DOI 10.37882/2223-2966.2025.05-2.18

ИССЛЕДОВАНИЕ МЕТОДОВ ВЕКТОРИЗАЦИИ НАУЧНЫХ ТЕКСТОВ ДЛЯ МНОГОЗАДАЧНОЙ КЛАССИФИКАЦИИ НА ОСНОВЕ РАЗЛИЧНОГО ОБЪЕМА ДАННЫХ

STUDY OF VECTORIZATION METHODS OF SCIENTIFIC TEXTS FOR MULTI-TASK CLASSIFICATION BASED ON VARIOUS DATA VOLUMES

K. Potapova
I. Isaeva
G. Gabrielyan

Summary. The study analyzed different vectorization methods in task classification. Two statistical methods were selected for vectorization of scientific articles: bag of words and TF-IDF, as well as one neural network model word2vec. A comparative analysis of different clustering models was conducted, after which two models were selected for the experiment: a modification of logistic regression and a random forest. To assess the impact of input data volume on classification quality, three scenarios were used: using only titles, using titles and abstracts, and using titles, abstracts, and article texts. Each scenario was tested on all vectorization methods and selected classification models, which allowed us to identify the relationship between data completeness, vectorization type, and the resulting classification quality metrics.

Keywords: vectorization, scientific articles, machine learning, classification, semantic analysis.

Потапова Ксения Александровна
старший преподаватель, МИРЭА — Российский технологический университет
potapova_k@mirea.ru
Исаева Ирина Андреевна
старший преподаватель, МИРЭА — Российский технологический университет
isaeva_i@mirea.ru
Габриелян Гайк Ашотович
старший преподаватель, МИРЭА — Российский технологический университет
gabrielyan@mirea.ru

Аннотация. В рамках исследования проведён анализ разных методов векторизации в задаче классификации. Выбраны два статистических метода для векторизации научных статей: мешок слов и TF-IDF, и одна нейросетевая модель word2vec. Проведён сравнительный анализ разных моделей кластеризации, после чего для эксперимента были выбраны две модели: модификация логистической регрессии и случайный лес. Для оценки влияния объёма входных данных на качество классификации использованы три сценария: использование только заголовков, использование заголовков и аннотаций, использование заголовков, аннотаций и текстов статей. Каждый сценарий тестировался на всех методах векторизации и выбранных моделях классификации, что позволило выявить зависимость между полнотой данных, типом векторизации и итоговыми метриками качества классификации.

Ключевые слова: векторизация, научные статьи, машинное обучение, классификация, семантический анализ.

Введение

Современный этап развития науки характеризуется экспоненциальным ростом объёма публикуемой информации. Ежегодно в международных базах данных индексируются миллионы научных статей, монографий, диссертационных исследований и иных форм академических публикаций, охватывающих разные научные области. Методы ручного присвоения УДК-индексов или тематической категоризации, подвержены человеческим ошибкам, и часто приводят к неточностям, затрудняющим поиск и анализ научных материалов.

Перспективным решением упорядочивания таких объёмов информации выступают алгоритмы, способные осуществлять анализ больших массивов научных текстов и выявлять скрытые паттерны связей между научными публикациями. Этот механизм пригодится при решении множества задач: это и автоматизация классификации текстов, и выявление междисциплинарных научных работ, и организация научного сотрудничества. Однако невозможно анализировать текст, созданный человеком, без предварительной обработки. Одним из ключевых этапов предобработки текстовой информации является векторизация — процесс преобразования текстовых данных в числовой формат, пригодный для машинного анализа.

В статье рассмотрены основные применяемые методы векторизации текста: два статистические метода (мешок слов и TF-IDF) и один метод, основанный на нейронных сетях. Также проведён их сравнительный анализ для многозадачной классификации научных статей по четырём направлениям: физика, математика, информатика

и медицина. Для оценки результатов использованы метрики: точность, среднее взвешенное и макроусреднение. Использование этих метрик в совокупности позволяет получить более полную картину качества моделей в условиях неравномерного распределения классов.

Данные собраны из открытых ресурсов, векторизация и классификация проведены на языке Питон с использованием ряда библиотек: pandas, numpy, sklearn, gensim, seaborn, matplotlib.

