Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 20.Б08-мм

Рабош Артём Леонидович

Определение ненаучного стиля текста

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
ассистент Чернышев Г.А.

Санкт-Петербург
2023

Оглавление

Введение

Написание качественного текста дипломной работы студента является важным аспектом его профессиональной подготовки, так как это позволяет продемонстрировать уровень его знаний, навыков и компетенций. Качественный текст должен быть логически последовательным, отвечать требованиям научного стиля, а также не содержать грамматических или пунктуационных ошибок. Грамотно написанный текст дипломной работы может быть преимуществом при приеме на работу. Он позволяет оценить уровень подготовки студента и его способность к аналитической и научной работе, что может повлиять на решение работодателя при выборе кандидата на должность.

Проверка дипломных работ является важным процессом, который гарантирует, что содержание и качество работы соответствуют академическим стандартам и требованиям. Кроме того проверка является сложным процессом, связанным с рядом трудностей, которые могут привести к необходимости дополнительной проверки. В отличии от проверки содержания работы, проверка формальных требований требует значительно меньших компетенций. В связи с этим рассматривается возможность автоматизации проверки формальных требований. Одним из таких требований является поддержание научного стиля в работе.

Mundane Assignment Police (MAP) — веб-приложение,написанное на языке Kotlin, которое автоматизирует проверку текстов учебных практик и ВКР. В данный момент MAP поддерживает нахождение различных ошибок оформления в PDF-файлах. В рамках этой работы предлагается расширить функционал сервиса и разработать алгоритм, позволяющий проверять тексты выпускных квалификационных работ на соответствие научному стилю.

1. Постановка задачи

Целью работы является исследование возможности применения различных методов обработки текста для задачи определения ненаучного стиля. Для достижения этой цели были сформулированы следующие задачи:

1. Провести обзор алгоритмов обработки текстов на естественном языке для задачи определения научного стиля текста.
2. Собрать и подготовить данные для тестирования алгоритмов.
3. Выбрать и реализовать алгоритмы классификации текстов.
4. Создать программную среду для апробации создаваемых алгоритмов.
5. Сравнить полученные алгоритмы.

2. Обзор

2.1. Языковые особенности ненаучного стиля текста

В академическом мире научный стиль является одним из основных стилей, используемых для написания научных работ, включая дипломные работы студентов. Он отличается от ненаучного стиля, который часто используется в повседневном общении и других сферах. В данном разделе рассмотрим основные отличия между научным и ненаучным стилем.

2.2. Морфологические особенности

Синтаксические и морфологические особенности научного стиля текста были подробно рассмотрены в ряде научных работ [7, 9, 10].

Обычно ненаучный текст отличается от научного следующими особенностями:

- Использование сокращений и аббревиатур: в ненаучных текстах часто используются сокращения и аббревиатуры, которые не являются стандартными в научном стиле.
- Использование сленга и неформальной лексики: ненаучные тексты могут содержать неформальные выражения, сленг и жаргон, которые не соответствуют нормам литературного языка.
- Нарушения грамматических правил: в ненаучных текстах могут встречаться нарушения грамматических правил, такие как неправильное использование падежей, времен и форм слов.
- Стилистические особенности: в ненаучных текстах могут присутствовать различные стилистические приемы, такие как метафоры, эпитеты, повторы и т.д., которые не используются в научном стиле.

- Использование диалогов и прямой речи: в ненаучных текстах может быть много диалогов и прямой речи, что не характерно для научного стиля.

2.3. Синтаксические особенности

Из синтаксических особенностей ненаучного текста можно выделить:

- Большое количество простых и сложносочиненных предложений: в ненаучных текстах обычно встречается много простых и сложносочиненных предложений, которые используются для передачи информации простым и понятным языком.

- Неполные предложения: ненаучные тексты могут содержать неполные предложения, которые не являются полностью грамматически правильными, но используются для эмоциональной или эффектной передачи информации.

