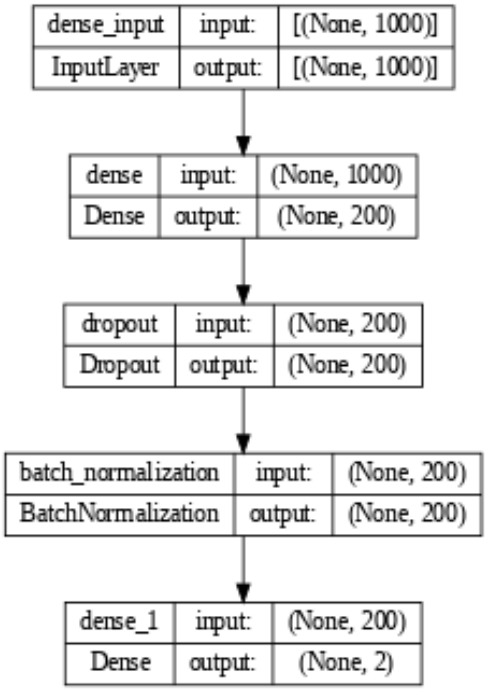


Рисунок 10 – Тестовая архитектура сети для сравнительного анализа методов векторизации

Для обучения модели был выбран оптимизатор Adam, функция потерь Categorical crossentropy (так как решается задача классификации на два класса) и метрика качества Accuracy (далее по тексту – точность).

Было обнаружено, что модель с эмбеддингами выдает более высокую точность работы (0.827 > 0.722) при установленном размере батча 64 и количестве эпох 35. Это можно объяснить следующими причинами:

1) Учет семантической информации: эмбеддинги позволяют модели учитывать семантическую информацию слов, представляя их в векторном пространстве;
2) Уменьшение размерности: эмбеддинги позволяют уменьшить размерность входных данных, что может улучшить обобщающую способность модели и ускорить обучение;
3) Лучше обработка редких слов: эмбеддинги могут лучше обрабатывать низкочастотные слова или слова, которые отсутствуют в словаре, благодаря способности обобщать информацию о словах на основе контекста.

Таким образом, использование Embeddings вместо Bag of Words позволило модели лучше улавливать семантическую информацию, уменьшить размерность данных и эффективно обрабатывать разнообразные текстовые данные, что в итоге привело к повышению точности работы модели. Дальнейшие эксперименты с подбором наиболее эффективной модели глубинного обучения осуществляются на базе метода векторизации текста с использованием Embeddings.

### 3.4 Классификация дневниковых записей

Ниже приведены эксперименты по предсказанию гендера, возрастной группы автора и эпохи создания записи. Предсказание гендера является задачей бинарной классификацией, остальные признаки относятся к задаче мультиклассовой классификации. В качестве бейзлайна для предсказания всех признаков в проведенных экспериментах была выбрана модель логистической регрессии с TF-IDF векторизатором как самая простая, не требующая долгого обучения и сложной настройки параметров.

Среди моделей машинного обучения также рассматривался наивный байесовский классификатор в сочетании с CountVectorizer – инструментов из библиотеки scikit-learn, который создает матрицу, где строки представляют

отдельные предложения, а столбцы – уникальные слова или токены в этих предложениях.

Алгоритмы глубинного обучения включали сверточные и рекуррентные архитектуры с использованием слоя с Word Embeddings. Количество эпох для всех моделей глубинного обучения устанавливалось равное 35, но в некоторых экспериментах для предотвращения переобучения была использована функция обратного вызова (callback) – Early Stopping в Keras. Эта функция прекращает обучение, когда метрика на валидационном наборе данных перестает улучшаться, что позволяет избежать переобучения. Этот механизм следит за val_loss (потери на валидационном наборе). Если потери не улучшаются в течение заданного числа эпох (patience), обучение автоматически останавливается. Таким образом, мы реагируем на начало переобучения модели после n-ой эпохи, предположительно предотвращая дальнейшие проблемы с обучением. Эта стратегия помогает сохранить время и вычислительные ресурсы, автоматически оптимизируя процесс обучения и не допуская излишнего «запоминания» данных.

Также использовался метод ReduceLROnPlateau, который аналогично является одним из методов обратного вызова. Он позволяет динамически уменьшать скорость обучения (learning rate) в процессе обучения модели, если достигнут плато в улучшении результатов. Она позволяет установить метрику, по которой будет отслеживаться плато (в нашем случае, monitor='val_loss'), коэффициент, на который будет уменьшаться скорость обучения (factor=0.5), рассмотренный параметр с числом эпох (patience=3) и др.

Так как большинство признаков имело неравномерное распределение классов, была использована стратификация данных при делении выборок на тестовые и обучающие, при которой учитывается одинаковое соотношение классов при делении выборок, чтобы избежать влияния дисбаланса классов на качество работы модели

Выбранная метрика для оценки моделей – Accuracy, которая дальше обозначается как точность (не путать с precision). Для получения этой метрики в моделях машинного обучения использовалась функция classification_report для генерации отчета о классификации, который содержит различные метрики оценки качества работы модели. В моделях глубинного обучения – функция evaluate из Keras, используемая для оценки производительности модели нейронной сети на тестовом наборе данных. На них делался прямой проход (forward pass) через сеть, после получения выходных значений модель возвращала значение метрики, указанной при компиляции модели, и функцию потерь (loss function).

Для обучения были использованы бесплатные вычислительные ресурсы Google Colab[46], предоставляющие доступ к гибридным графическим обработчикам (GPU) и центральным процессорам (CPU) с ограничением памяти в 12.67 ГБ и времени на использование GPU, что требовало в работе оптимизации процесса, ресурсов и параметров, таких как размер батчей, словаря и сложность моделей. Также использовалась платформа Kaggle[47] с ограничением работы процессоров до 30 часов в неделю.

### 3.4.1 Классификация дневниковых записей по полу авторов

Классификация текстов по полу является сложной задачей из-за того, что стиль письма может сильно варьироваться даже у лиц одного пола. Однако, мужчины и женщины часто имеют различия в выборе слов, употреблении эмоциональной лексики, структуре предложений и тематике текстов[48]. Использование автоматических методов анализа текста может помочь выявить эти различия и провести классификацию.

[46] Google Colab. [Б. м.]. URL: https://colab.research.google.com/ (место обращения: 18.04.2024).

[47] Kaggle. [Б. м.]. URL: https://www.kaggle.com/ (дата обращения: 18.04.2024).

[48] Кирилина А. В. Гендер и язык. Антология. М. : Языки славянской культуры, 2005. С. 32-34.

Предсказание пола авторов по тексту является достаточно популярной задачей не только в силу практической применимости в маркетинге, рекламе, персонализации контента и т.д., но и в силу уникальности этого признака и доступности ряда формальных причин:

- Явные лингвистические маркеры: пол может проявляться через использование определенных слов и конструкций, которые статистически чаще употребляются мужчинами или женщинами; это делает задачу более выполнимой для алгоритмов машинного обучения;
- Этические аспекты: модели, определяющие пол, обычно требуют меньшего объема личных данных по сравнению с другими задачами, например, определение уровня образования, при котором требуется доступ к специфичным и зачастую конфиденциальным данным, тогда как данные о поле могут быть собраны анонимно и с меньшими этическими рисками;
- Сравнительная простота в оценке: пол проще может быть проверен и подтвержден в литературе и реальных данных; также четкие бинарные категории упрощают обработку данных и обучение моделей.

В итоге, предсказание пола по тексту представляет собой четко определенную, практически значимую задачу. Существуют разные модели по автоматическому определению пола автора текста: Text2Gender[49] на основе архитектуры BERT, гендерный классификатор электронных писем[50] с использованием логистической регрессии, решающих деревьев и метода

[49] Thakur V., Tickoo A. Text2Gender: A Deep Learning Architecture for Analysis of Blogger's Age and Gender // arXiv preprint arXiv:2305.08633. 2023. 10 p.

[50] Cheng N., Chandramouli R., Subbalakshmi K. P. Author gender identification from text // Digital investigation. 2011. Т. 8. №. 1. P. 78-88.

опорных векторов; нейронная сеть на базе LSTM с включением синтаксической структуры предложения[51] и др.

В нашем решении задачи были использованы методы классического машинного обучения (логистическая регрессия, наивный Байесовский классификатор) и глубинного обучения (CNN1D, LSTM, BiLSTM).

Большинство дневников в исследовательском корпусе принадлежит мужчинам, что видно по количественным данным с распределением значений по признаку в Таблице 6 и более наглядно на рисунке 11.

Таблица 6 – Количественные данные признака «Пол автора»

| **Название признака** | **Значение признака** | **Количество текстов** |
|---|---|---|
| Пол автора | Мужской пол | 361652 |
| | Женский пол | 75578 |

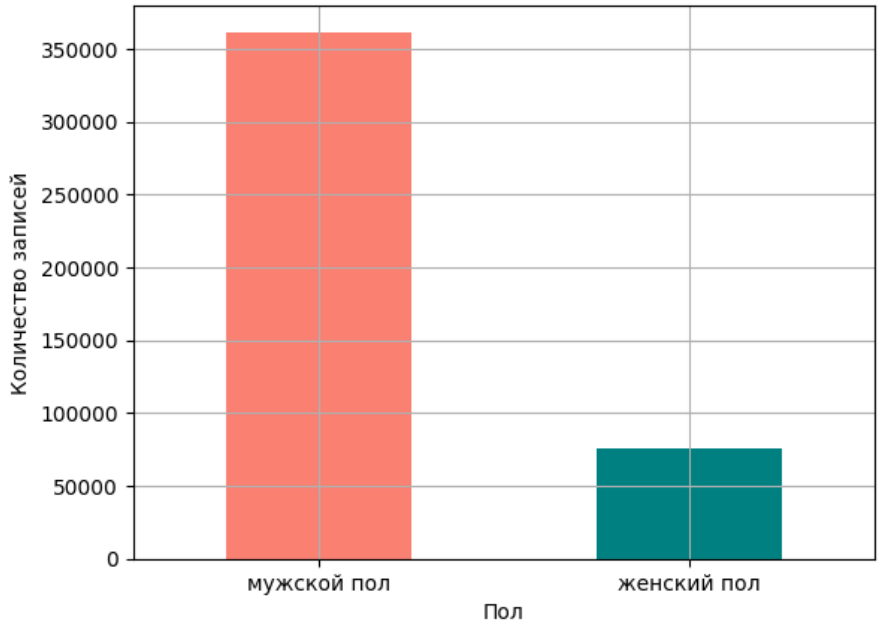


Рисунок 11 – Распределение текстов по гендеру

Для модели логистической регрессии был использован TfidfVectorizer для преобразования текстовых данных в числовые признаки. Модель была обучена на полученных признаках и протестирована на тестовой выборке.

[51] Сбоев А. Г. и др. Модель нейронной сети для включения синтаксической структуры предложения в задачу классификации пола автора русского текст // Вестник НИЯУ МИФИ. 2023. Т. 8. №. 6. С. 569-576.

Результаты показали высокую точность (accuracy) в 89%, с макро-усредненным f1-score 0.78. Для наивного Байеса был использован CountVectorizer для преобразования текстовых данных в матрицу признаков. Модель продемонстрировала точность 83%, с макро-усредненным f1-score 0.74.

На матрице ошибок (Рисунок 12) видим, что логистическая регрессия имеет меньше ошибочных предсказаний (7703 для женского пола и 1505 для мужского пола), чем байесовский классификатор (4528 для женского пола и 10033 для мужского пола). Логистическая регрессия демонстрирует более высокую точность в предсказании мужского пола, в то время как байесовский классификатор более успешен в определении женского пола.

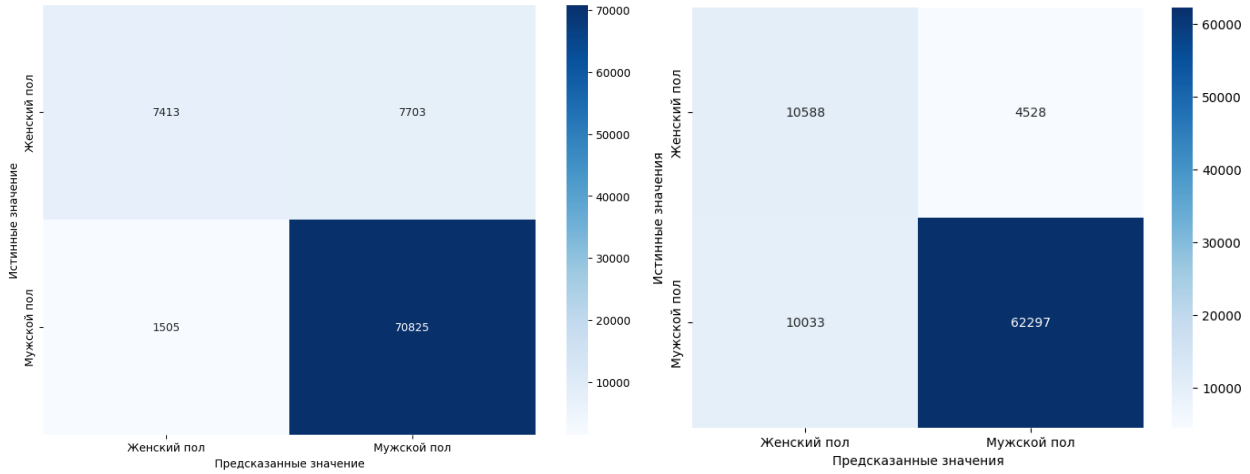


Рисунок 12 – Сравнение матриц ошибок для моделей логистической регрессии и наивного Байесовского классификатора соответственно

Три различные модели использовались при нейросетевом подходе. Каждая модель имела свою уникальную архитектуру, включая слои Embedding, Conv1D или LSTM с различными параметрами (Таблица 7). Каждая модель также использовала BatchNormalization и Dropout для регуляризации данных. Модели были скомпилированы с оптимизатором «adam» и функцией потерь «categorical_crossentropy». Обучение происходило с размером батча с числом 64 тренировочных объектов. Слой эмбеддинга составил словарь размером 5000 и размерностью выходного пространства 64, применяемую к последовательностям длиной 100. Точность моделей

составила 83.57% (Conv1D), 86.82% (LSTM + Conv1D) (Рисунок 13) и 86.57% (BiLSTM + LSTM).

