Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 21.Б10-мм

Определение стиля библиографической записи с помощью машинного обучения

Копань Артём Юрьевич

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
ассистент кафедры ИАС Г. А. Чернышев

Санкт-Петербург
2023

Оглавление

Введение

Каждый год множество студентов делает учебные практики и выпускные квалификационные работы. Все эти работы должны соответствовать единообразному стандарту оформления, но соблюсти его не всегда бывает просто. В официальном стандарте оформления отчётов по учебным практикам и ВКР есть множество неочевидных нюансов, поэтому студенты нередко допускают ошибки в оформлении работы. Одна из распространённых ошибок — несоответствие стиля оформления библиографии.

В системе вёрстки LaTeX ссылки на источники (библиография) указываются по особым правилам: все ссылки в тексте и библиографические записи в списке литературы должны иметь единообразный вид. Совокупность этих правил называется *библиографическим стилем*. Важно, чтобы в тексте отчёта по учебной практике или в тексте ВКР стиль соответствовал одному из стандартов, например, ГОСТу. При проверке работы на ошибки оформления нужно в том числе уметь определять, правильный ли стиль используется в тексте.

Насчитывается множество библиографических стилей, и на первый взгляд различия между ними незначительны. Задача состоит в том, чтобы научиться надёжно различать их.

Использование обычных правил или регулярных выражений здесь будет слишком трудоёмким [1]. Поэтому для решения поставленной задачи было решено использовать методы машинного обучения, которые позволят распознать закономерности в библиографических записях, оформленных с использованием различных стилей, и эффективно предсказывать стиль переданной на вход записи. Несмотря на то, что реализованные решения рассматриваемой задачи уже существуют, было решено реализовать собственную разработку для внедрения в сервис автоматической проверки оформления учебных практик и выпускных квалификационных работ MAP.

Mundane Assignment Police (MAP) [2] — это веб-сервис, предназначенный для автоматической проверки текстов учебных практик и ВКР

на соответствие принятому стилю оформления и отсутствие стилистических ошибок.

1. Постановка задачи

Целью работы является разработка алгоритма для предсказания стиля библиографической записи по самой записи. Для её выполнения были поставлены следующие задачи:

1. Выявить признаки, по которым различаются библиографические стили, и составить список основных разновидностей стилей.

2. Найти существующие алгоритмы, решающие поставленную задачу, описать их достоинства и недостатки.

3. Найти достаточное количество файлов стилей (.bst) и библиографий (.bib) и сформировать датасет для обучения модели машинного обучения.

4. Разработать и реализовать алгоритм выделения и предобработки значимых признаков в библиографических записях.

5. Провести экспериментальное исследование и сделать вывод, какой алгоритм машинного обучения лучше справляется с поставленной задачей.

2. Обзор

2.1. Вёрстка библиографии в LaTeX

Библиографический стиль — это набор правил, регулирующих внешний вид текста библиографической записи (в частности, в системе вёрстки LaTeX). Библиографический стиль декларирует, в каком порядке в записи должны располагаться автор источника, его название, год издания и издательство, какой вид должна иметь ссылка на источник в тексте публикации, что стоит выделить курсивом или полужирным шрифтом и т.д. В России в соответствии с видом библиографической записи используются правила государственных стандартов:

1. Для списка литературы — ГОСТ 7.1–2003.
2. Для библиографических ссылок — ГОСТ Р 7.0.5–2008.
3. Для ссылок на электронные ресурсы — ГОСТ 7.82–2001.

Кроме того, используются ГОСТ 7.80–2000, ГОСТ Р 7.0.12–2011 и ГОСТ 7.11–2004 [6, 7].

Библиографические стили можно подразделить на несколько основных групп [5]:

- **plain** — верстает нумерованный список литературы, в котором источники сортируются в алфавитном порядке по имени первого автора, затем по годам и по названиям публикаций. Например, **amsplain**:

  [1] *10th IEEE international conference on high performance computing and communications, HPCC 2008, 25–27 sept. 2008, dalian, china,* IEEE Computer Society, 2008.