- Использование вопросительных и восклицательных предложений: в ненаучных текстах можно часто встретить вопросительные и восклицательные предложения, которые помогают передать эмоциональный окрас.

- Использование перифраза: в ненаучных текстах может использоваться перифраз — замена слова или фразы более простым или понятным выражением.

- Отсутствие строгой логики: в ненаучных текстах может быть меньше строгой логической последовательности, чем в научных, и можно встретить более свободное расположение аргументов и фактов.

2.4. Алгоритмы обработки текста

В данном разделе рассматриваются работы, решающие задачу определения ненаучного стиля текста

2.4.1. Статистический анализ текста

Проблема определения стилистической направленности текста, включая научный стиль, была затронута в магистерской диссертации Дубовик Анны Романовны [8]. В работе было проведено исследование, направленное на разработку алгоритма автоматического определения стиля текстов. Были использованы методы классического машинного обучения и статистического анализа текстов. В результате экспериментов было показано, что метрика точность (Precision) для классификации научных текстов составила 0.95, что говорит о высокой точности разработанного алгоритма.

Автор выделил множество статистических характеристик текстов различной функциональной направленности. Однако, статистический анализ не подходит для анализа отдельных предложений, так как не учитывает контекст, в котором они используются, и не обращает внимания на семантику и грамматические конструкции. Более того, статистический анализ не учитывает структуру предложения и не может распознать различные части речи, что делает его неэффективным для оценки качества и стиля текста на уровне отдельных предложений.

2.4.2. Large Lanugage Models

Large Language Models (LLM) — это глубокие нейронные сети, обученные на огромных объемах текстовых данных с целью понимания естественного языка и выполнения различных задач обработки текста.

Некоторые из наиболее известных LLM включают в себя модели, такие как GPT-3 (Generative Pretrained Transformer 3) [5] от OpenAI, T5 (Text-to-Text Transfer Transformer) от Google [4], и BERT (Bidirectional Encoder Representations from Transformers) [1]. Эти модели обучены на огромных объемах текстов и могут эффективно решать задачи обработки текста, такие как машинный перевод, генерация текста, классификация.

LLM имеют огромный потенциал для автоматической обработки и анализа естественного языка.

2.4.3. FormalityBERT

Формальность — одно из требований научного текста. В английском языке задаче определения формальности предложения посвящено множество статей, многие из которых используют в качестве предсказательной модели нейросетевые методы. Большинство этих моделей используют датасет GYAFC. Он состоит из 110000 пар формальной и неформальной версий предложения. Обученная на данном датасете модель Char-BiLSTM показывает значение 0.88 F1-score [3] на тестовой выборке. Однако для русского языка не существует подобных датасетов. Тем не менее существует обученная на GYAFC мультиязычная модель, которую можно использовать для определении формальности текстов на русском языке. Данная модель работает для отдельных предложений, но требует больших вычислительных мощностей.

2.4.4. Выводы

В рамках проведенного обзора не удалось обнаружить работ, которые бы полноценно решали задачу определения ненаучного стиля. Алгоритм представленный в работе Дубовик Анны Романовны [8], не позволяет классифицировать отдельные предложения, к тому же он был предложен, до появления больших языковых моделей, которые на данный момент показывают лучшие результаты [2], чем методы классического машинного обучения. Обширный набор синтаксических и морфологических особенностей создает трудности в написании правил, по которым можно классифицировать научный стиль. При этом, подходы нацеленные на другую ближайшую задачу — определение формальности, не могут быть напрямую переиспользованы для решения этой. Это объясняется тем, что формальность является лишь одним из требований научного стиля. Всё это поднимает вопрос об усовершенствовании метода классификации научных текстов, который позволит рассматривать отдельные предложения, без использования множества правил.

3. Метод

В данной части работы приводится описание разработанной системы, некоторых её особенностей и проблем, которые они решают.