Таблица 7 – Сравнение разных архитектур и параметров в классификации гендера автора текста

| **Основная модель** | **Параметры ключевого слоя / модели** | **Точность модели** | **Значение ошибки** |
|---|---|---|---|
| LogitRegression | TfidfVectorizer | **0.8947** | - |
| Multinominal Naive Bayes | CountVectorizer | 0.8334 | - |
| Conv1D | (filters=64 / 128, kernel_size=3, padding='same', activation='relu') | 0.8356 | 1.3024 |
| LSTM + Conv1D | Conv1D(256, 3, activation='relu'), LSTM(128, return_sequences=True) | **0.8682** | 0.4065 |
| BiLSTM + LSTM | Bidirectional(LSTM(64, return_sequences=True)) | 0.8657 | 0.4098 |

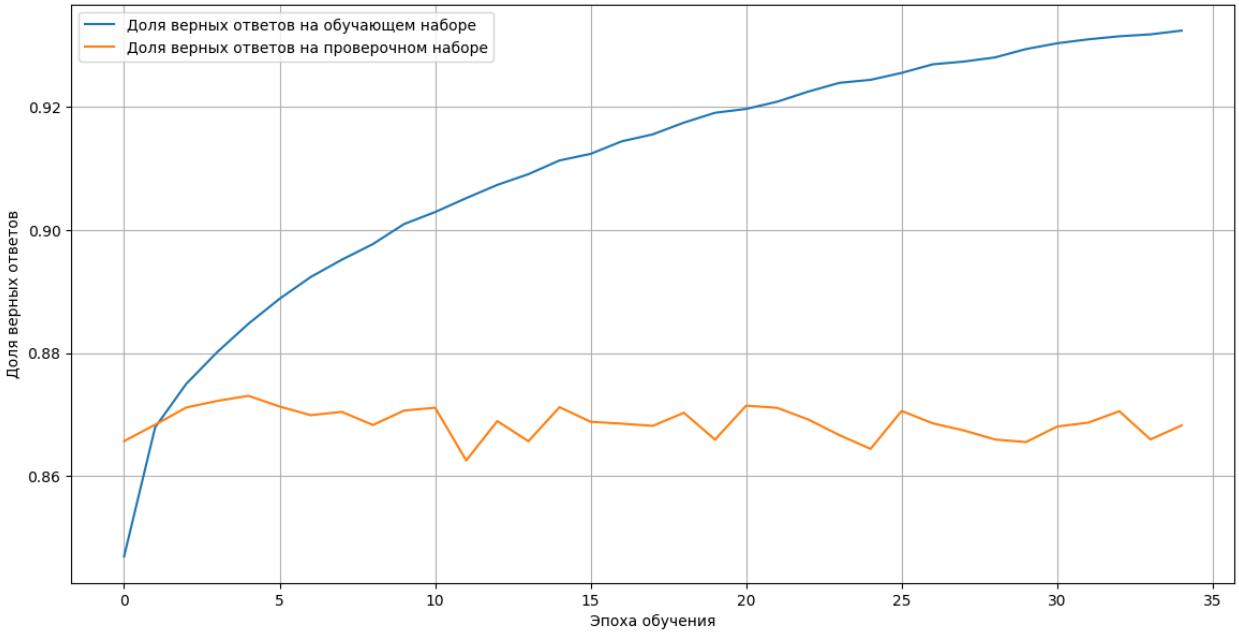


Рисунок 13 – График точности работы сборной модели из CNN1D и LSTM при классификации гендера автора

Дополнительно для логистической регрессии, показавшей высокий результат, был использован метод определения коэффициентов весов (get_feature_names_out(), coef_.flatten()), позволяющий определить, какие

слова сильнее всего повлияли на решение отнесения текста к тому или иному классу. Список слов с наибольшими коэффициентами выглядит следующим образом: *ирка (6.70), аликс (6.01), зинуля (5.91), юрка (5.78), митька (5.14), кратово (4.70), дора (4.68), юр (4.64), монета (4.55), жена (4.51), теща (4.18)* и др. Коэффициенты в скобках, присвоенные каждому слову, указывают на степень влияния каждого слова на предсказание гендера автора текста. Чем выше коэффициент, тем сильнее слово связано с гендером. Можно видеть, что классификатор текста в значительной мере опирается на имена собственные, топонимы и уменьшительно-ласкательные варианты имен, что подчеркивает специфику выбранного материала. Также можно обратить внимание на слова, обозначающие семейные связи (*жена (4.51), теща (4.18)*) и социально-культурные термины (*консерватория (3.93), катихизатор (3.74), семинария (3.49), литургия (3.40)*), которые могут нести гендерные оттенки в зависимости от культурного и исторического контекста. В целом, можно сделать вывод, что модель установила связь между определенными словами и полом автора текста. Эти слова имеют выраженные гендерные ассоциации в контексте, например, обращений к противоположному полу, что помогает при принятии решений модели. Более того, это подтверждает тезис о большом количестве онимов в текстах дневниковых записей.

Дополнительно была предложена эвристика в виде использования для обучения лемматизированных текстов с сохранением глаголов в прошедшей форме. Гипотеза, нацеленная на поднятие точности работы модели, основывается на следующем – сохранение прошедших форм глагола может играть значительную роль в классификации текста по гендеру, особенно в языках с ярко выраженными грамматическими родами, таких как русский язык. В русском языке глаголы в прошедшем времени изменяются по роду, и сохранение этой формы может дать полезные сигналы модели. Например, «ходил» (мужчина) против «ходила» (женщина). Для дневниковых записей такой подход кажется релевантным, поскольку чаще всего записи ведутся от

первого лица: (15155, 44) *Я **сделал** несколько предложений, как сделать роль, не увеличивая ее в объеме. Всем нравится. Принимают. Но боюсь, что все это делается для того, чтобы успокоить меня и скорей начать съемки*; (13893, 36) *Москва. Сегодня состоялось собрание членов Московского отдела. Я **сделала** доклад об «Искусстве Индии». Создалось глубокое и тихое настроение. На миг мы пришли в соприкосновение с душой индийского народа и пережили настроение, которое создало его мысли, его картины и его дивную архитектуру.*

Лемматизация всех остальных слов помогает избавиться от морфологических вариаций, не влияющих на распознавание пола, сохраняя при этом наиболее значимые из них. Из очевидных недостатков такого подхода: дополнительная сложность; автоматическая обработка может допустить ошибки при распознавании форм глаголов и их временных характеристик, что является следствием синтаксической омонимии, это рискует снизить достоверность данных; неполная лемматизация может привести к отклонениям в анализе частей речи и синтаксических зависимостей, что может запутать модель вместо улучшения ее эффективности; дневниковые записи могут вестись и не от первого лица, также часто встречаются описания действий третьих лиц.

Для проверки этой гипотезы тексты были обработаны созданным морфоанализатором (см. ение Д) с помощью библиотеки Pymorphy2[52] и записаны в столбец «text_preprocessing_past_verbs» вместе с исходным лемматизированным текстом.

Изменение значения точности на новых данных проверялось на модели логистической регрессии как на базовой и на модели, которая лучше всего себя показала на лемматизированных текстах – сочетание архитектур LSTM и Conv1D.

[52] Pymorphy2. [Б. м.]. URL: https://pymorphy2.readthedocs.io/en/stable/ (дата обращения: 02.03.2024).

Таблица 8 – Сравнение разных архитектур и параметров в классификации пола автора с сохранением в обучающих текстах форм прошедшего времени глаголов

| **Основная модель** | **Параметры ключевого слоя / модели** | **Точность модели** | **Значение ошибки** |
|---|---|---|---|
| LogitRegression | TfidfVectorizer | **0.9172** | - |
| LSTM + Conv1D | Conv1D(256, 3, activation='relu'), LSTM(128, return_sequences=True) | 0.8846 | 0.2154 |

Видим по результатам в таблице 8 и на рисунке 14, что значения точности модели выросли примерно на 3%. Список слов с наибольшими коэффициентами пополнился формами прошедшего времени глаголов: *видел (11.37), получил (11.20), ходил (9.44), смотрел (8.44), читал (8.41), написал (7.62), занимался (7.55), думал (7.46) и др.*

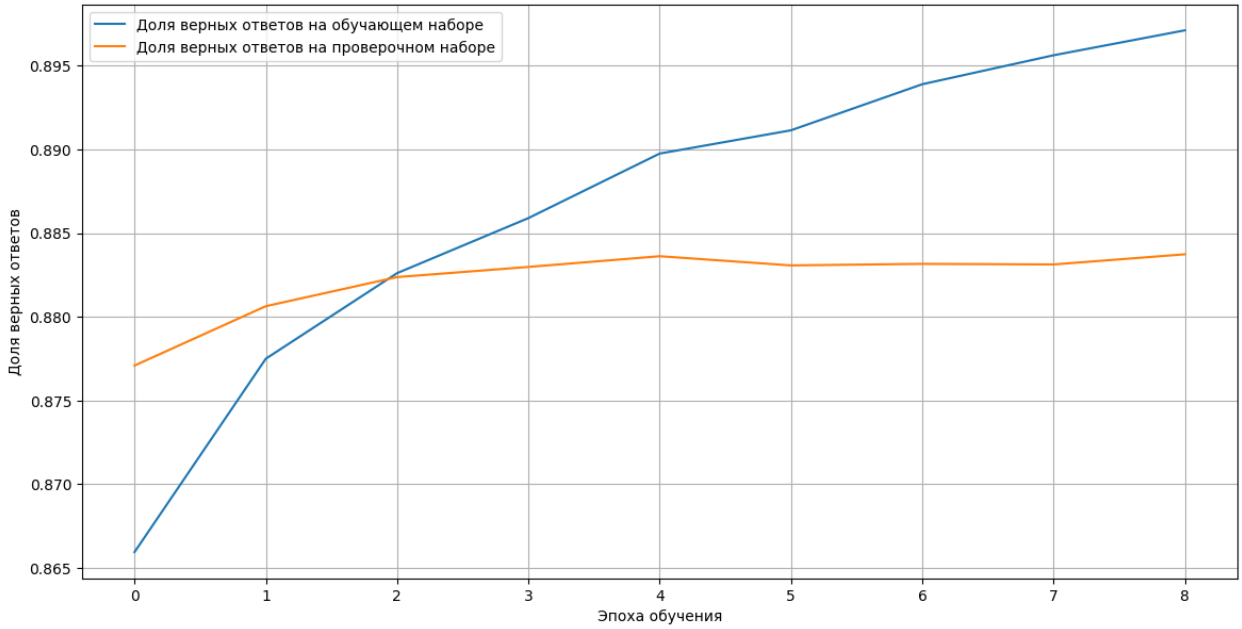


Рисунок 14 – График точности работы сборной модели из CNN1D и LSTM при классификации гендера автора с сохранением форм прошедшего времени

Модель логистической регрессии с использованием TfidfVectorizer показала наивысшую точность (89% и 92%) в обоих случаях. Это может быть связано с тем, что TF-IDF эффективно выделяет значимые слова, которые могут быть ключевыми для идентификации гендера писателя, работая с разреженными данными. Предположительно, для определения гендера автора текста важнее обращать внимание на уникальные слова, чем на

семантическую информацию и контекст слов, которые учитываются при обработке последовательностей слов в нейросетях с помощью эмбеддингов слов. Что также доказывает небольшой прирост в значении точности работы модели с сохранением форм прошедшего времени обучающих текстах, где также свою значимость имеют отдельные слова.

### 3.4.2 Классификация дневниковых записей по возрастным группам авторов

Задача предсказание возраста автора по тексту является актуальной и востребованной в области обработки естественного языка и машинного обучения. Многие исследователи разрабатывали модели, ставя перед собой эту задачу: наивный байесовский классификатор, обученный на текстах из социальных сетей, предсказывающий написан ли текст подростком или взрослым[53]; разные модели машинного обучения (Support Vector Machine, Random Forests, AdaBoost и др.), предсказывающие возрастные группы по постам Twitter – 13-17, 18-24, 25 и старше[54]. Применимость методов для такой задачи варьируется от маркетингового анализа и таргетированной рекламы до социологических и психологических исследований, где возрастные аспекты играют ключевую роль.

Наши данные с дневниковыми записями имеют следующими характеристики: средний возраст – 45 года, медианный возраст – 43 год, основная выборка лежит в границах от 30 до 57 лет (Рисунки 15-16).

[53] Asogwa D. C. et al. Development of a machine learning algorithm to predict author's age from text // International Journal of Research, 2019. Т. 7. № 10. 380-389 p.

[54] Morgan-Lopez A. A. et al. Predicting age groups of Twitter users based on language and metadata features // PloS one. 2017. Т. 12. №. 8. 12 p.