Структура данных

Набор данных содержит 8567 строк и 22 столбца: у значения темы («section») четыре возможных значения: физика, математика, информатика и медицина. В столбцах «title», «annotation» и «text» соответственно содержатся заголовок, аннотация и текст статьи. В ходе анализа данных также будет использоваться столбец с датой.

На рисунках 1 и 2 представлены графики распределения статей в базе данных по годам написания и медианная длина статьи в зависимости от года написания.

Анализ методов векторизации

Для векторизации текстов выбраны методы мешок слов (CountVectorizer), взвешенный мешок слов (TF-IDF) и метод, основанный на использовании нейросетей для получения эмбеддингов (Word2Vec). Word2Vec — это модель для создания векторных представлений слов, учитывающая семантическую близость. TF-IDF — это статистическая мера для оценки важности слова в документе относительно коллекции документов, а мешок слов преобразует текст в матрицу частот слов, где каждая ячейка — количество употреблений слова в документе, без учёта семантики или значимости. Сравнительный анализ преимуществ и недостатков каждой модели приведён в таблице 1.

Анализ методов классификации

Для построения модели предсказания целевых переменных следует выбрать оптимальный алгоритм.

Так как основной целью исследования является сравнение разных моделей векторизации текстов, будем использовать только два алгоритма классификации: модификацию логистической регрессии и случайный лес. Логистическая регрессия является линейной моделью, которую часто используют в начале подбора оптимальной модели классификации, а её модификация позволяет не акцентировать внимание на подборе гиперпараметров. Случайный лес реализует ансамблевый подход, агрегирующий предсказания множества деревьев решений, обученных на случайных подвыборках данных и признаков. Его выбор связан со способностью выявлять нелинейные зависимости и устойчивостью к шуму. Таким образом, комбинация этих двух моделей обеспечивает диверсификацию критериев оценивания за счёт использования и линейного, и нелинейного подходов.