- **alpha** — сортирует аналогично plain, но в библиографиях и ссылках источники помечаются аббревиатурой, состоящей из первых букв фамилий трёх первых авторов и последних двух цифр года выхода публикации, например, JPG+05. Знак + ставится, когда

в публикации больше трёх соавторов. Аббревиатуру единственного автора образуют первые три буквы фамилии. Например, **amsalpha**:

[ABD+95] Gad Ariav, Cynthia Mathis Beath, Janice I. DeGross, Rolf Hoyer, and Chris F. Kemerer (eds.), *Proceedings of the sixteenth international conference on information systems, amsterdam, the netherlands, december 10–13, 1995*, Association for Information Systems, 1995.

- **abbrv** — библиография верстается аналогично стилю plain, но имена авторов сокращаются до инициалов. Например, **h-physrev-5**:

  [1] D. Palmer-Brown and M. Kang, Adfunn: An adaptive function neural network, in *Adaptive and Natural Computing Algorithms*, edited by B. Ribeiro, R. F. Albrecht, A. Dobnikar, D. W. Pearson, and N. C. Steele, Springer Computer Series, pp. 1–4, Coimbra, Portugal, 2005, Springer.

- **ugost** — стили, соответствующие стандартам ГОСТ РФ. В соответствии с ГОСТ 7.0.5–2008 список из четырёх и менее соавторов предшествует названию публикации и выводится без сокращения, а более длинные списки сокращаются до первых трёх соавторов и следуют за названием. Эти стили, в свою очередь, подразделяются на несколько подвидов:

  - **ugost2003, ugost2008, ugost2003l, ugost2008l** — источники следуют порядку цитирования в тексте публикации. Кроме того, стили с индексом «l» выводят полный список авторов перед названием публикации вне зависимости от их числа.

    Пример: **ugost2008**

    [1] Proceedings of the 13th All-Russian Scientific Conference "Digital libraries: Advanced Methods and Technologies, Digital Collections", RCDL 2011, Voronezh, Russia, October 19–22, 2011

/ Ed. by Leonid A. Kalinichenko, Alexander Sychev, Sergey A. Stupnikov. — Vol. 803 of CEUR Workshop Proceedings, CEUR-WS.org, 2011. — Access mode: http://ceur-ws.org/Vol–803.

Пример: **ugost2008l**

[2] Lanza-Gutierrez José Manuel, Gómez-Pulido Juan Antonio, Vega-Rodríguez Miguel A., Sánchez-Pérez Juan Manuel. Optimizing Energy Consumption in Heterogeneous Wireless Sensor Networks by Means of Evolutionary Algorithms // Applications of Evolutionary Computing, EvoApplications2012: EvoCOMNET, EvoCOMPLEX, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoNUM, EvoPAR, EvoRISK, EvoSTIM, EvoSTOC / Ed. by Cecilia Di Chio, Alexandros Agapitos, Stefano Cagnoni et al. — Vol. 7248 of LNCS. — Malaga, Spain : Springer Verlag, 2012. — 11–13 April. — P. 1–10.

– **ugost2003s, ugost2008s, ugost2003ls, ugost2008ls** — сортируют источники либо по фамилии первого автора, либо по названию публикации, если число соавторов больше четырёх.

Пример: **ugost2008s**

[1] Alander Jarmo, Moghadampour Ghodrat, Ylinen Jari. Comparison of elevator allocation methods // Proceedings of the Second Nordic Workshop on Genetic Algorithms and their Applications (2NWGA) / Ed. by Jarmo T. Alander. — Proceedings of the University of Vaasa, Nro. 13. — Vaasa (Finland) : University of Vaasa, 1996. — 19.–23. . — P. 211–214.

- **IEEE** — семейство стилей, стандартизированных Институтом инженеров электротехники и электроники (Institute of Electrical and Electronics Engineers, IEEE).