3.1. Датасет

Для задачи классификации научного текста на русском языке не существует открытого размеченного набора данных. Поэтому для проведения экспериментов потребовалось собрать и разметить данные вручную. Было выбрано два источника данных: курсовые и выпускные квалификационные работы студентов кафедры системного программирования СПбГУ, а также статьи с сайта Habr.

Для сбора и обработки данных был использован язык Python, а также библиотек Selenium и BeautifulSoup4.

3.1.1. Работы студентов

Работы студентов доступны на сайте se.math.spbu.ru в формате PDF. Было загружено 1064 работы. Для извлечения текста из полученных файлов было опробовано несколько библиотек для обработки PDF-файлов (PyPDF2, PDFminer). При использовании указанных библиотек наблюдалось некачественное форматирование и неправильное распознавание текстовых данных. В связи с этим было принято решение о применении сервиса MAP для разбора текстовых материалов. Итоговое число текстов составило 419. Полученные текста были разделены на предложения с помощью регулярных выражений и библиотеки Natasha.

3.1.2. Habr

Хабр (англ. “Habr”) — это интернет-ресурс на русском языке, предназначенный для публикации и обмена информацией в области информационных технологий, программирования, науки и бизнеса. Для данной работы были выгружены текста и метаданные (дата, категории) всех статей с ресурса Хабр.

Итого было получено 39 тысяч предложений из работ студентов, а также 260 тысяч предложений с хабра.

3.2. Анализ данных

Для анализа случайным образом были выбраны по 10 тысяч предложений из Хабра и работ студентов соответственно.

3.2.1. Формальность

Для оценки уровня формальности предложений была применена предобученная модель XLMR-Formality. В ходе анализа было обнаружено наличие некоторой корреляции между оценками формальности и происхождением текста. Работы студентов ожидаемо получили лучшие оценки, в то время как статьи на Хабре меньшие.