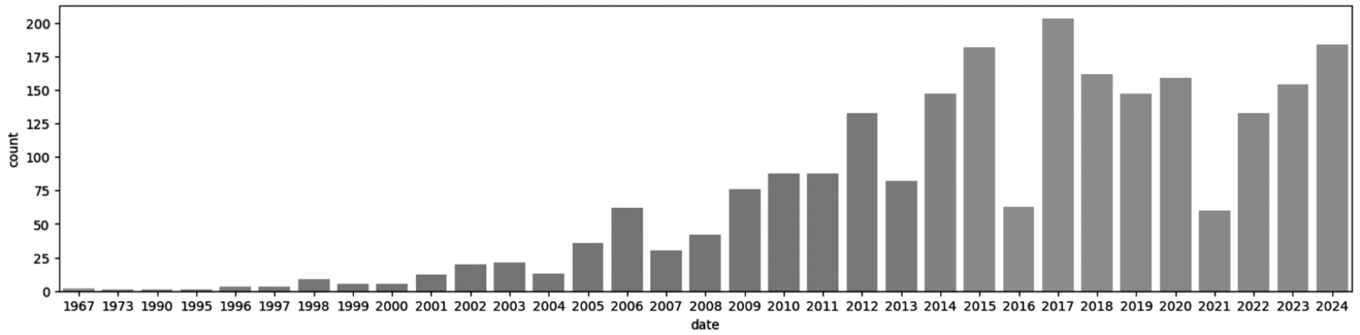


Рис. 1. Количество статей в базе данных по годам написания

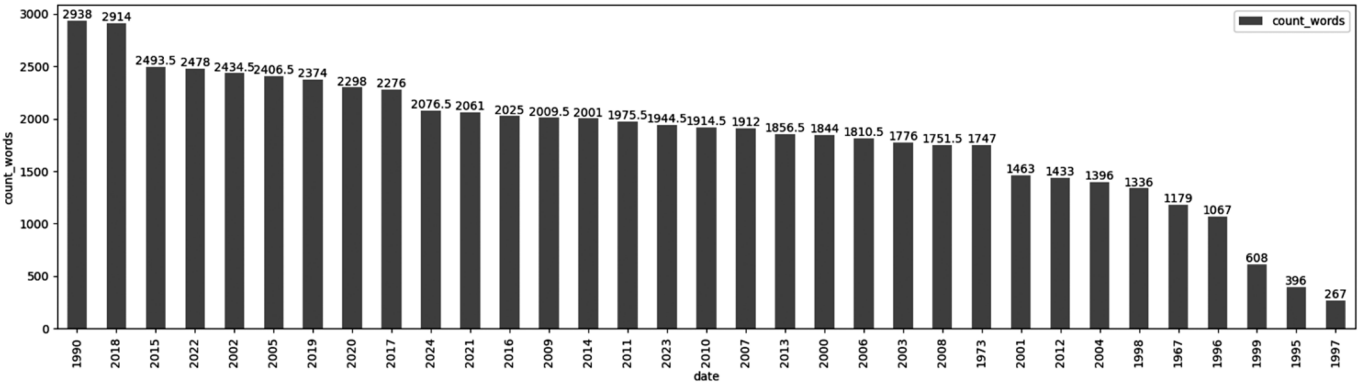


Рис. 2. Медианная длина статьи в зависимости от года написания

Таблица 1.

Сравнение разных методов классификации

Метод	Преимущества	Недостатки
CountVectorizer	Простота реализации Высокая скорость обработки данных Хорошая интерпретируемость	Не учитываются грамматические особенности текста Теряется семантический контекст слов [1]
TF-IDF	Учитывает зависимость важности слова от объёма текста [1] Простая вероятностная интерпретация результатов Хорошие результаты в задачах классификации [3]	Значение коэффициента встречаемости может быть низким за счёт большого количества синонимов в русском языке [2] Высокая зависимость от корпуса текстов
Word2Vec	Учитывает семантику слов [4] Хорошее масштабирование на большие объёмы текстовых данных	Требовательность к вычислительным ресурсам Неспособность работать со словами, которых не было в обучающем корпусе [4]

Таблица 2.

Сравнение разных методов классификации

Метод	Преимущества	Недостатки
DecisionTree	не требует тщательной предобработки данных модель способна обрабатывать нелинейные зависимости [5]	склонность к переобучению низкая обобщающая способность, требует тщательной настройки параметров
RandomForest	устойчив к переобучению работает с шумными данными	медленный на больших данных сложный в интерпретации
LogisticRegression	хорошо интерпретируемая модель быстрое обучение устойчивость к шуму с L2-регуляризацией	чувствительность к мультиколлинеарности не подходит для нелинейных задач [5]
LogisticRegressionCV	автоматический подбор параметров регуляризации более низкий риск переобучения чем у обычной логистической регрессии	сохраняет недостатки базовой логистической регрессии медленнее базовой логистической регрессии

В таблице 2 рассмотрены преимущества и недостатки использованных при сравнении моделей классификации: случайный лес (RandomForest), логистическая регрессия (LogisticRegression), модификация логистической регрессии (LogisticRegressionCV) и дерево решений (DecisionTree) [5].

Для построения модели классификации требуется разбить данные на обучающую и тестовую выборку [6]. Сначала будем использовать только заголовок статьи. Разбиваем в соотношении 0.33 тестовых данных и 0.66 обучающих данных.

Метрики

При оценке качества моделей использовались метрики:

- Точность (accuracy) — высчитывает правильно спрогнозированную долю выборки на основе истинно-положительных (TP), истинно-отрицательных (TN), ложно-положительных (FP) и ложно-отрицательных (FN) показателей (1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Среднее взвешенное (weighted avg) — усреднение метрик по классам с учетом их размера. w_i представляет собой вес, присвоенный каждому значению x_i. x_i — значение в наборе данных [7]. В среднем взвешенном каждый класс имеет вес, пропорциональный его размеру (2):

$$Weighted\ Average = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \quad (2)$$

- Макро усреднение (macro avg) — усреднение метрик по классам без учёта их размера. Для четырёх классов вес будет равен 0.25, а x_i — значение в наборе данных (3):

$$Macro\ average = \frac{0.25 * x_1 + 0.25 * x_2 + 0.25 * x_3 + 0.25 * x_4}{4} \quad (3)$$

Все требуемые метрики есть в методе classification_report библиотеки sklearn [8]. Также, для многозадачной

классификации отображается микросреднее — усреднение общего числа истинно положительных, ложно отрицательных и ложно положительных результатов для каждого класса, и метрики f-score, presicion и recall. Примеры вывода можно увидеть на рисунках 7 и 8.