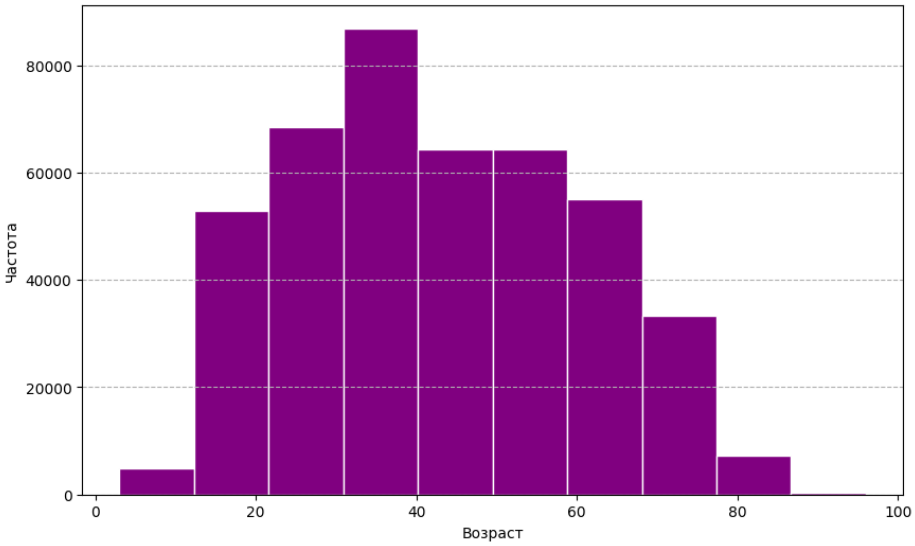


Рисунок 15 – Распределение текстов по возрастам

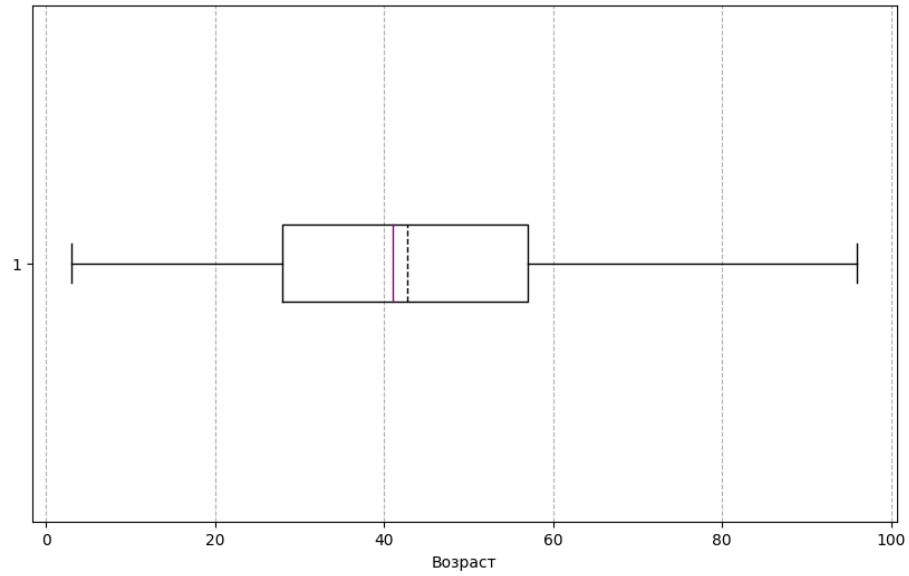


Рисунок 16 – Диаграмма размаха для возрастов

Для выделения возрастных групп был использован метод кластеризации для выявления скрытых паттернов в данных и более точного разделения выборки на возрастные группы. Для выполнения была написана модель K-means с использованием TF-IDF и понижением размерности с помощью SVD (Singular Value Decomposition), который разлагает матрицу на три более простые матрицы, что позволяет сократить размерность. По результатам работы модель выделила три кластера, которые имеют достаточно четкие границы на рисунке 17.

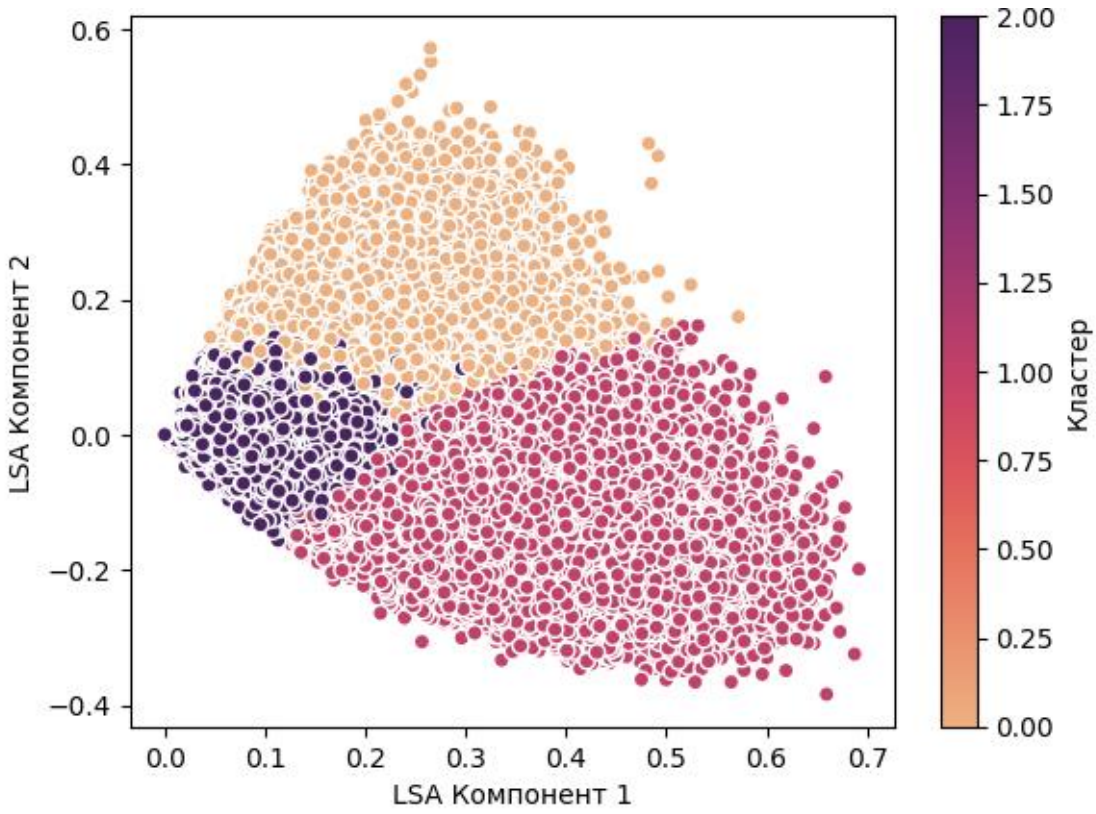


Рисунок 17 – Кластеризация текстов по возрасту с помощью K-means

Чтобы понять, какие возраста входили в кластеры, была написана функции для вывода топ-10 возрастов по частоте в группах. Первая группа включала в себе большинство значений от 55, вторая группа состояла из выборки значений от 32 до 50, третья включала в большинстве значения меньше 32. На основе этих данных с небольшим смещение для сохранения баланса классов были выделены возрастные группы для обучения модели, указанные в таблице 9. Стоит заметить, что в обучение не были включены 28614 текстов, относящихся к возрасту до 17 лет, поскольку в силу большого отличия они могут усилить коррелированность трех остальных групп.

Таблица 9 – Количественные данные признака «Возрастные группы авторов»

| **Название признака** | **Значение признака** | **Количество текстов** |
|---|---|---|
| Возрастные группы | Молодая возрастная группа (18-35 лет) | 143177 |
| | Средний-зрелый возраст (36-55 лет) | 148669 |
| | Старшая возрастная группа (56+ лет) | 102484 |

Такое деление помогает адекватно распределить выборку и обеспечить, что каждая группа будет достаточно представлена. Избежание слишком

широких возрастных интервалов позволяет лучше сбалансировать данные (Рисунок 18) и избежать смещения модели в сторону более представленных возрастных групп. Использование этих групп позволяет лучше учитывать различия в лексике, стилях письма и тематиках, которые могут быть характерны для разных возрастов.

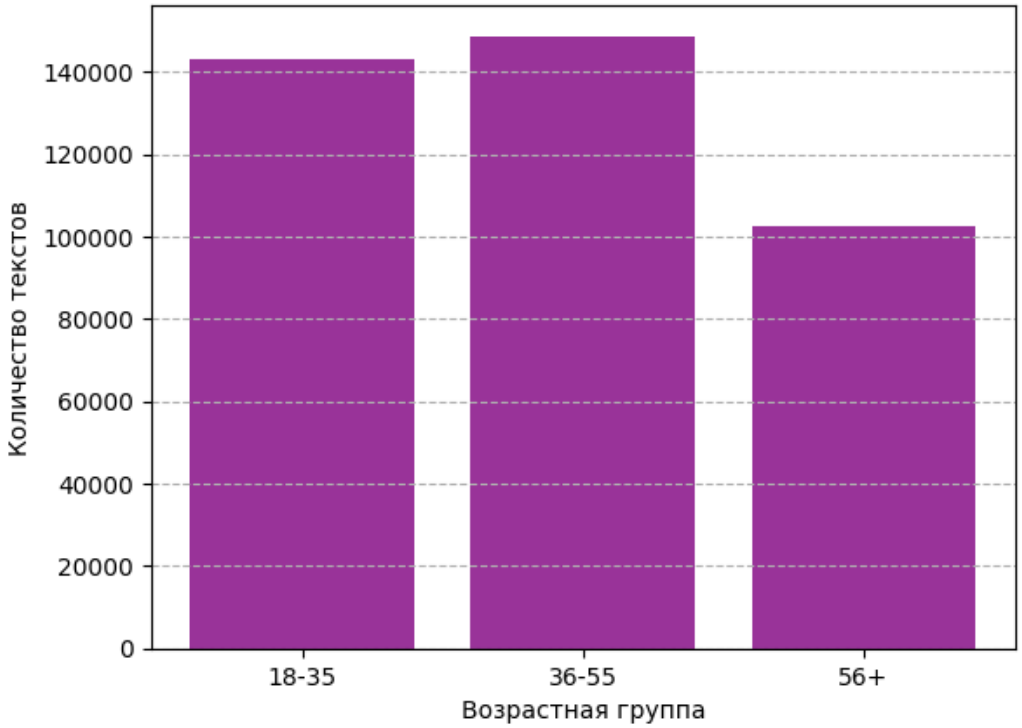


Рисунок 18 – Итоговое распределение текстов по классам возрастных групп авторов

Логистическая регрессия и наивный байесовский классификатор были обучены с аналогичными предыдущим моделям для гендерного признака методами векторизации. Модели продемонстрировали точность 71% и 66% соответственно.

Исходя из представленных матриц ошибок для предсказания возрастных групп авторов текста с применением логистической регрессии и наивного байесовского классификатора на Рисунке 19, можно сказать, что обе модели показывают хорошие результаты в предсказании возрастной группы 18-35 лет, но имеют тенденцию к ошибкам при определении старших возрастных групп.

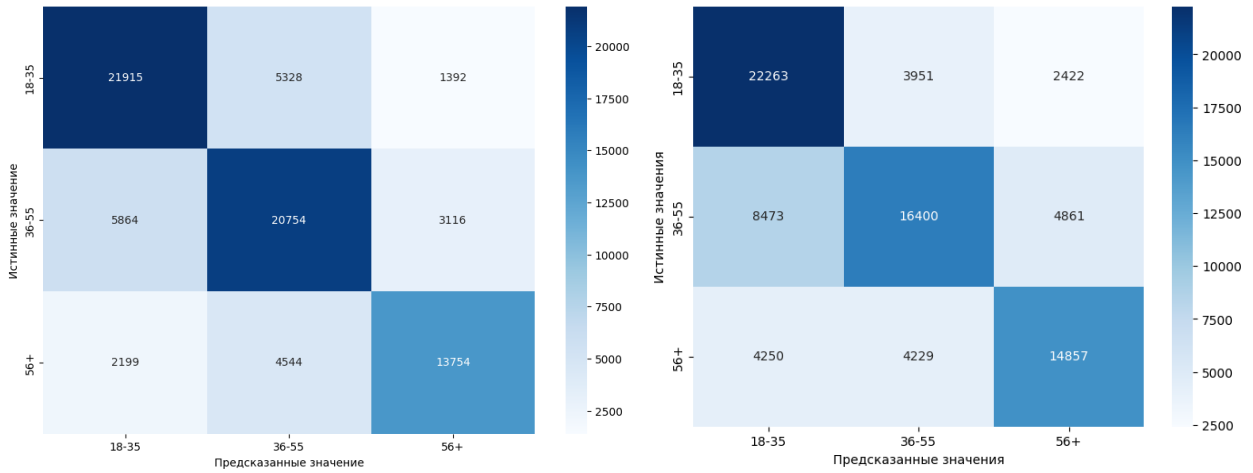


Рисунок 19 – Сравнение матриц ошибок для моделей логистической регрессии и наивного Байесовского классификатора соответственно

Список слов с наибольшими коэффициентами влияния на предсказание не показал значимых единиц, которые можно было бы отнести к разным группам: *ирка (10.87), барин (6.54), нардом (6.13), чёс (4.79), георгиевна (4.71), столярный (4.54), военком (4.41), игорёк (4.38) и др.*

В ходе экспериментов с моделями глубинного обучения были применены следующие алгоритмы (Таблица 10): одномерная сверточная нейронная сеть с параметрами (два слоя на 64 и 128 фильтров, размер ядра 3, заполнение «same», функция активации «relu») со значение точности в 63%, LSTM совместно с Conv1D (256 фильтров, размер ядра 3, функция активации «relu») (64%) и BiLSTM с LSTM (двунаправленный LSTM с 64 нейронами и возвратом последовательностей) (67%).