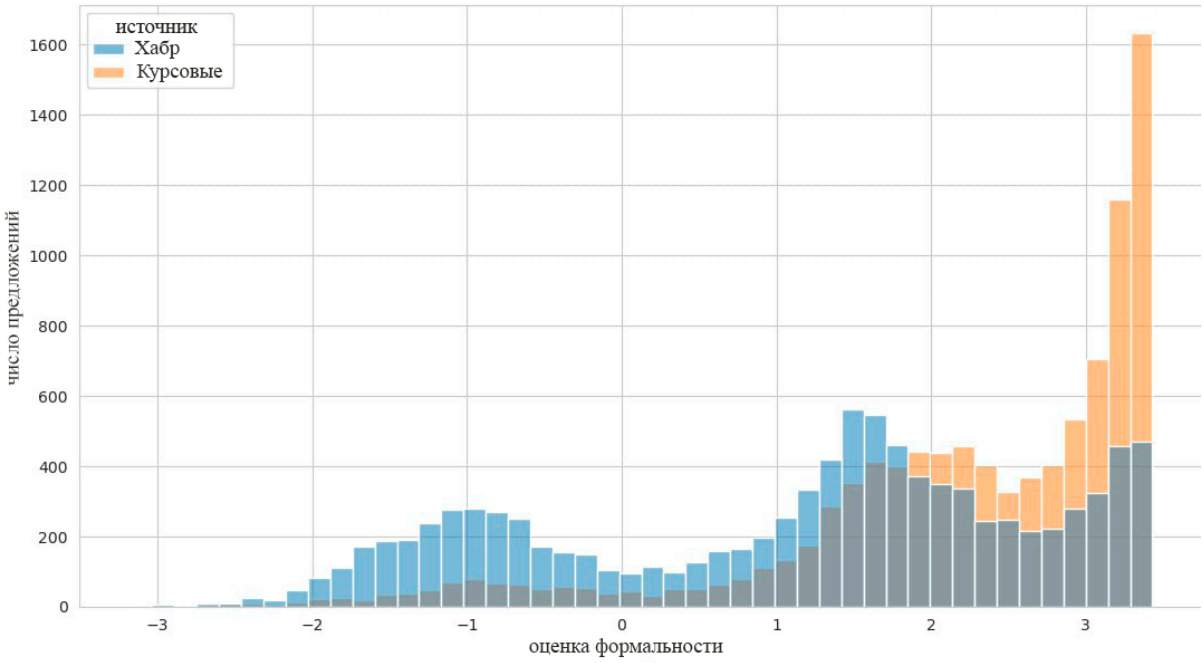

Рис. 1: Распределение оценок формальности

3.2.2. Эмоциональная окраска

Одним из требований научного текста является его нейтральная эмоциональная окраска. В рамках проведенного исследования были получены оценки тональности предложений при помощи предобученной на датасете Rusentiment модели ruBERT$_{\text{BASE}}$ от DeepPavlov. Анализ данных показал, что, как и ожидалось, практически все предложения из работ студентов имеют нейтральную эмоциональную окраску. Однако статьи на Хабре показали лишь небольшое количество эмоциональных предложений.

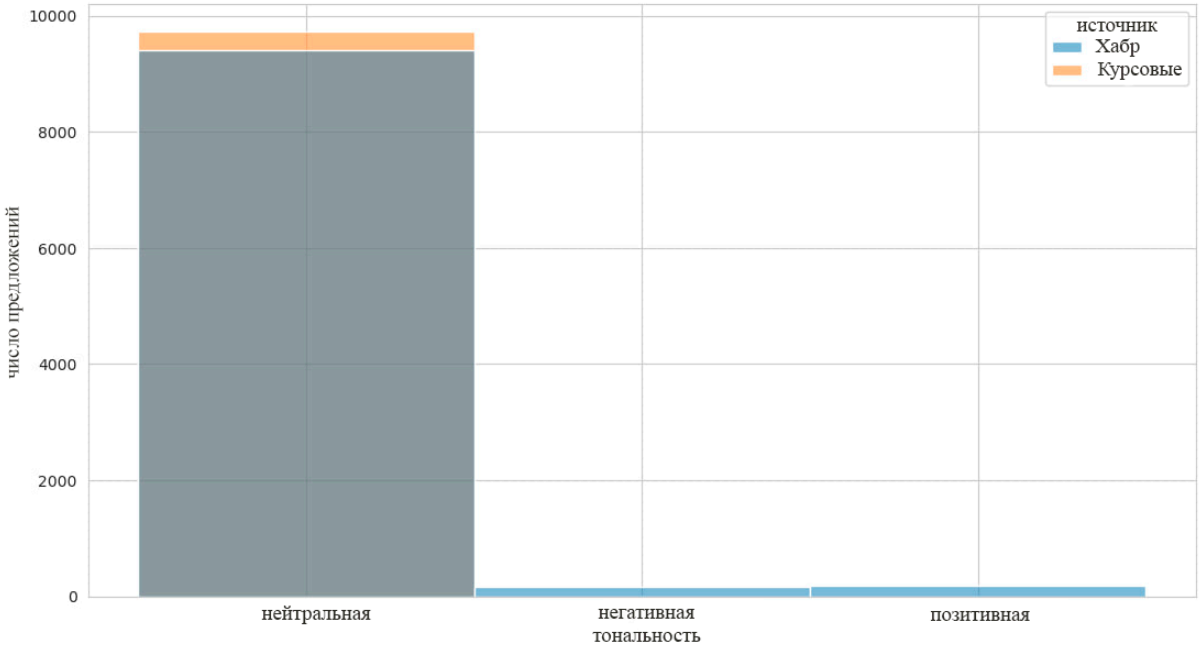

Рис. 2: Распределение эмоциональности

3.2.3. Морфология

Также важным требованием является обезличенность выражений. Предложение не должны содержать личных местоимений, а также других слов, связанных с первым лицом. Для анализа был использован морфологический анализатор Natasha. Среди 42195 предложений из работ студентов 82 предложения нарушали это правило.