  Пример: **IEEEtranSN**

  [Kalmykova and Kogalovsky(2014)] L. Kalmykova and M. R. Kogalovsky, Eds., *Selected Papers of XVI All-Russian Scientific Conference "Digital libraries: Advanced Methods and Technologies, Digital Collections Dubna,*

Russia, October 13–16, 2014, ser. CEUR Workshop Proceedings, vol. 1297. CEUR-WS.org, 2014. [Online]. Available: http://ceur-ws.org/-Vol–1297

Как можно видеть, большинство этих стилей отличаются друг от друга весьма незначительно, поэтому различать их — весьма нетривиальная задача.

2.2. Обзор аналогов

При поиске аналогов проектируемого сервиса было найдено только одно существующее решение: open-source-инструмент [4] от исследователя компании Crossref Dominika Tkaczyk.

Автору этого решения удалось добиться точности предсказаний в 94.7% при использовании алгоритма линейной регрессии.

Стоит сказать несколько слов о недостатках этого решения:

1. Алгоритм выделения признаков не работает с пробелами. Возможно, учёт пробелов при выделении признаков может существенно повлиять на точность предсказаний.

2. Используется всего 17 библиографических стилей.

3. Реализация

3.1. Выбор метода

При выборе метода для определения библиографического стиля записи у нас было три альтернативы:

1. Использовать правила, декларативно описывающие структуру библиографической записи определённого стиля, и валидатор, проверяющий соблюдение этих правил для каждой записи;
2. Использовать регулярные выражения.
3. Использовать алгоритмы классификации на основе машинного обучения.

Согласно статье [1], правила в большинстве случаев проигрывают машинному обучению. В статье разбирается более сложная задача парсинга библиографической записи, но в задаче определения стиля также есть много частных случаев, которые надо учитывать при написании правил, поэтому выявленная тенденция применима и в данном случае. Написание регулярных выражений для большого количества стилей тоже пришлось отбросить, т.к. это занимает очень много времени, а вероятность допустить ошибку очень высока. Поэтому было решено применить в данной работе алгоритмы машинного обучения.

3.2. Используемые технологии

Для реализации основной части сервиса был выбран язык программирования Python (а именно версия Python 3.10). Он обладает рядом весомых преимуществ:

- наличие хорошего готового инструментария для машинного обучения и анализа данных (библиотеки pandas, scikit-learn, catboost и т.д.);
- удобная работа с файлами и файловой системой;

- простота интеграции с другими технологиями (в проекте MAP нужно будет интегрировать решение в основную часть приложения, написанную на Kotlin).

Для генерации датасета также использовались консольные утилиты Linux, язык сценариев Bash и компилятор pdfLaTeX.

3.3. Генерация датасета

Первоочередной задачей на этом этапе стал поиск достаточного количества библиографических записей (файлов .bib) и файлов библиографических стилей (.bst).

Библиографические записи были получены преимущественно из данных на сайте dblp.org — открытом сборнике библиографий по компьютерным наукам. Из-за своеобразной структуры сайта скачивание производилось вручную, из сборников bib-файлов по годам. Таким образом было скачано в общем более шести миллионов библиографических записей.

Библиографические стили LaTeX были взяты с открытого архива документации и программного обеспечения для TeX ctan.org. Был скачан 91 bst-файл.

Далее для машинного обучения требовалось извлечь из текста библиографий некие числовые признаки (features, “фичи”). Чтобы их подсчитать, сначала нужно было сгенерировать сам текст в формате pdf. Для этого использовался дистрибутив LaTeX pdfLaTeX. Для непосредственной генерации использовался скрипт на bash bibgen.sh[1]. Он вызывался из Python-программы generate_pdf.py, которая последовательно перебирала все bib-файлы и bst-файлы и подставляла их имена в заранее созданный шаблон для генерации. В итоге было получено 19877 pdf-файлов с разным количеством библиографических записей.

Следующим этапом стал парсинг полученных pdf-файлов и подсчёт числовых признаков.