Таблица 10 – Сравнение разных архитектур и параметров в классификации возрастной группы автора

| **Основная модель** | **Параметры ключевого слоя / модели** | **Точность модели** | **Значение ошибки** |
|---|---|---|---|
| LogitRegression | TfidfVectorizer | **0.7154** | - |
| Multinominal Naive Bayes | CountVectorizer | 0.6634 | - |
| Conv1D | (filters=64 / 128, kernel_size=3, padding='same', activation='relu') | 0.6352 | 0.8564 |

| | | | |
|---|---|---|---|
| LSTM + Conv1D | Conv1D(256, 3, activation='relu'), LSTM(128, return_sequences=True) | 0.6395 | 0.7765 |
| BiLSTM + LSTM | Bidirectional(LSTM(64, return_sequences=True)) | **0.6689** | 0.7396 |

Таким образом, лучший результат, равный 71%, был достигнут с использованием модели логистической регрессии с TfidfVectorizer. Предположительно, это произошло из-за того, что модель лучше подходит для данной задачи классификации возрастных групп в конкретном наборе данных. Также возможно, что признаки, извлеченные с помощью TfidfVectorizer, более эффективно передали информацию для классификации возраста. Среди моделей глубинного обучения лучше всех показала себя скомбинированная модель BiLSTM и LSTM (67%) (Рисунок 20).

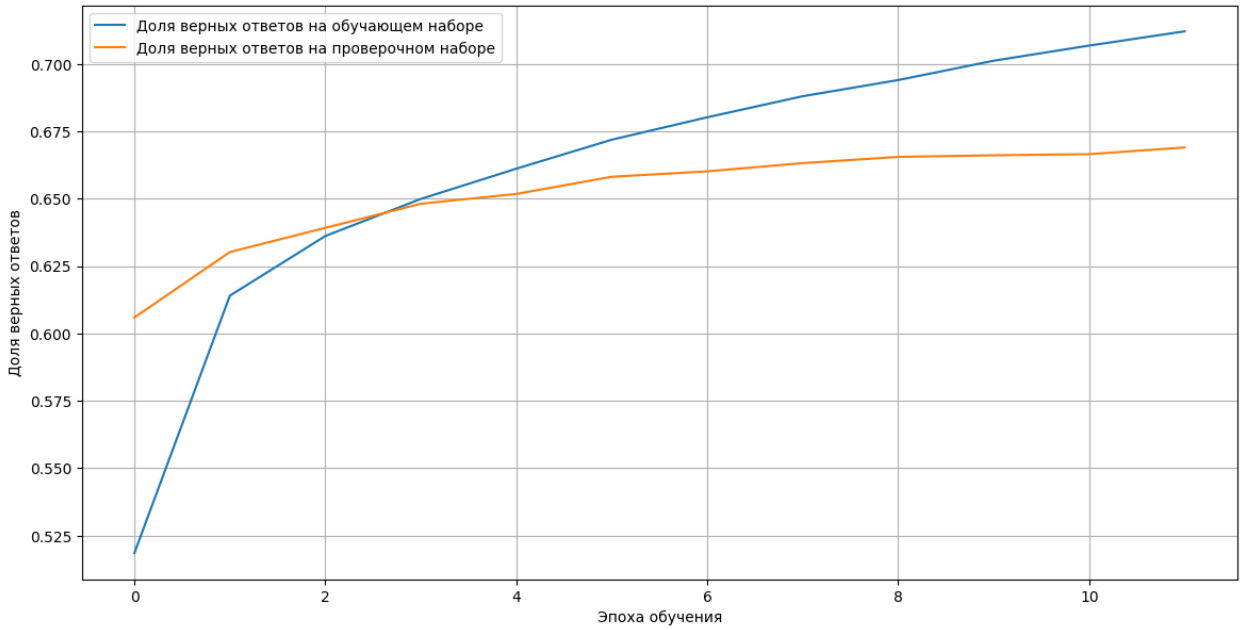


Рисунок 20 – График точности работы сборной модели из BiLSTM и LSTM при классификации возрастной группы автора

### 3.4.3 Классификация временного периода создания дневниковой записи

Предсказание временного периода создания текста позволяет исследователям и историкам определить контекст его создания. По этим данным можно сделать выводы о культурных, социальных и политических

условиях того времени. Также это позволяет лингвистам отслеживать эволюцию языка, идентифицировать изменение в грамматике, лексиконе и синтаксисе; филологам это косвенно помогает в изучении хронологии литературных трудов. Обнаружение временного периода написания текста важно для проверки подлинности исторических документов, что актуально для задачи аутентификации документов. В археологии, архивоведении и искусствоведении модели по предсказанию этого признака могут быть использованы для анализа и систематизации больших объемов текстовой информации, улучшая точность и скорость исследования.

Хотя задача предсказания времени создания текста представляет научный интерес и имеет широкие прикладные возможности, ее решение достаточно редко встречается. Предположительно, это обусловлено следующими факторами: сложность задачи – изменения в языке, стиле письма и тематике могут усложнить оценку времени создания текста; недостаточно количество аннотированных данных; индивидуальные особенности текстов – некоторые тексты могут быть написаны в уникальном стиле или содержать элементы, которые неоднозначны для определения конкретной эпохи.

В рамках подготовки к разработке модели нейросети для предсказания даты создания текста был проведен разведочный анализ данных. Основное внимание уделено распределению текстов по векам и определению ключевых временных периодов, которые имеют значительное количество записей.

Результаты разведочного анализа показали, что большинство текстов относится к 20 веку (~85% выборки), меньше всего корпус содержит текстов 18 века (1.4%) (Рисунки 21-22). Такое распределение может быть объяснено несколькими факторами: исторические события (мировые войны и значимые политические изменения часто приводят к увеличению письменной документации и личной фиксации мыслей), развитие печати и масс-медиа (20 век характеризуется значительным прогрессом в технологиях печати и распространения информации), артефакты сбора данных (корпус

сформирован из источников, больше ориентированных на недавнее прошлое, что приводит к перекосам в сторону более новых текстов).

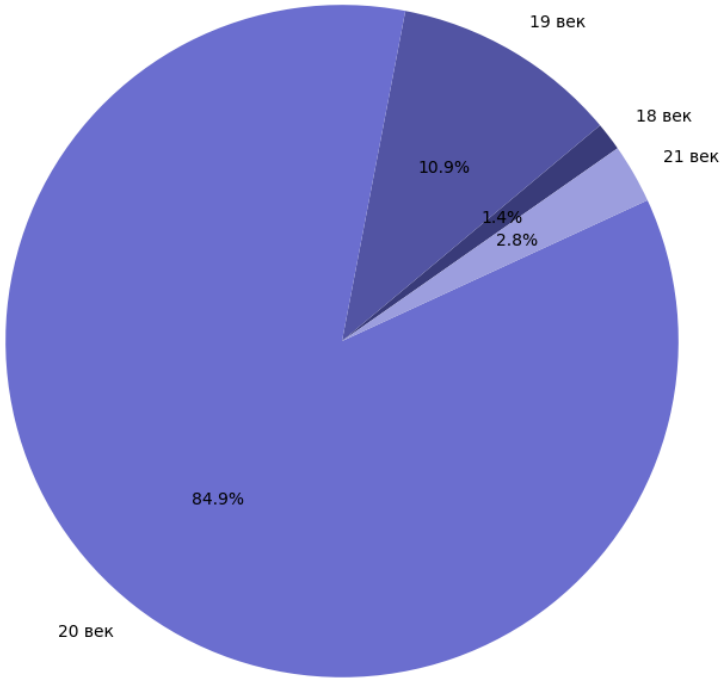


Рисунок 21 – Распределение текстов по столетиям

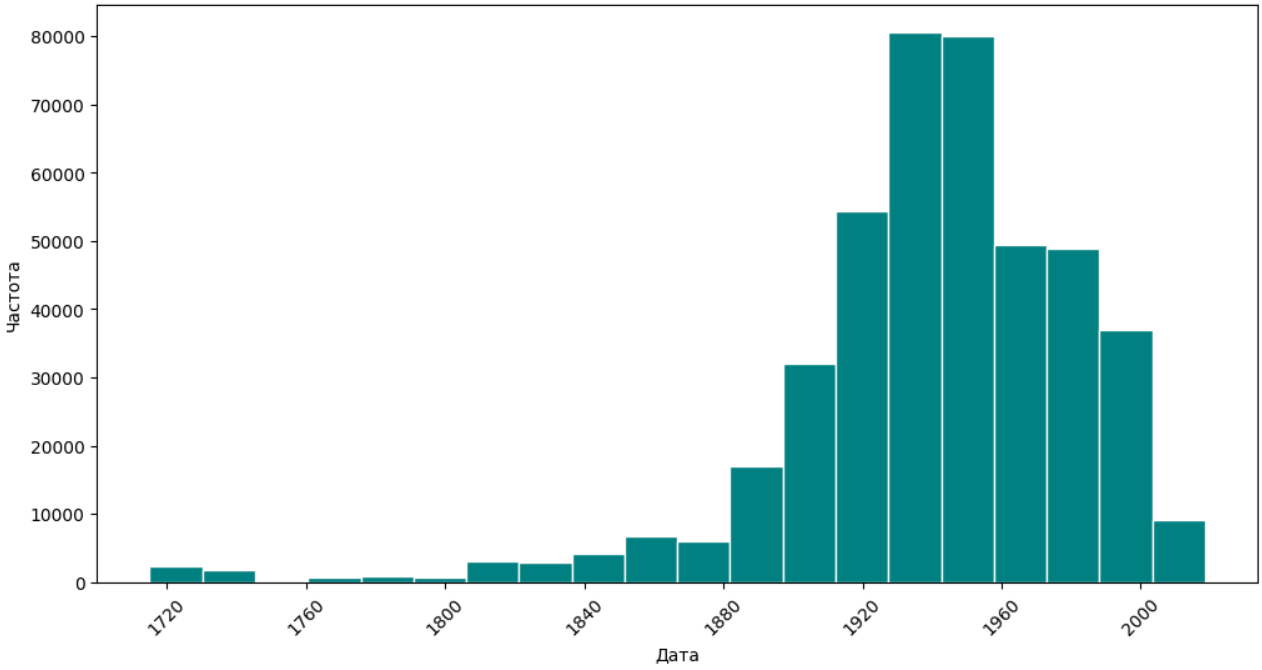


Рисунок 22 – Распределение текстов по значениям признака «date»

Дополнительно было определено, что топ-10 лет с наибольшим количеством текстов включает года из периода Второй мировой войны (1942,

1943, 1941, 1944, 1945) и период после Первой мировой войны (1918, 1919, 1917) (Рисунок 23).

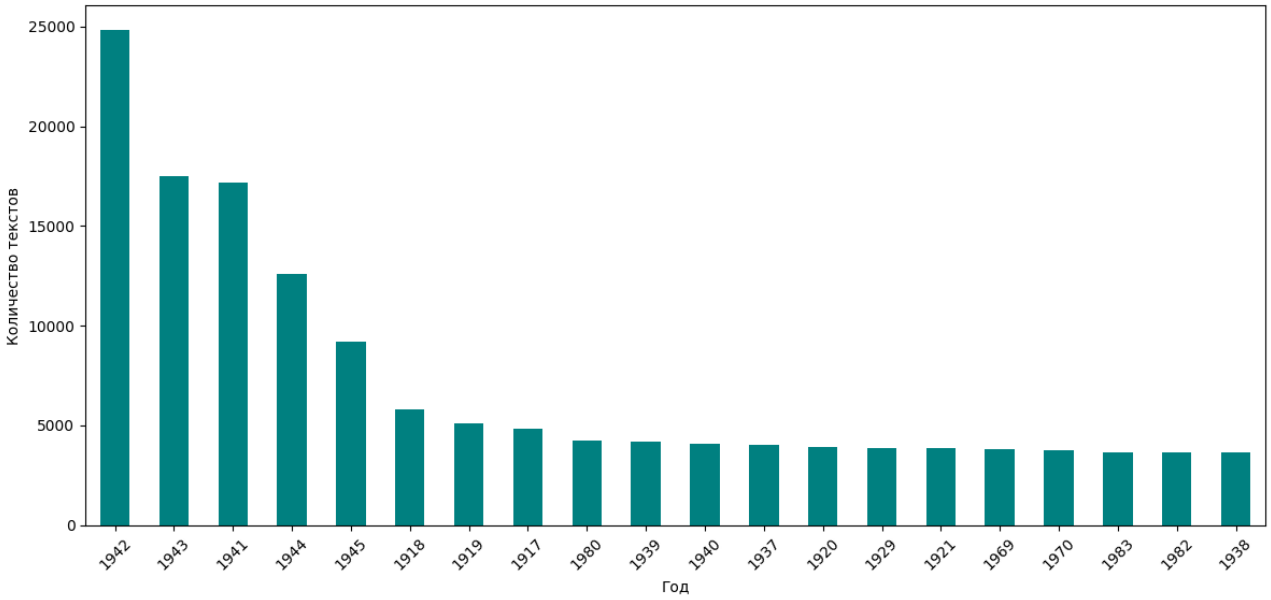


Рисунок 23 – Топ-20 годов с наибольшим количеством записей

Стоит заметить, что года с самым большим количеством текстов в 19 и 21 столетиях распределены равномернее, чем в 18 и 20 веках (Рисунок 24). Года, встречающиеся чаще остальных, в 19 веке составляют разброс от 1891 до 1900, в 21 веке – от 2001 до 2010. Для 18 века можно видеть, что концентрация созданных записей находится на отметке 1730-х годов. Для 20 века значениями с высокой частотой стали 1920-е, 1940-е и 1990-е годы.