3.3. Выбор метода

При выборе метода определения было несколько альтернатив:

1. Создать строгие правила, по которым можно будет находить ошибки в тексте.

2. Использовать машинное обучения для классификации.

Ручное создание правил может привести к более точной классификации, чем использование машинного обучения. Однако поддержание актуальности и эффективности этих правил требует постоянного внимания и усилий со стороны специалиста. Поэтому было выбрано машинное обучение.

3.4. Инфраструктура для обучения и оценки моделей

Для обучения моделей выбран популярный фреймворк PyTorch, как более гибкий и простой для понимания. Для работы с предобученными моделями использовалась библиотека HuggingFace Transformers.

3.5. Выбор и обучение моделей

В рамках данной работы было проведено дообучение существующих языковых моделей для задачи классификации, а также сравнение качества классификации полученных моделей моделей. Для обучения моделей использовался графический процессор (GPU) NVIDIA GeForce GTX 1080, для оптимизации параметров использовался алгоритм AdamW на основе градиентного спуска.

Ниже представлены модели, выбранные для задачи:

1. multilingual $\text{BERT}_{\text{BASE}}$ — оригинальный BERT [1], обученные на текстах из Википедии на 100 самых популярных языках.

2. DeepPavlov $\text{ruBERT}_{\text{BASE}}$ — дообученная на новостях на русском языке multilingual $\text{BERT}_{\text{BASE}}$.

3. sberbank-ai ruBERT$_{\text{BASE}}$ — обучена на 30 GB русского текста среди которого: Википедия, новости, часть корпуса Taiga [6] и немного книг.

4. cointegrated ruBERT$_{\text{tiny}}$ — основанный на BERT энкодер для русского языка. Оптимизирован для работы на CPU.

Таблица 1: Параметры моделей

Модель	Размер	Время инференса (CPU)	Время инференса (GPU)
multilingual BERT$_{\text{BASE}}$	678.47MB	321 ± 53 ms	25 ± 8ms
DeepPavlov ruBERT$_{\text{BASE}}$	678.47MB	374 ± 28 ms	29 ± 11 ms
sberbank-ai ruBERT$_{\text{BASE}}$	680.20MB	449 ± 90 ms	15 ± 2 ms
cointegrated ruBERT$_{\text{tiny}}$	44.96MB	19 ± 6 ms	7 ± 3 ms

Датасет был разбит на обучающую и валидационную выборку следующим образом: последние 30 работ студентов и последние 300 статей были отнесены к валидационной выборке, остальные — к обучающей.

Ввиду большого количества данных, вместо полной ручной разметки, было решено разметить следующим образом: работы студентов признаются научным, если в них нет морфологических ошибок, а оценка эмоциональности — нейтральная. Остальные предложения отнесены к ненаучному стилю. Также согласно этой разметке ненаучных предложений оказалось больше. Несбалансированность классов может привести к переобучению, поэтому для балансировки были случайно удалены предложения из Хабра.

Для поиска оптимальных гиперпараметров, было проведено несколько экспериментов. В статье [1] были предложено несколько вариантов оптимальных гиперпараметров для дообучения BERT. Параметры приведенные в таблице 2 показали наилучшие значения метрик на валидационной выборке для всех обучаемых моделей.

Во время первой эпохи, чтобы избежать переобучения, веса кодировщика были заморожены, обучались только веса классификатора. Начиная со второй эпохи скорость обучения линейно снижалась с 2e5 до нуля.

Таблица 2: Гиперпараметры обучения

Параметр	Значение
Число эпох	4
Максимальная скорость обучения	2e-5
Размер серии	32

4. Эксперимент

Для оценки качества решения задачи был размечен небольшой датасет (100 предложений). В качестве метрик были выбраны Accuracy, Precision, Recall. Значения метрик обученных моделей представлены в таблице 1.