[1]Все указанные здесь файлы с кодом можно найти в репозитории GitHub https://github.com/ArtyomKopan/Practical_Training_1

Таблица 1: Сравнение времени чтения PDF-файла разными библиотеками

Библиотека	Время чтения файла (с)
PyPDF2	5.5
PDFminer	17.2
PDFium	0.8

Библиотека для работы с pdf выбиралась из нескольких вариантов:

- PyPDF2;
- PDFminer;
- PDFium.

PyPDF2 — самая популярная Python-библиотека для работы с pdf, но оказалось, что она неправильно распознаёт длинные тире в тексте и вставляет вместо них вертикальные черты. Был проведён эксперимент с измерением времени считывания pdf-файла с помощью каждой из перечисленных библиотек (для замера использовался файл daiugost2003.pdf объёмом в 90 страниц). Результаты представлены в таблице 1.

Как можно видеть, наилучшие результаты оказались у библиотеки PDFium, поэтому было решено использовать её.

Для исследования использовались два варианта предобработки признаков, сгенерированных различным образом.

3.3.1. Подсчёт числовых признаков

В первом случай использовался простой подсчёт числовых и категориальных признаков (а именно количество вхождений некоторого символа пунктуации, а также тип ссылки на запись) по входной строке библиографической записи. Были подсчитаны следующие признаки:

- square_brackets — количество пар квадратных скобок;
- round_brackets — количество пар круглых скобок;

- slashes — количество косых черт;
- inverse_slashes — количество обратных косых черт;
- quotes — количество пар кавычек;
- dots — количество точек;
- commas — количество запятых;
- semicolons — количество точек с запятой;
- colons — количество двоеточий;
- abstract — наличие слова “Abstract” или “abstract” в записи — категориальная переменная;
- ands — количество слов “and” в записи — категориальная переменная;
- ampersands — количество амперсандов;
- page_ref — тип ссылки на страницу (p., P., pp., стр., С.) — категориальная переменная;
- begin_ref — тип начала библиографической записи ([1], [TrD90], [Ski(2022)], [SDD, 99]) — категориальная переменная;
- tirets — количество тире и дефисов;
- key — наличие слова “Key:” в записи — категориальная переменная;
- annotation — наличие слова “Annotation” в записи — категориальная переменная;
- capital_letters — доля слов, начинающихся с заглавной буквы;
- years — число чисел, похожих на год (2022, 22, 90, 1990) в строке;

- sine — есть ли сокращения [s.l.] и [s.n.] (0 = нет, 1 = s.l., 2 = [s.l.], 3 = (s.l.)) — категориальная переменная;
- et_al — есть ли сокращение et al. (0 = нет, 1 = et al., 2 = [et al.], 3 = (et al.)) — категориальная переменная;
- etc — есть ли сокращение etc. — категориальная переменная.

В качестве зависимой переменной используется название стиля конкретной библиографической записи (style_name).

3.3.2. Векторизация

Затем был применён другой подход — *векторизация*, идея которого основана на статье [4]. Сначала в библиографической записи конкретные слова заменяются на токены:

- слово в верхнем регистре — uppercase_word, upword;
- слово, которое начинается с большой буквы — capitalized_word, capword;
- другое слово — other_word, othword;
- большая буква (capital_letter, caplet);
- маленькая буква (small_letter, smallet);
- год (year) — число, которое *скорее всего* является годом, т.е. лежит в диапазоне от 1900 до 2100;
- число, не являющееся годом (number, num);
- пробел (space, sp). Выделение расстановки пробелов как отдельного признака является одним из основных отличий от аналога [4].

Знаки пунктуации сохраняются и считаются токенами сами по себе. В качестве зависимой переменной также используется название стиля записи.