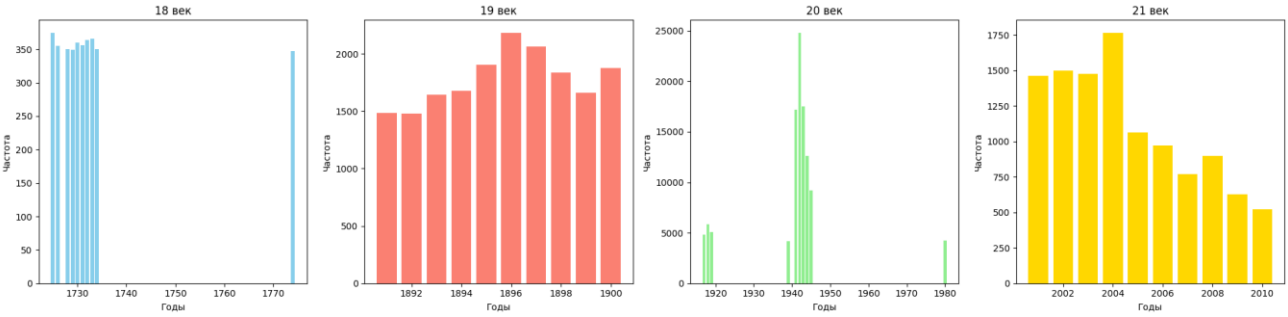


Рисунок 24 – Распределение самых частотных годов по столетиям

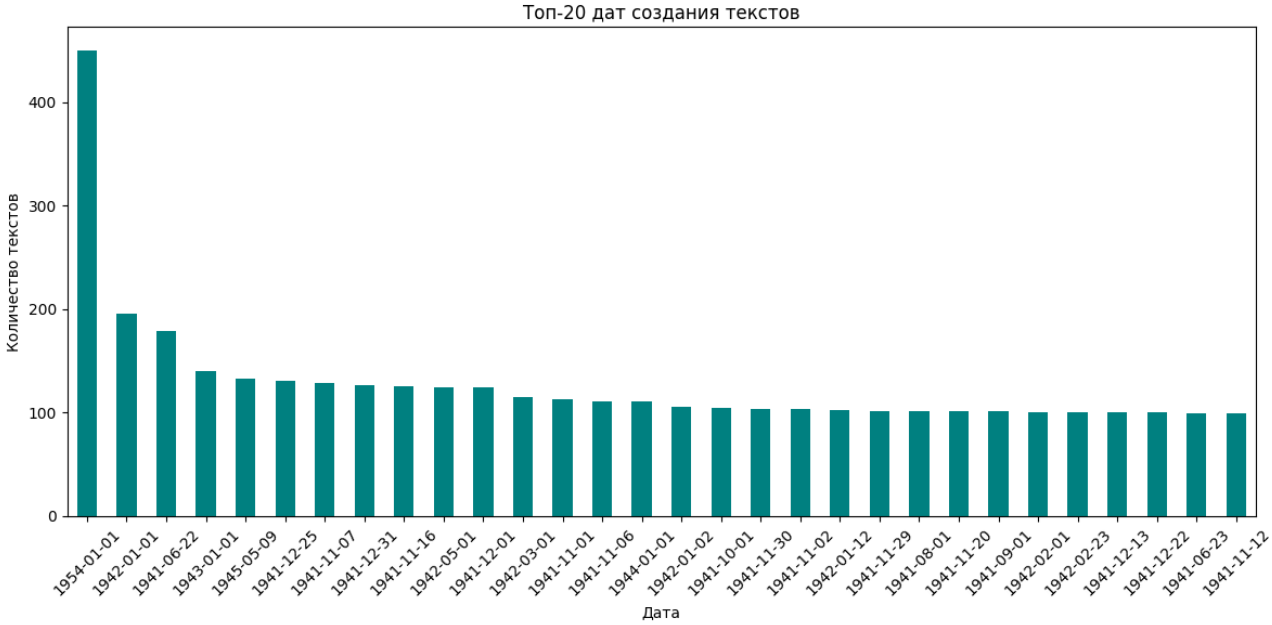


Рисунок 25 – Топ-30 дат с наибольшим количеством записей

Рассмотрим методологию моделирования. При разработке классификатора дат создания текстов основной фокус уделяется логическому разделению данных на классы, чтобы максимально эффективно использовать информацию и снизить сложность задачи для модели. Рассмотрение разделения данных на классы должно учитывать историческую значимость периодов, обильность данных и проблемы с дисбалансом классов. Выбор классов должен основываться не только на исторических фактах, но и на доступности обучающих данных для каждого класса. Разбиение должно способствовать балансу между точностью и обучаемостью модели. Анализ данных для каждого предложенного класса поможет определить достаточность данных для обучения и валидации модели. Более того, комбинация методов может включать как разделение на десятилетия внутри века, так и общую классификацию по векам для менее представленных периодов.

При подготовке данных для модели классификации предсказания даты создания текста можно рассмотреть три подхода: поделить по датам, поделить по годам, поделить по столетиям. Первый подход требует полной детализации данных, предсказания в рамках 68438 уникальных значений дат, что в

пропорции к размеру корпуса и вычислительных возможностей является нереалистичной идеей в достижении высокой точности работы модели. Также стоит заметить, что в данных большой дисбаланс дат. Большинство самых частотных дат корпуса по количеству текстовых вхождений составляют годы Великой Отечественной войны. Альтернативный подход – предсказывать год в рамках столетия. Этот подход также предполагает детализацию предсказаний, но, с другой стороны, также требует значительно больше данных для обучения по каждому году, и сохраняется повышенная сложность модели. Последний подход использует категоризацию по векам. В рамках такого обучение уменьшится количество классов и сложность модели по сравнению с другими вариантами деления классов. При таком подходе будет происходить потеря точности предсказаний, поскольку век содержит в себе слишком широкий диапазон лет, но такая категоризация является самым оптимальным вариантом в ситуации с ограниченным количеством данных по некоторым периодам. Стоит заметить, что деление текстов только на столетия не релевантно с учетом дисбаланса количества материала по векам. Самое большое количество записей, а именно – 371657, составляет 20 столетие. Во избежание переобучения модели поделим этот век на несколько классов дополнительно по следующим годам:

- 1900-1920 годы – в этот период приходится ряд ключевых исторических событий, таких как Русская революция и Первая мировая война, которые могли оказать влияние на содержание дневниковых записей;
- 1921-1940 – в этот период включены годы между двумя мировыми войнами, Гражданская война, годы большого террора, которые были временем социальных изменений и культурных трансформаций;
- 1941-1945 годы – период Великой Отечественной войны может быть выделен как отдельный класс из-за значительного влияния войны на повседневную жизнь людей;

- 1946-1960 годы – период восстановления после войны, начало Холодной войны и другие события;
- 1961-1979 годы – в этот период в СССР происходили значительные социальные и политические изменения; дневники этого времени могут содержать отражение жизни в условиях коммунистического режима, культурных трансформаций, событий хрущевской «оттепели» и становления брежневской эпохи;
- 1980-1999 годы – период эпохи перестройки и распада Советского Союза; дневники этого времени могут отражать переход от социализма к капитализму, политические и экономические трансформации, переживания людей в условиях нестабильности.

Подобное разделение на классы позволит учесть ключевые исторические события и изменения, которые происходили в течение 20 века, а также учесть специфику дневниковых записей на русском языке.

Так как не все столетия имеют достаточный объем по годам, тексты по ним были объединены в рамках всего века. Таким образом, были выделены следующие эпохи для предсказания времени создания записи:

- 1700-1799 годы – время правления царей, просвещенческое движение, развитие культуры и науки; дневники этого периода могут содержать описание повседневной жизни, социальных обычаев, религиозных убеждений и политических событий;
- 1800-1899 годы – эпоха романтизма, индустриализации, роста империализма, что может фиксироваться в дневниках через описание промышленного прогресса, социальные противоречия, культурные изменения;
- 2000-2010 годы – период развития интернета, глобализации и новых технологий.

Каждый из этих периодов имеет свои особенности и отражает важные исторические события и социокультурные изменения, которые могут быть

интересны для анализа через призму дневниковых записей. Количественное распределение данных можно увидеть в Таблице 11 и наглядное соотношение классов на рисунке 26.

Таблица 11 – Количественные данные признака «Эпоха создания записи»

| **Название признака** | **Значение признака** | **Количество текстов** |
|---|---|---|
| Эпоха создания записи | 1700-1799 | 5967 |
| | 1800-1899 | 45834 |
| | 1900-1920 | 61136 |
| | 1921-1940 | 66805 |
| | 1941-1945 | 81364 |
| | 1946-1960 | 41657 |
| | 1961-1979 | 60461 |
| | 1980-1999 | 60187 |
| | 2000-2018 | 14367 |

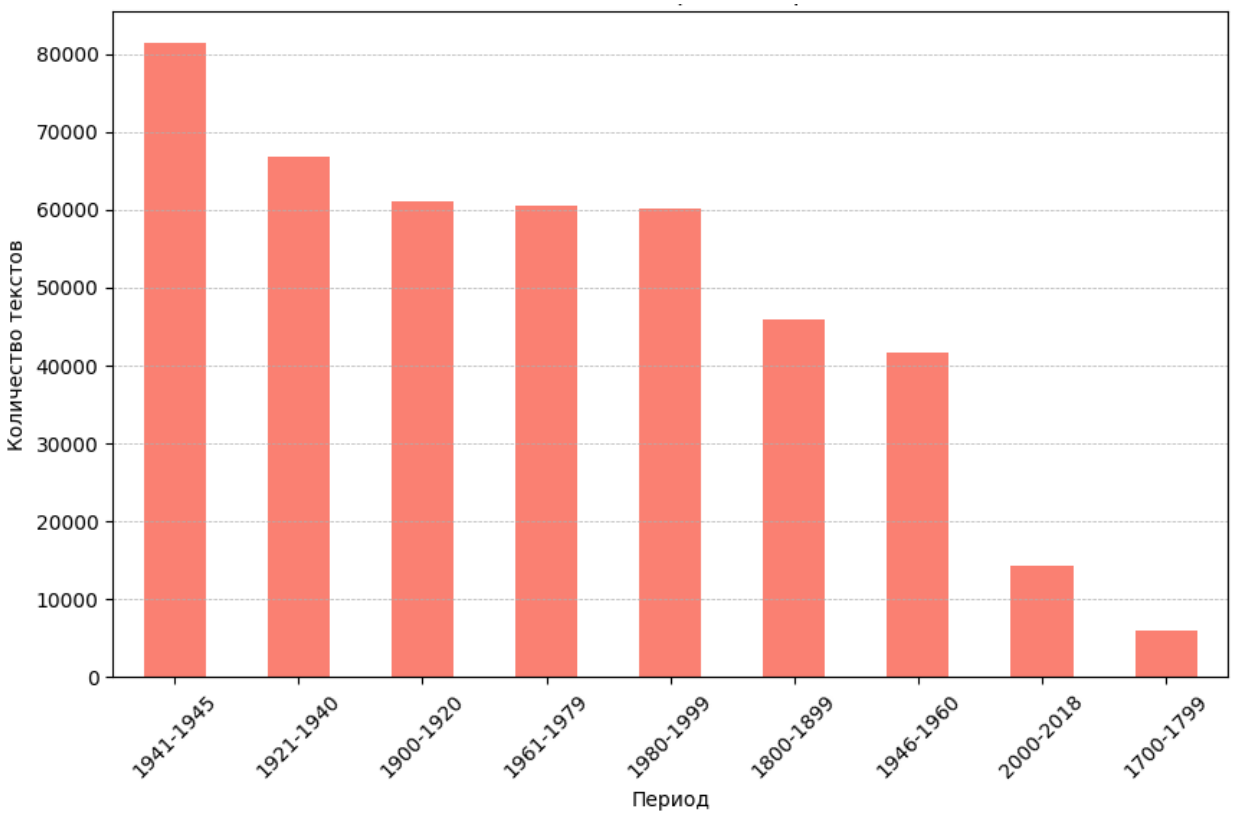


Рисунок 26 – Итоговое распределение текстов по классам периодов создания записи

Модель логистической регрессии показала точность равную 68%, модель наивного байесовского классификатора – 67%. Из матриц ошибок на рисунке 27 видим, что более точно модели предсказывают отнесение текстов к 18 столетию. Слова, которые повлияли на предсказание моделей, и их коэффициенты выглядят следующим образом: *верста (11.58), сей (11.11), князь (10.04), обедать (9.67), пяток (9.50), оный (9.16), ездиль (7.46), понеделка (6.20) (от понеделок – первый день после недели)* и др. Большинство слов можно идентифицировать как характерные для более ранних временных периодов (*верста, сей, оный, понеделка*), вероятнее всего, они указывали на периоды с 1700-х до начала 20 века. Слово «*князь*», например, указывает на повествования, связанные с дворянством и его ролью в обществе. Такое слово будет более характерно для дореволюционной России, указывая на периоды до 1917 года.