Мешок слов

Первый используемый метод — мешок слов. Он основан на идее простого подсчёта слов в документе.

Результатом применения метода CountVectorizer из библиотеки scikit-learn [9] является формирование разреженной матрицы признаков, где строки соответствуют документам, а столбцы — уникальным словам корпуса. Пример кода показан на рисунке 3.

Значения элементов матрицы отражают частоту встречаемости терминов в соответствии с лексикографическим порядком сортировки словаря [9]. Параметр ngram_range позволяет задать диапазон n-граммных комбинаций (например, (1, 2) для учета униграмм и биграмм), что обеспечивает сохранение зависимостей между словами.

После векторизации проведено обучение моделей случайный лес и модификации логистической регрессии. Использованы метрики точность (accuracy), macro avg, среднее взвешенное (weighted avg).

С увеличением количества данных метрики точности растут. На рисунке 4 показаны метрики обученных моделей логистической регрессии и случайного леса для обучения на заголовках и аннотациях, а также для обучения на заголовках, аннотациях и текстах статей.

При анализе полученных метрик можно сделать вывод, что количество данных резко увеличивает качество классификации, доходя до 97% точности при использовании заголовка, аннотации и текста.

TF-IDF

Второй используемый метод векторизации основан на TF-IDF. Это статистическая мера для оценки важности слова в документе относительно коллекции документов. TF это частота, обозначающая, насколько часто определенное слово появляется в данном документе. IDF это обратная частота документа, она измеряет, насколько уникально слово является по всей коллекции документов. В формуле 4 n_k это число вхождений слова t в документ, $\sum_k n_k$ — общее количество слов в документе, $|D|$ — число документов в коллекции, $|\{d_i \in D | t \in d_i\}$ — число документов из коллекции D, в которых встречается t:

$$tfidf = \frac{n_t}{\sum_k n_k} * log \frac{|Dc|}{|\{d_i \in D | t \in d_i\}|} \quad (4)$$

TF-IDF присваивает высокий вес словам, которые часто встречаются в конкретном документе, но редко — в других документах корпуса.

Полученные метрики для моделей модифицированной логистической регрессии и классификатора случайного леса для разного количества используемых данных показаны на рисунке 5.

```
vec = CountVectorizer(ngram_range=(1,1))
bow = vec.fit_transform(X_train)
```

Рис. 3. Векторизация текстов методом «CountVectorizer»

Count Vectorizer

LogRegCV

заголовок	precision	recall	f1-score	support
0.0	0.79	0.79	0.79	803
1.0	0.64	0.62	0.63	666
2.0	0.87	0.88	0.87	664
3.0	0.76	0.78	0.77	695
accuracy			0.77	2828
macro avg	0.77	0.77	0.77	2828
weighted avg	0.77	0.77	0.77	2828

заголовок, аннотация	precision	recall	f1-score	support
IT	0.82	0.84	0.83	476
math	0.71	0.74	0.72	367
med	0.96	0.86	0.91	452
physics	0.83	0.87	0.85	419
accuracy			0.83	1714
macro avg	0.83	0.83	0.83	1714
weighted avg	0.83	0.83	0.83	1714

заголовок, аннотация, текст	precision	recall	f1-score	support
IT	1.00	0.99	1.00	488
math	0.97	0.93	0.95	394
med	0.99	1.00	0.99	405
physics	0.94	0.97	0.96	427
accuracy			0.97	1714
macro avg	0.97	0.97	0.97	1714
weighted avg	0.97	0.97	0.97	1714

RandomForest

заголовок	precision	recall	f1-score	support
0.0	0.74	0.69	0.71	868
1.0	0.49	0.69	0.57	450
2.0	0.75	0.79	0.77	643
3.0	0.71	0.58	0.64	867
accuracy			0.68	2828
macro avg	0.67	0.69	0.67	2828
weighted avg	0.69	0.68	0.68	2828