Таким образом, каждая библиографическая запись преобразуется в строку, состоящую из вышеприведённых токенов. Затем полученная строка передаётся в класс *CountVectorizer* из модуля *sklearn.feature_ extraction.text* и преобразуется в таблицу n-грамм (n-gramms). *N-грамма* — это последовательность из n слов, идущих подряд в исходном тексте. Каждая выделенная n-грамма становится отдельным признаком, значение которого равно количеству вхождений этой n-граммы в переданную строку.

3.4. Обучение моделей

Рассматриваемая задача представляет собой *задачу мультиклассификации*, т.е. классификацию не с двумя, а с большим количеством классов. Цель: как можно точнее присвоить метку класса (название библиографического стиля) каждой записи в датасете. Для решения этой задачи были выбраны различные *классификаторы*, использующие машинное обучение.

1. Линейная классификация.
2. Наивная байесовская классификация.
3. Случайный лес.
4. Деревья решений, построенные с использованием метода градиентного бустинга.

Модели, показавшие наилучшие результаты, были сохранены в файлы формата .sav с помощью библиотеки *joblib*. Они доступны для загрузки и дальнейшего практического применения без необходимости обучения.

4. Эксперимент

4.1. Условия эксперимента

Эксперимент проводился в среде выполнения Jupyter Notebook на удалённом сервере с 80 CPU Intel Xeon Gold 6230 и GPU NVIDIA GP104 GeForce GTX 1080.

Для эксперимента использовался датасет, полученный способом, описанным в предыдущей главе: всего около 6 млн библиографических записей с 91 разным стилем.

Назовём *первым вариантом предобработки признаков* набор данных, полученный из сформированного датасета, полученный подсчётом числовых признаков (см. 3.3.1); *второй вариант предобработки признаков* получен путём выделения токенов и векторизации (см. 3.3.2).

4.2. Метрики

Для измерения точности классификации использовались следующие метрики:

1. Доля правильных ответов алгоритма (accuracy score):

$$\texttt{accuracy}(y, \hat{y}) := \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i).$$

2. Точность (precision):

$$P(A, B) := \frac{|A \cap B|}{|B|}.$$

   Показывает способность классификатора не отмечать отрицательный образец как положительный.

3. Полнота (recall):

$$R(A, B) := \frac{|A \cap B|}{|A|}.$$

Показывает способность классификатора находить все положительные образцы.

4. F_1:
$$F_1(A, B) := \frac{2 \times P(A, B) \times R(A, B)}{P(A, B) + R(A, B)}.$$
Средневзвешенное гармоническое среднее precision и recall.

Отметим, что в случае задачи мультиклассификации precision, recall и F_1-score вычисляются для каждого класса в отдельности. Существует несколько методов их усреднения: *micro, macro, samples, weighted* [3]. В проводимом эксперименте использовалось усреднение *samples*.

4.3. Результаты

В таблице 2 приведены средние значения метрик, полученные при обучении выбранных моделей на первом варианте предобработки признаков.

Таблица 2: Метрики качества различных алгоритмов на первом варианте предобработки признаков

Алгоритм	Accuracy	Precision	Recall	F_1
Линейная классификация	0.41	0.41	0.40	0.37
Наивная байесовская классификация	0.36	0.42	0.37	0.33
Случайный лес	0.54	0.55	0.52	0.50
Дерево решений	0.62	0.60	0.61	0.60

Таблица 3: Метрики качества различных алгоритмов на втором варианте предобработки признаков

Алгоритм	Accuracy	Precision	Recall	F_1
Линейная классификация	0.66	0.65	0.62	0.61
Наивная байесовская классификация	0.55	0.58	0.55	0.55
Случайный лес	0.68	0.67	0.65	0.64
Дерево решений	0.84	0.83	0.82	0.82

В таблице 3 приведены средние значения метрик, полученные при обучении выбранных моделей на втором варианте предобработки признаков.

Таким образом, лучший результат (доля правильных ответов = 84%) показал алгоритм дерева решений, оптимизированный с помощью градиентного бустинга, на векторизованных данных при запуске на GPU.

Заключение

В ходе работы над учебной практикой были достигнуты следующие результаты:

1. Составлен список основных разновидностей библиографических стилей LaTeX с примерами.