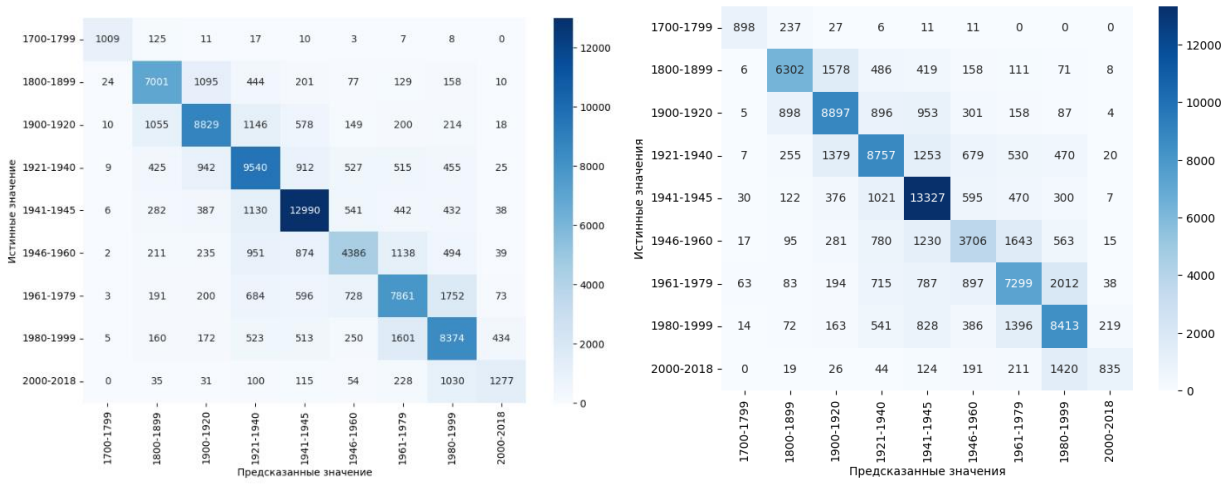


Рисунок 27 – Сравнение матриц ошибок для моделей логистической регрессии и наивного Байесовского классификатора соответственно

Результаты и параметры нейросетевых моделей также можно видеть в таблице 12. Модель с использованием сверточной архитектурой показала точность равную 61%, в сочетании с LSTM значение стало немного выше – 62%. BiLSTM в сочетании LSTM показала наиболее высокий результат равный 71%.

Таблица 12 – Сравнение разных архитектур и параметров в классификации эпохи написания записи

| **Основная модель** | **Параметры ключевого слоя** | **Точность модели** | **Значение ошибки** |
|---|---|---|---|
| LogitRegression | TfidfVectorizer() | 0.7006 | - |
| Multinominal Naive Bayes | CountVectorizer() | 0.6673 | - |
| Conv1D | (filters=64 / 128, kernel_size=3, padding='same', activation='relu') / BatchNormalization(),GlobalMaxPooling1D(), Dropout(0.5) | 0.6120 | 1.0455 |
| LSTM + Conv1D | (128, return_sequences=True) + (256, 3, activation='relu') / SpatialDropout1D(0.4), Dropout(0.4) | 0.6179 | 1.0058 |
| BiLSTM + LSTM | (LSTM(64, return_sequences=True)) + (64 / 64, return_sequences=True) | **0.7095** | 0.4618 |

Из проведенного анализа моделей классификации текстов по эпохе создания записи (Таблица 12) видно, что различные архитектуры нейронных сетей имеют различные уровни точности и ошибок при решении данной задачи. Модель BiLSTM в связке с LSTM показала наилучшие результаты с точностью 71%, что свидетельствует о её способности эффективно определять временной период создания текстовых записей. Эта модель объединяет двунаправленный LSTM и классические LSTM слои, что позволяет учесть как контекст прошлых слов, так и будущих, что важно для анализа последовательных данных.

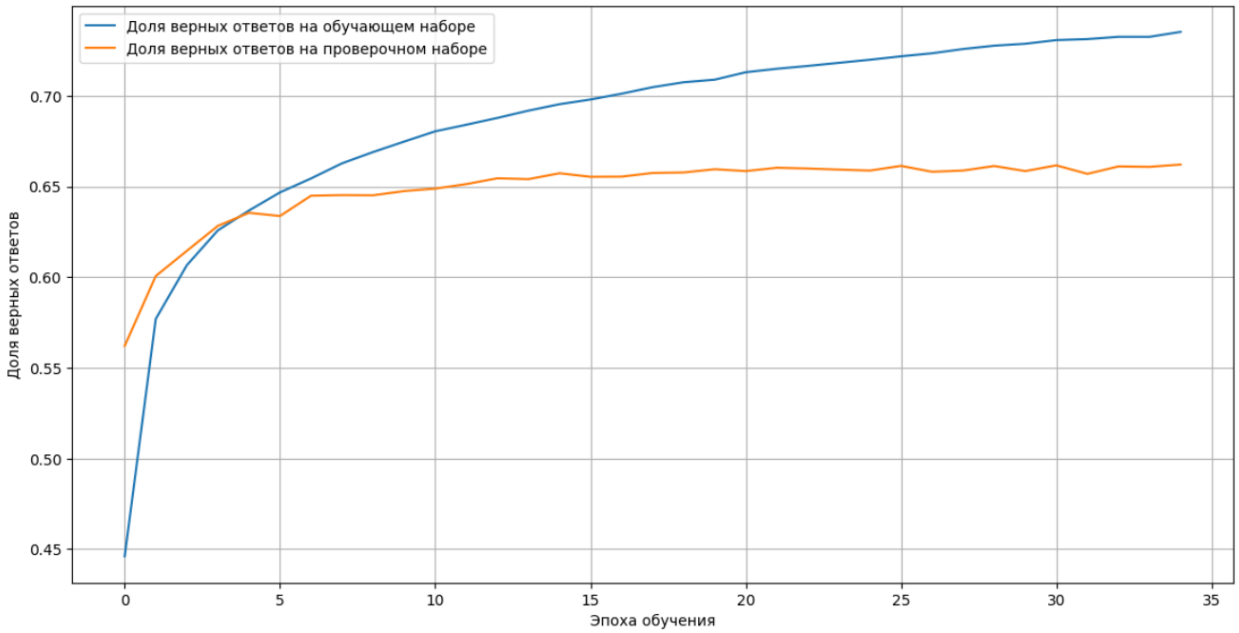


Рисунок 28 – График точности работы сборной модели из BiLSTM и LSTM при классификации эпохи написания записи

**Выводы к главе 3**

После проведения комплексной предобработки (удаление специальных символов, стоп-слов, токенизации, лемматизации, удаления иностранных вкраплений) текстовых данных итоговой корпус составил 2139 дневников с 437230 текстами и 39544413 токенами.

Эксперимент с Bag of Words показал меньшую точность работы модели (72%) по сравнению с использованием Embeddings (83%), несмотря на одинаковые модели тестирования. В процессе формирования обучающей выборки было исследовано влияние разбиения последовательности индексов слов на окна фиксированной длины с заданным шагом для эмбеддингов. Эксперименты показали, что уменьшение длины отрезка текста снижает точность, в то время как увеличение шага разбиения улучшает точность модели.

Среди рассмотренных признаков наиболее популярными для предсказаний являются задачи бинарной классификации пола автора текста и его возраста в то время, как задача предсказания эпохи создания текстов практически не встречается из-за сложности, неоднозначности и отсутствия репрезентативных данных.

Классификация текстов по полу автора представляет собой сложную задачу из-за вариабельности стилей письма, но мужчины и женщины демонстрируют различия в выборе слов, эмоциональной лексике и структуре предложений. Модели, определяющие пол, требуют меньше личных данных, что уменьшает этические риски, и пол легче проверить и подтвердить в данных, что делает эту задачу более доступной для алгоритмов машинного обучения.

Были проведены сравнения различных подходов к классификации текстов по гендеру автора. Модель логистической регрессии, использующая TfidfVectorizer, показала высокую точность в 89%, что связано с эффективным выделением значимых слов для идентификации гендера писателя. Наивный

Байес с CountVectorizer также продемонстрировал хорошие результаты с точностью 83%. Нейросетевой подход с использованием трех различных моделей (Conv1D, LSTM + Conv1D, BiLSTM) показал точность от 83.57% до 86.82%. Дополнительно был разработан модуль с сохранением глаголов в прошедшей форме для обучающих лемматизированных текстов, повысивший точность моделей примерно на 3%.

При предсказании возрастных групп использование модели K-means с TF-IDF и сингулярным разложением позволило выделить три возрастных кластера. Разделение выборки на возрастные группы (18-35 лет, 36-55 лет, 56+ лет) помогает обеспечить баланс данных и учитывать различия в лексике, стиле письма и тематиках, характерных для разных возрастов. Исключение текстов от авторов моложе 17 лет из обучающей выборки помогает избежать коррелированности остальных данных и сбалансировать модель.

Модель логистической регрессии справилась с предсказание 3 возрастных групп лучше остальных моделей, показав значение точности в 71%. Среди моделей глубинного обучения лучший результат в 67% показала комбинация LSTM и BiLSTM.

Задача предсказания времени создания текста сталкивается с рядом сложностей, таких как изменения в языке, недостаточное количество аннотированных данных. Результаты разведочного анализа показали, что большинство текстов относится к 20 веку, что может быть объяснено историческими событиями, развитием технологий печати и особенностями формирования корпуса данных. Наблюдается неравномерное распределение количества текстов по годам в разные века, причем 19 и 21 века имеют более равномерное распределение, чем 18 и 20 века. При разработке модели классификатора важно учитывать логическое разделение данных на классы с учетом исторической значимости периодов, обилия данных и проблем дисбаланса классов. Подготовка данных для модели классификации предсказания времени создания текста может включать три подхода:

разделение по датам, годам или столетиям. Каждый из подходов имеет свои особенности и требования к количеству данных для обучения модели. Был выбран подход деления по столетиям и периодам внутри них: 1800-1899, 1900-1920, 1921-1940, 1941-1945, 1946-1960, 1961-1979, 1980-1999, 2000-2018.

Нейросетевая модель, сочетающая архитектуры BiLSTM и LSTM, показала наиболее высокое значение точности равное 71%. Близкое к этому значение показала также модель логистической регрессии (70%). Наивный байесовский классификатор, модели сверточной сети и ее сочетания с рекуррентной показали результаты ниже – 67%, 61%, 62% соответственно.

Важно отметить, что большинство нейросетевых моделей заканчивали свое обучения во избежание переобучения на эпохах, не достигавших отметки 20, что является относительно низким показателем.

# ЗАКЛЮЧЕНИЕ

Проведенное исследование по подбору методов и архитектур алгоритмов глубинного обучения, направленных на предсказание скрытых атрибутов по тексту, таких как гендер, возраст автора и время создания записи, позволило автоматически выявить важные связи между особенностями языка и демографическими признаками. Дневниковых записи из проекта «Прожито» являются ценным источником данных для исследований в области социолингвистики и машинного обучения. В процессе исследования были выполнены задачи по изучению социолингвистического портрета, сбору и предобработке текстов, сравнительному анализу различных подходов к выделению признаков и архитектур моделей, а также оценке производительности алгоритмов.

Социолингвистический портрет как методологический инструмент, позволяет глубже понять взаимодействие между языком и обществом, а дневниковые записи выступают важным источником для исследования языка в его естественном и «необработанном» виде. Использование машинного и глубинного обучения приносит новые методологические возможности, значимо расширяя возможности анализа текстов и интерпретации социальных различий.

Приведены и проанализированы основные модели машинного обучения, такие как логистическая регрессия и наивный байесовский классификатор, используемые для обучения с учителем, и алгоритм K-means для обучения без учителя. Описаны принципы построения и применения нейронных сетей, включая выбор активационных функций, методы нормализации и стратегии оптимизации, а также эффективные архитектуры для задач, связанных с пониманием контекста и последовательностей в тексте – одномерные сверточные и рекуррентные сети. Рассмотрены различные подходы к векторизации текстов, такие как Bag of Words, TF-IDF и Word Embeddings,

показана важность правильного выбора методов для повышения точности классификации. Подчеркивается значимость глубокого обучения, особенно с использованием одномерных сверточных и рекуррентных сетей, для улучшения результатов классификации текстов. Внедрение современных подходов в конфигурацию нейронных сетей, включая нормализацию, регуляризацию и оптимизацию, является критическим для достижения высоких результатов. Ключевые метрики, такие как точность, полнота и F1-мера, помогают эффективно оценивать модели классификации текстов.

Был собран корпус, содержащий 2 139 дневников, 437 230 текстов, 39 544 413 токенов. Текст с максимальным количеством токенов содержал 9 855 единиц. Средняя длина предобработанных текстов составила 90 токенов.

Эксперименты по методам векторизации текста с влиянием отрезка анализа и шага разбиения на значение точности показали, что уменьшение длины отрезка анализа вело к снижению значений точности, в случае шага разбиения исходного текста на обучающие векторы уменьшение повышало значение точности. Также эксперименты с такими методами как Bag of Words и Word Emdeddings показали преимущество второго алгоритма из-за учета семантической информации, лучшей обработки редких слов и уменьшения размерности входных данных.