Модель	Accuracy	Precision	Recall
multilingual $\text{BERT}_{\text{BASE}}$	0.86	0.90	0.84
DeepPavlov $\text{ruBERT}_{\text{BASE}}$	0.89	0.89	0.89
sberbank-ai $\text{ruBERT}_{\text{BASE}}$	0.88	0.89	0.86
cointegrated $\text{ruBERT}_{\text{tiny}}$	0.85	0.85	0.83

Таблица 3: Метрики на валидационной выборке

Наилучшим с точки зрения точности оказалась модель multilingual $\text{BERT}_{\text{BASE}}$. Однако данная модель требует больших вычислительных мощностей. С практической точки зрения можно выделить $\text{ruBERT}_{\text{tiny}}$ Скорость инференса (CPU) и вес модели меньше чем у остальных, при этом имеем лишь незначительные потери в качестве.

Заключение

В ходе работы были выполнены следующие задачи:

1. Проведен обзор алгоритмов обработки текстов на естественном языке для задачи определения научного стиля.
2. Собран и обработан датасет.
3. Разработано несколько алгоритмов для классификации текстов.
4. Создана программная среда для апробации создаваемых алгоритмов.
5. Получены значения ключевых метрик для различных алгоритмов.

В дальнейшем планируется оптимизировать модель, для работы на CPU, разработать интерфейс и внедрить модель в сервис MAP. Открытый код проекта доступен по ссылке[1] на репозиторий Github.

[1] https://github.com/artyomrabosh/practice_map — репозиторий на Github (дата обращения: 14.04.2023).

Список литературы

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers) / Ed. by Jill Burstein, Christy Doran, Thamar Solorio. — Association for Computational Linguistics, 2019. — P. 4171–4186. — URL: https://doi.org/10.18653/v1/n19-1423.

[2] Deep Learning–Based Text Classification: A Comprehensive Review / Shervin Minaee, Nal Kalchbrenner, Erik Cambria et al. // ACM Comput. Surv. — 2021. — apr. — Vol. 54, no. 3. — 40 p. — URL: https://doi.org/10.1145/3439726.

[3] Dementieva Daryna, Trifinov Ivan, Likhachev Andrey, Panchenko Alexander. Detecting Text Formality: A Study of Text Classification Approaches. — 2022. — 2204.08975.

[4] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer / Colin Raffel, Noam Shazeer, Adam Roberts et al. // J. Mach. Learn. Res. — 2020. — jan. — Vol. 21, no. 1. — 67 p.

[5] Language Models are Few-Shot Learners / Tom Brown, Benjamin Mann, Nick Ryder et al. // Advances in Neural Information Processing Systems / Ed. by H. Larochelle, M. Ranzato, R. Hadsell et al. — Vol. 33. — Curran Associates, Inc., 2020. — P. 1877–1901. — URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[6] Shavrina T. Shapovalova O. A. TO THE METHODOLOGY OF CORPUS CONSTRUCTION FOR MACHINE LEARNING: «TAIGA» SYNTAX TREE CORPUS AND PARSER. — 2017.

[7] Гусева Е.Ю. Дикова Т.Ю. Полякова Ю.Д. Методические рекомендации для обучающихся к курсу «Научный стиль речи» // Методические рекомендации для обучающихся к курсу «Научный стиль речи». — 2018.

[8] Дубовик Анна. Автоматическое определение стилистической принадлежности текстов по их статистическим параметрам // Компьютерная лингвистика и вычислительные онтологии. — 2018. — 01.

[9] Е. В. Красильникова. Обучение научному стилю речи в практике преподавания русского языка как иностранного // Обучение научному стилю речи в практике преподавания русского языка как иностранного. — 2017.

[10] Оспанова Б.Р. Учебное пособие по научному стилю речи // Учебное пособие по научному стилю речи. — 2003.