2. Произведён поиск аналогичных решений, они проанализированы и выявлены их недостатки. В ходе работы был найден алгоритм, решающий аналогичную задачу, но для меньшего количества стилей.

3. Сформирован датасет для обучения моделей машинного обучения.

4. Были разработаны и реализованы два различных способа выделения и предобработки признаков.

5. Проведено экспериментальное исследование, в ходе которого было выяснено, что наилучшую точность классификации на данном наборе входных данных обеспечивает алгоритм дерева решений, оптимизированный градиентным бустингом.

Код проекта и файлы с данными для обучения моделей можно найти по ссылке в GitHub репозитории[2].

В будущем планируется интегрировать реализованное решение в проект MAP и наладить взаимодействие с другими компонентами сервиса. Также обученная модель нуждается в дальнейшем улучшении точности классификации.

[2] https://github.com/ArtyomKopan/Practical_Training_1

Приложение

Значения гиперпараметров для моделей, показавших лучший результат при использовании первого варианта предобработки данных:

- линейная классификация:

```
SGDClassifier(
  loss='modified_huber',
  penalty=12,
  alpha=0.0001,
  max_iter=1000,
  n_iter_no_change=5
)
```

- наивная байесовская классификация:

```
GaussianNB(var_smoothing=10**-6)
```

- случайный лес:

```
RandomForectClassifier(
  criterion='entropy',
  n_estimators=30,
  max_depth=11
)
```

- градиентный бустинг:

```
CatBoostClassifier(
  iterations=1000,
  learning_rate=0.05,
  depth=4,
  task_type='GPU'
)
```

Параметры векторизации:

```
vectorizer = CountVectorizer(
  tokenizer=lambda txt: txt.split(),
  ngram_range=(2, 2)
)
```

Значения гиперпараметров для моделей, показавших лучший результат при использовании второго варианта предобработки данных:

- линейная классификация:

  ```
  SGDClassifier(loss='hidge')
  ```

- наивная байесовская классификация:

  ```
  MultinomialNB(fit_prior=True)
  ```

- случайный лес:

  ```
  RandomForestClassifier(
    criterion='entropy',
    n_estimators=40,
    max_depth=12
  )
  ```

- градиентный бустинг:

  ```
  CatBoostClassifier(
    iterations=250,
    learning_rate=0.05,
    depth=6
  )
  ```

Список литературы

[1] Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers / Dominika Tkaczyk, Andrew Collins, Paraic Sheridan, Joeran Beel // Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries. — JCDL '18. — New York, NY, USA : Association for Computing Machinery, 2018. — P. 99–108. — URL: https://doi.org/10.1145/3197026.3197048.

[2] Mundane Assignment Police (MAP). — https://github.com/Darderion/map.

[3] Scikit-learn. 3.3.2.9. Precision, recall and F-measures. — URL: https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics (дата обращения: 2022-12-23).

[4] Tkaczyk Dominika. What's your (citations') style? // Crossref. — 2019. — URL: https://www.crossref.org/blog/whats-your-citations-style/ (дата обращения: 2022-12-07).

[5] В. Кузнецов А. Основы LATEX. Учебное пособие. — М.: НИЯУ МИФИ, 2021. — P. 129–144. — ISBN: 78-5-7262-2680-7. — URL: https://ctan.math.utah.edu/ctan/tex-archive/info/russian/basiclatex-ru/BasicLatex.pdf (дата обращения: 2022-12-11).

[6] Международные стандарты в библиографии // ONLINE Scientific Journal "Child and Society". — 2014. — URL: http://childandsociety.ru/ojs/index.php/cas/pages/view/bibliography_international/ (дата обращения: 2022-12-07).

[7] Правила оформления списка литературы // ONLINE Scientific Journal "Child and Society". — 2014. — URL: http://childandsociety.

ru/ojs/index.php/cas/pages/view/bibliography/ (дата обращения: 2022-12-07).