Лучшей нейросетевой моделью для предсказания гендера автора текста стала модель, комбинирующая архитектуры LSTM и Conv1D (87%). Значение точности работы логистической регрессии, выбранной базовой, составило 89%. Для задачи бинарной классификации была предложена эвристика с сохранением форм прошедшего времени, основанная на гипотезе поднятия точности в силу введения дневников чаще всего от первого лица, что является следствием сохранения маркера рода в русском языке. С помощью Pymorphy2 был написан морфоанализатор, позволяющий сохранить формы прошедшего времени из исходных текстов в предобработанных текстах. Несмотря на очевидные недостатки подхода, заключающиеся в явлении морфологической

и синтаксической омонимии и в ошибках при автоматическом распознавании форм, точности моделей выросли примерно на 3%, что является хорошим результатом в условиях исходных высоких показателей точности близких к значениям 90%.

При классификации возраста авторов текстов сложной задачей было деление выборки на возрастные группы, поскольку большая часть значений лежала в границах от 30 до 57 лет. Для поиска закономерностей в данных и, в целом, возможности деления на группы была написана модель кластеризации K-means с использованием TF-IDF и понижением размерности через SVD. Выделенные моделью три группы (18-35 лет, 36-55 лет, 56+ лет) использовались в качестве меток классов и были обозначены как молодая возрастная группа, средний зрелый возраст и старшая возрастная группа соответственно.

Модели машинного обучения, предсказывающие возрастные группы авторов, показали 71% точности для логистической регрессии и 66% точности для наивного байесовского классификатора. Обе модели показали хорошие результаты в предсказании молодой возрастной группы (18-35 лет). Среди моделей глубинного обучения лучший результат был достигнут моделью с комбинацией архитектур LSTM и BiLSTM – 67%.

По результатам разведочного анализа и в целях составить сбалансированные классы с сохранением отнесенности текстов к ключевым историческим событиям и репрезентативности выборок было выделено 9 периодов для классификации текстов по времени их создания – 1700-1799, 1800-1899, 1900-1920, 1921-1940, 1941-1945, 1946-1960, 1961-1979, 1980-1999, 2000-2018. В этой задаче нейросетевая модель, сочетающая BiLSTM и LSTM, показала наиболее высокий результат, побив значение точности работы логистической регрессии в 70% на 1%. Значение точностей остальных моделей находилось в пределах от 61% до 67%.

С помощью метода определения коэффициентов весов для слов было установлено, что классификатор гендера автора текста в значительной мере опирается на имена собственные, топонимы и уменьшительно-ласкательные варианты имен, в случае с сохранением в обучающих текстах форм прошедшего времени глаголов такими словами закономерно стали глаголы прошедшего времени. Аналогичный список для возрастных групп не содержал единиц, которые можно было отнести к разным группам. Слова, повлиявшие на предсказание временного периода, содержали множество слов, характерные для более ранних временных рамок – примерно с 1700-х до начала 20 века.

Таким образом, наиболее подходящими моделями для предсказания признаков пола, возрастной группы и временного периода создания текста можно назвать рекуррентные нейронные сети в сочетание с двунаправленными LSTM или CNN1D. Также важно заметить, что в большинстве задач модели классического машинного обучения показали себя лучше, чем нейросетевые модели. Модель логистической регрессии уступила нейросети только в задаче многоклассовой классификации в предсказании 9 временных периодов. Такие результаты, предположительно, связаны с относительной простотой и интерпретируемостью модели логистической регрессии, отлично подходящей для задач с линейными зависимостями между предикторами (словами) и целевыми переменными. Также классические алгоритмы машинного обучения часто обладают преимуществом в задачах с ограниченным объемом данных и хорошо структурированными признаками. Недостаток данных в моделях глубинного обучения часто приводил к ее переобучению. Но задача многоклассовой классификации временных периодов является сложной из-за большого количества классов и неоднородности данных внутри каждого периода. Здесь нейросетевые модели, такие как рекуррентные нейронные сети с двунаправленными LSTM могут лучше справляться благодаря своей способности извлекать сложные

нелинейные зависимости и учитывать контекст слов. Также по результатам можно сделать вывод о методах выделения признаков, нацеленных на частоту встречаемости отдельных слов или на семантику и контекст, – предсказания гендера и возрастных групп по тексту больше зависит от отдельных слов, чем от контекста, в отличие от задачи предсказания временного периода, где важным также являются конструкции слов.

По результатам работы были созданы 17 моделей предсказания гендера, возрастной группы автора текста и временного периода создания текста с достаточно высокими показателями точности работы. Итоги подтверждают, что использование алгоритмов машинного и глубинного обучения позволяет достичь высокой точности в предсказании демографических атрибутов по тексту. Установленные корреляции между языковыми особенностями и демографическими признаками открывают новые возможности для создания точных и эффективных предиктивных систем. Исследование также подчеркивает важность комплексного подхода к построению моделей предсказания социолингвистических признаков, что вносит значительный вклад в развитие данных областей.

В перспективе планируется масштабируемость количества признаков, и количества уникальных значений внутри признака, например, предсказание десятилетий вместо более широких временных периодов, создание более сложных архитектур нейросетей с использованием динамического встраивания эмбеддингов и создание отдельных модулей оптимизации, которые будут направлены на повышение точности предсказания демографических признаков авторов. Эти модули могут включать в себя методы регуляризации, оптимизацию гиперпараметров моделей, а также другие техники, направленные на улучшение обобщающей способности моделей. Также корпус может быть использован для создания моделей распознавания именованных сущностей в силу большого содержания онимов,

тематического моделирования из-за широкого диапазона исторического и культурного контекста и т.д.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1. Батура Т. В. Методы автоматической классификации текстов // Программные продукты и системы. 2017. Т. 30. № 1. С. 85-99.
2. Белл Р. Социолингвистика: цели, методы и проблемы. [Пер. с англ.]. М.: Междунар. отношения. 318 с.
3. Богданова Е. В. Языковые особенности жанра дневника // Филологические науки. Вопросы теории и практики. 2008. №. 1-1. С. 28–33.
4. Бызов А. А. Интеллектуальный анализ текстов в социальных науках // Социология: методология, методы, математическое моделирование. 2019. №. 49. С. 131-160.
5. Галушкин А. И. Нейронные сети: основы теории. / А. И. Галушкин. Изд-во: Горячая линия. Телеком, 2012. 496 с.
6. Гольдберг Й. Нейросетевые методы в обработке естественного языка / пер. с англ. А. А. Слинкина. М. : ДМК Пресс, 2019. 282 с.
7. Евгеньева А. П. Малый академический словарь. М. : Институт русского языка Академии наук СССР. 1957-1984. URL: https://rus-academic-dict.slovaronline.com/ (дата обращения: 13.11.2023).
8. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. / О. Жерон. Вильямс, 2018. 688 с.
9. Зализняк А. А. Дневник: к определению жанра // Новое литературное обозрение. 2010. №. 106. С. 162-180.
10. Казанцев А. А., Прохоров М. В., Худякова П. С. Обзор подходов к классификации текстов актуальными методами // Экономика и качество систем связи. 2021. №. 1 (19). С. 57-67.
11. Кирилина А. В. Гендер и язык. Антология. М. : Языки славянской культуры, 2005. 624 с.
12. Крысин Л. П. Очерки по социолингвистике / Л. П. Крысин. М. : ФЛИНТА, 2021. 360 с.

13. Литературная энциклопедия: Словарь литературных терминов: В 2-х т. / Под ред. Н. Бродского, А. Лаврецкого, Э. Лунина, В. Львова-Рогачевского, М. Розанова, В. Чешихина-Ветринского. М.; Л. : Изд-во Л. Д. Френкель, 1925. URL: https://rus-literary-terms.slovaronline.com/ (дата обращения: 13.11.2023).
14. Лутошкина В. В. и др. Открытый электронный архив эго-документов “Прожито”: сохранение личных историй // Человек, сообщества, государства в социально-гуманитарных исследованиях : Сборник материалов XVIII Всероссийской (с международным участием) научной конференции студентов, магистрантов, аспирантов и молодых ученых / отв. ред. В. В. Расколец. 2023. № 18. С. 92-97.
15. Мельниченко М. А., Тышкевич Н. Б. "Прожито" от рукописи до корпуса: сбор, разметка, анализ дневниковых текстов // Цифровая гуманитаристика: ресурсы, методы, исследования. 2017. С. 134-137.
16. Николаева Т. М. «Социолингвистический портрет» и методы его описания // Русский язык и современность. Проблемы и перспективы развития русистики. Доклады Всесоюзной научной конференции. Часть 2. М. : Наука, 1991. 342 с.
17. Раковская Е. Е. Обзор методов искусственного интеллекта для решения задач классификации текстов // System Analysis and Mathematical Modeling. 2020. Т. 2. №. 4. С. 32-43.
18. Сбоев А. Г. и др. Модель нейронной сети для включения синтаксической структуры предложения в задачу классификации пола автора русского текст // Вестник НИЯУ МИФИ. 2023. Т. 8. №. 6. С. 569-576.
19. Филатова Н. М. Подходы к изучению эго-документов в современной исторической науке в свете" лингвистического поворота" // Документ и "документальное" в славянских культурах: между подлинным и мнимым. 2018. С. 24-40.

20. Шалев-Шварц Ш., Бен-Давид Ш. Идеи машинного обучения: от теории к алгоритмам / пер. с англ. А. А. Слинкина. М. : ДМК Пресс, 2019. 436 с.
21. Швейцер А. Д. Введение в социолингвистику. / А. Д. Швейцер, Л. Б. Никольский. М., 1978. 216 с.
22. Ahmed M., Seraj R., Islam S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation // Electronics. 2020. Т. 9. №. 8. 12 p.
23. Alotaibi F. M. Classifying text-based emotions using logistic regression // VAWKUM Transactions on Computer Sciences. 2019. Т. 7. №. 1. P. 31-37.
24. Anandarajan M., Hill C., Nelson T. Classification Analysis: Machine Learning Applied to Text // Practical Text Analytics: Maximizing the Value of Text Data. Switzerland : Springer, 2019. P. 131-149.
25. Ashraf M. A., Nawab R. A., Nie F. A study of deep learning methods for same-genre and cross-genre author profiling // Journal of Intelligent & Fuzzy Systems. 2020. Т. 39. №. 2. P. 2353-2363.
26. Asogwa D. C. et al. Development of a machine learning algorithm to predict author’s age from text. International Journal of Research, 2019. Т. 7. №. 10. P. 380-389.
27. Bird S., Klein E., Loper E. Natural language processing with Python: analyzing text with the natural language toolkit. "O'Reilly Media, Inc.", 2009. 463 p.
28. Brownlee J. Deep learning for natural language processing // Machine Learning Mystery, Vermont, Australia. 2017. 414 p.
29. Cheng N., Chandramouli R., Subbalakshmi K. P. Author gender identification from text // Digital investigation. 2011. Т. 8. №. 1. P. 78-88.
30. Dogra V. et al. A complete process of text classification system using state-of-the-art NLP models // Computational Intelligence and Neuroscience. 2022. Т. 2022. 26 p.

31. Elspaß S. The use of private letters and diaries in sociolinguistic investigation // The handbook of historical sociolinguistics. 2012. P. 156-169.
32. Fishman J. Preface. Advances in the Sociology of Language. Paris : The Hague, 1971. T. 1.
33. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. The MIT Press, 2016. 777 p.
34. Hawari M. A. A., Khodra M. L. Predicting latent attributes by extracting lexical and sociolinguistics features from user tweets // 2014 International Conference on Data and Software Engineering (ICODSE). IEEE, 2014. P. 1-5.
35. Hobson L. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python / L. Hobson, M. Dyshel, H. Hapke. Manning; First Edition, 2019. 544 p.
36. Howard J., Ruder S. Universal language model fine-tuning for text classification // arXiv preprint arXiv:1801.06146. 2018. 8 p.
37. Huang L. et al. Normalization techniques in training dnns: Methodology, analysis and application // IEEE transactions on pattern analysis and machine intelligence. 2023. T. 45. №. 8. P. 10173-10196.
38. Jacovi A., Shalom O. S., Goldberg Y. Understanding convolutional neural networks for text classification // arXiv preprint arXiv:1809.08037. 2018. 10 p.
39. Jurafsky D. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition / D. Jurafsky, J. H. Martin. New Jersey, 2008. 558 p.
40. Kamath C. N., Bukhari S. S., Dengel A. A Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification. Proceedings of the ACM Symposium on Document Engineering 2018. DocEng'18. 11 p.

41. Kingma D. P., Ba J. Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. 2014. 15 p.
42. Kowsari K. et al. Text Classification Algorithms: A Survey // arXiv e-prints. 2019. 68 p.
43. Labov W. The Study of Language in its Social Context. / W. Labov. In «Studium Generale». T. 23. 1970. P. 30-87.
44. Li H. Deep learning for natural language processing: advantages and challenges // National Science Review. 2018. T. 5. №. 1. P. 24-26.
45. Li Q. et al. A survey on text classification: From traditional to deep learning // ACM Transactions on Intelligent Systems and Technology (TIST). 2022. T. 13. №. 2. P. 1-41.
46. Minaee S. et al. Deep learning-based text classification: a comprehensive review // ACM computing surveys (CSUR). 2021. T. 54. №. 3. P. 1-40.
47. Mohri M., Rostamizadeh A., Talwalkar A. Foundations of machine learning. MIT press, 2018. 462 p.
48. Morgan-Lopez A. A. et al. Predicting age groups of Twitter users based on language and metadata features // PloS one. 2017. T. 12. №. 8. 12 p.
49. Nguyen D. et al. Computational sociolinguistics: A survey // Computational linguistics. 2016. T. 42. №. 3. P. 537-593.
50. Nwankpa C. et al. Activation functions: Comparison of trends in practice and research for deep learning // arXiv preprint arXiv:1811.03378. 2018. 30 p.
51. Polyakov I. V. et al. Text classification problem and features set. // Vestn. NGU. Ser.: Informatsionnye tekhnologii [Novosibirsk State Univ. Journ. of Information Technologies]. 2015, T. 13. №. 2. P. 55-63.
52. Qader W. A., Ameen M. M., Ahmed B. I. An overview of bag of words; importance, implementation, applications, and challenges // 2019 international engineering conference (IEC). IEEE, 2019. P. 200-204.

53. Raghunadha Reddy Dr. T. et al. A Deep Learning Approach for Author Profiling using Word Embeddings // International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2023. T. 11, № V. 8 p.
54. Raghunadha Reddy T. A Survey on Author Profiling Techniques / Reddy T. Raghunadha, Vardhan B. Vishnu, Reddy P. Vijayapal // International Journal of Applied Engineering Research. 2016. T. 11. №. 5. P. 3092-3102.
55. Santoshi K., Archana U., Priyanka D. Text Classification Using Machine Learning Techniques // Specialusis Ugdymas. 2022. T. 1. №. 43. P. 8108-8117.
56. Subbulakshmi T. et al. Analysis of Traditional and Deep Learning Architectures in NLP: Towards Optimal Solutions // 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC). IEEE, 2023. P. 760-765.
57. Suneera C.M., Prakash J. Performance analysis of machine learning and deep learning models for text classification // 2020 IEEE 17th India council international conference (INDICON). IEEE, 2020. 6 p.
58. Thakur V., Tickoo A. Text2Gender: A Deep Learning Architecture for Analysis of Blogger's Age and Gender // arXiv preprint arXiv:2305.08633. 2023. 10 p.
59. Tran Q. et al. Comparing the Robustness of Classical and Deep Learning Techniques for Text Classification // 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022. P. 1-10.
60. Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin. Learning from Data: A Short Course. AMLBook, 2012. 213 p.

## ЭЛЕКТРОННЫЕ РЕСУРСЫ


1. Python. URL: https://www.python.org/
2. MyStem. URL: https://yandex.ru/dev/mystem/
3. NLTK. URL: https://www.nltk.org/
4. BeautifulSoup. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
5. Langdetect. URL: https://pypi.org/project/langdetect/
6. Google Collab. URL: https://colab.research.google.com/
7. Keras. URL: https://keras.io/
8. Scikit-learn. URL: https://scikit-learn.org/stable/
9. Kaggle. URL: https://www.kaggle.com/code
10. Pymorphy2. URL: https://pymorphy2.readthedocs.io/en/stable/

# ПРИЛОЖЕНИЕ А

## Примеры данных из выгруженного корпуса

Таблица А.1 – Пример данных из таблицы *persons*

| id | birth | death | approx_dates | sex |
|---|---|---|---|---|
| 4 | 1890-05-25 | 1938-02-10 | 0 | 1 |
| 5 | 1906-03-12 | 1989-12-06 | 0 | 1 |
| 6 | 1892-11-29 | 1953-08-20 | 0 | 1 |
| 11 | 1911-01-03 | 1997-10-06 | 0 | 1 |
| 12 | 1891-01-01 | 1959-12-31 | 1 | 1 |
| 13 | 1880-11-28 | 1921-08-07 | 0 | 1 |
| 14 | 1880-01-01 | 1947-01-01 | 1 | 1 |
| 15 | 1835-01-01 | 1912-12-31 | 1 | 0 |

Таблица А.2 – Пример данных из таблицы *diaries*

| id | person | premier |
|---|---|---|
| 3 | 4 | 0 |
| 4 | 5 | 0 |
| 5 | 6 | 0 |
| 6 | 7 | 0 |
| 8 | 11 | 0 |
| 9 | 12 | 0 |
| 10 | 13 | 0 |
| 11 | 14 | 0 |
| 12 | 15 | 0 |

Таблица А.3 – Пример данных из таблицы *notes*

| **id** | **diary** | **text** | **date** | **dateTop** | **notDated** |
|---|---|---|---|---|---|
| 693 | 3 | Сейчас бьет на башенных часах 12. Полночь. Жена уже спит. Дети — давно. Делал сегодня двухчасовую прогулку, поэтому не так угнетен ежедневными заботами (они во время | 28.09.1932 | 0000-00-00 | 0.0 |

| | | | | | |
|---|---|---|---|---|---|
| | | прогулки будто ниспадали с меня), вместо них вырисовывались главные заботы. 1) Определенно поставить вопрос о перемещении из Праги. 2) О невозможности жить на то, что получаю. 3) Продолжать писать роман «Правда» (о провокаторе Малиновском). Одолевают «гады». Недавно был из Карлсбада ГИ.П. Это просто беда! Ему нужна машина в аренду. Но он капризен как больной ребенок. … Но слова из-под моего контроля вырываются целые дни, а хочется организовать их так, чтобы было известное количество времени для литературной работы. Я сильно отстал от нее (сегодня написал маленький рассказик «Нога» из серии трамвайных и начал рассказы октябрьские). | | | |
| 694 | 3 | Читал Шаляпина — воспоминания. Оказывается, историк Ключевский был, по словам Шаляпина, хороший актер. Вызвал корреспондента ТАСС в связи со статьей о налогах с торгпредства. Пришел | 06.10.1932 | 0000-00-00 | 0.0 |

| | | | | | |
|---|---|---|---|---|---|
| | | Лингарт с докладом о делах, потом Богомолова и Крачевский. В 101/2 пошел в баню… Прощай еще один мой безлитературный день! | | | |
| 695 | 3 | При социализме заводы суть храмы и центры человеческого жития. Они при капитализме были в загоне, ибо рассматривались как аппарат, обслуживающий жизнь, а не творческий. Теперь это творческие узлы. | 20.10.1932 | 0000-00-00 | 0.0 |
| 699 | 3 | На торжественном заседании в Большом театре. В президиуме — политбюро. Воодушевление. Нам (мне, Пискареву и другим «октябристам») не достало места. Попросил охраняющих помочь. Один из них мотивирует отказ: «Товарищи, больше стульев нельзя сюда ставить (хотя свободного места много). Вы плановость нарушите». Мы перешли на другую сторону зала. Здесь нам не только не отказали в стульях, но сами их принесли. | 06.11.1932 | 0000-00-00 | 0.0 |
| 701 | 3 | День провели в отдыхе. | 08.11.1932 | 0000-00-00 | 0.0 |

## ПРИЛОЖЕНИЕ Б

### Вспомогательные функции для создания общего корпуса текстов

```
# Общий датасет, куда будем добавлять информацию
final_dataset = pd.DataFrame(columns=['diary', 'sex',
'birth'])

unique_diary_indexes =
notes['diary'].loc[notes['diary'].apply(lambda x:
str(x).isnumeric())].unique()

for diary_index in unique_diary_indexes:
    try:
        diary_index = int(diary_index)

        # Получаем значение person из diaries.csv
        person_value = diaries.loc[diaries['id'] ==
diary_index, 'person'].values[0]

        # Получаем sex и birth из persons.csv если
person_info не пустое
        person_info = persons.loc[persons['id'] ==
person_value, ['sex', 'birth']]

        final_dataset = pd.concat([final_dataset,
pd.DataFrame({'diary': [diary_index], 'sex':
[person_info['sex'].values[0]], 'birth':
[person_info['birth'].values[0]]})], ignore_index=True)

    except IndexError:
```

```
        print(f"IndexError возникает для индекса
{diary_index}. Пропускаем эту строку.")

final_dataset =
final_dataset.rename(columns={'diary_num': 'diary'})

notes = notes[pd.to_numeric(notes['diary'],
errors='coerce').notna()]
notes['diary'] = notes['diary'].astype(int)

for index, row in notes.iterrows():
    diary_value = row['diary']
    if isinstance(diary_value, (int, float)):
        matching_row =
final_dataset[final_dataset['diary'] == diary_value]
        if not matching_row.empty:
            sex_value = matching_row['sex'].values[0]
            birth_value =
matching_row['birth'].values[0]
            notes.loc[index, 'sex'] = sex_value
            notes.loc[index, 'birth'] = birth_value
    else:
        print(f"Значение в строке {index} столбца
'diary' не является числовым.")
```

## ПРИЛОЖЕНИЕ В

### Вспомогательные функции для предобработки текстов

```
def remove_html_tags(text):
  """Функция для удаления HTML-тегов"""
  soup = BeautifulSoup(text, 'html.parser')
  return soup.get_text()

def remove_punctuation(text):
  """Функция удаления знаков пунктуации"""
  return "".join([ch if ch not in string.punctuation
else ' ' for ch in text])

def remove_numbers(text):
  """Функция удаления чисел"""
  return "".join([i if not i.isdigit() else ' ' for i
in text])

def remove_multiple_spaces(text):
  """Функция удаления множественных пробелов"""
  return re.sub(r'\s+', ' ', text, flags=re.I)

def remove_single_letters(text):
  """Функция для удаления отдельно стоящих букв"""
  return re.sub(r'\b\w\b', '', text)

def remove_non_cyrillic_words(text):
  """Функция для удаления слов с иностранными буквами
кроме кириллицы"""
  return re.sub(r'\b(?![А-Яа-яЁё])\w+\b', '', text)
```

```
def remove_roman_numerals(text):
  """Функция удаления римских цифр"""
  return re.sub(r'\b[ivxlcdmIVXLCDMxxiii]+\b', '',
text)

def remove_word_with_i(text):
  """Функция для удаления слов с буквой 'i'"""
  words = text.split()
  return ' '.join([word for word in words if 'i' not in
word.lower()])

russian_stopwords = stopwords.words("russian")
russian_stopwords.extend(['…', '«', '»', '...', '–',
''])

def lemmatize_text(text):
    tokens = mystem.lemmatize(text)
    tokens = [token for token in tokens if token not in
russian_stopwords and token != " "]
    text = " ".join(tokens)
    return text

from langdetect import detect, LangDetectException

def detect_language(text):
    try:
        if pd.isna(text):
            return 'undefined', text # Если текст
является NaN, возвращаем 'undefined'
```

```
        return detect(text)
    except LangDetectException:
        return 'error' # Если язык не удается
определить
```

## ПРИЛОЖЕНИЕ Г

### Вспомогательные функции для формирования векторных представлений

```
def getSetFromIndexes(wordIndexes, xLen, step):
"""
Формирование обучающей выборки по листу индексов слов /
Разделение на короткие векторы
wordIndexes: последовательность индексов
xLen: длина окна
step: смещение (шаг окна)
"""
xSample = []
wordsLen = len(wordIndexes) # количество слов
index = 0 # начальный индекс
while (index + xLen <= wordsLen):
# проверяем, что индексы слов находятся в допустимом
диапазоне
window = wordIndexes[index:index+xLen]
window = [idx if idx < 1000 else 0 for idx in window]
xSample.append(window)
index += step # смещаемся вперед на step
return xSample
def createSetsMultiClasses(wordIndexes, xLen, step):
""" Формирование обучающей и проверочной выборки из
двух листов индексов от двух классов
wordIndexes: последовательность индексов
xLen: длина окна
step: смещение (шаг окна)
```

```
Функция возвращает выборку и соответствующие векторы
классов
"""
# для каждого класса создаем обучающую/проверочную
выборку из индексов
nClasses = len(wordIndexes)
classesXSamples = [] # список размером: кол-во
классов*кол-во окон в тексте*длину окна
for wI in wordIndexes:
classesXSamples.append(getSetFromIndexes(wI, xLen,
step)) # добавляем в список текст индексов, разбитый на
кол-во окон*длину окна
# общий xSamples
xSamples = [] # список размером: суммарное кол-во окон
во всех текстах*длину окна
ySamples = [] # список размером: суммарное кол-во окон
во всех текстах*вектор длиной 2
for t in range(nClasses):
xT = classesXSamples[t]
for i in range(len(xT)):
xSamples.append(xT[i])
ySamples.append(utils.to_categorical(t, nClasses))
xSamples = np.array(xSamples)
ySamples = np.array(ySamples)
return (xSamples, ySamples)
```

## ПРИЛОЖЕНИЕ Д

### Функция для поиска глаголов прошедшего времени с использованием морфонализатора

```
# Создание анализатора
morph = pymorphy2.MorphAnalyzer()

# Функция для поиска глаголов прошедшего времени
def find_past_tense_verbs(text):
    words = text.split()  # Разделение текста на слова
    past_verbs = []
    for word in words:
        parsed_word = morph.parse(word)[0]  # 
Морфологический анализ слова
        if 'VERB' in parsed_word.tag and 'past' in 
parsed_word.tag:  # Проверка на глагол прошедшего 
времени
            past_verbs.append(word)
    return ' '.join(past_verbs)

# Применение функции ко всему столбцу 'text' и 
соединение с 'text_preprocessing'
tqdm.pandas(desc="Processing rows")
data['text_preprocessing_past_verbs'] = 
data.progress_apply(lambda row: 
str(row['text_preprocessing']) + ' ' + 
find_past_tense_verbs(row['text']) if 
isinstance(row['text_preprocessing'], str) else 
find_past_tense_verbs(row['text']), axis=1)
```