заголовок, аннотация	precision	recall	f1-score	support
IT	0.80	0.72	0.76	540
math	0.52	0.66	0.58	300
med	0.85	0.87	0.86	397
physics	0.75	0.69	0.72	477
accuracy			0.73	1714
macro avg	0.73	0.73	0.73	1714
weighted avg	0.75	0.73	0.74	1714

заголовок, аннотация, текст	precision	recall	f1-score	support
IT	1.00	1.00	1.00	486
math	0.89	0.94	0.91	361
med	1.00	0.98	0.99	418
physics	0.94	0.92	0.93	449
accuracy			0.96	1714
macro avg	0.96	0.96	0.96	1714
weighted avg	0.96	0.96	0.96	1714

Рис. 4. Использование разного количества данных при обучении моделей логистической регрессии и случайного леса

TF-IDF

Logreg CV

	заголовок				заголовок, аннотация				заголовок, аннотация, текст			
	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
IT	0.79	0.80	0.79	479	0.82	0.83	0.83	479	1.00	1.00	1.00	485
math	0.61	0.62	0.62	372	0.70	0.73	0.71	364	0.94	0.94	0.94	381
med	0.88	0.88	0.88	406	0.95	0.90	0.92	432	1.00	0.99	0.99	410
physics	0.79	0.76	0.77	457	0.85	0.85	0.85	439	0.94	0.95	0.95	438
accuracy			0.77	1714			0.83	1714			0.97	1714
macro avg	0.77	0.77	0.77	1714	0.83	0.83	0.83	1714	0.97	0.97	0.97	1714
weighted avg	0.77	0.77	0.77	1714	0.84	0.83	0.83	1714	0.97	0.97	0.97	1714

Random Forest

	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
IT	0.75	0.72	0.74	504	0.78	0.70	0.74	538	1.00	1.00	1.00	486
math	0.52	0.63	0.57	317	0.53	0.65	0.58	312	0.88	0.94	0.91	357
med	0.80	0.79	0.79	416	0.87	0.80	0.84	441	1.00	0.98	0.99	415
physics	0.72	0.66	0.69	477	0.69	0.72	0.70	423	0.95	0.91	0.93	456
accuracy			0.70	1714			0.72	1714			0.96	1714
macro avg	0.70	0.70	0.70	1714	0.72	0.72	0.72	1714	0.96	0.96	0.96	1714
weighted avg	0.71	0.70	0.71	1714	0.74	0.72	0.73	1714	0.96	0.96	0.96	1714

Рис. 5. Метрики качества разных моделей кластеризации с разным количеством данных для TF-IDF векторизации

В методе настроен параметр max_features — параметр, ограничивающий количество признаков в выходной матрице. Выбираются топ-N слов с наибольшей частотой встречаемости в корпусе и может значительно ускорить обучение моделей.

Можно заметить, что значения метрик незначительно выросли по сравнению с векторизацией текстов с помощью мешка слов. Сохраняется зависимость точности от размера используемых данных.

Word2Vec

Word2Vec это модель для создания векторных представлений слов, учитывающая семантическую близость. Архитектура модели на рисунке 6.

Модель состоит из двух компонентов — CBOW (Continuous Bag of Words) и Skip-Gram. Первый компонент предсказывает целевое слово по его контексту в пределах заданного окна, Skip-Gram решает обратную задачу: по целевому слову предсказывает контекстные слова. В Skip-Gram входной вектор целевого слова проходит через скрытый слой, а на выходе модель генерирует вероятности для окружающих слов через несколько softmax-слоёв. В результате Word2Vec формирует векторные представления, где слова с похожим значением или ролью в предложении оказываются близко в векторном пространстве [11].

Для применения модели word2vec используем метод word2vec из библиотеки gensim [12]. Также необходима функция для преобразования векторов, так как в моделях классификации необходимы векторы одинаковой длины, поэтому все значения векторов, полученных в результате векторизации word2vec, надо дозаполнить нулями до достижения одинаковой длины. Отдельно

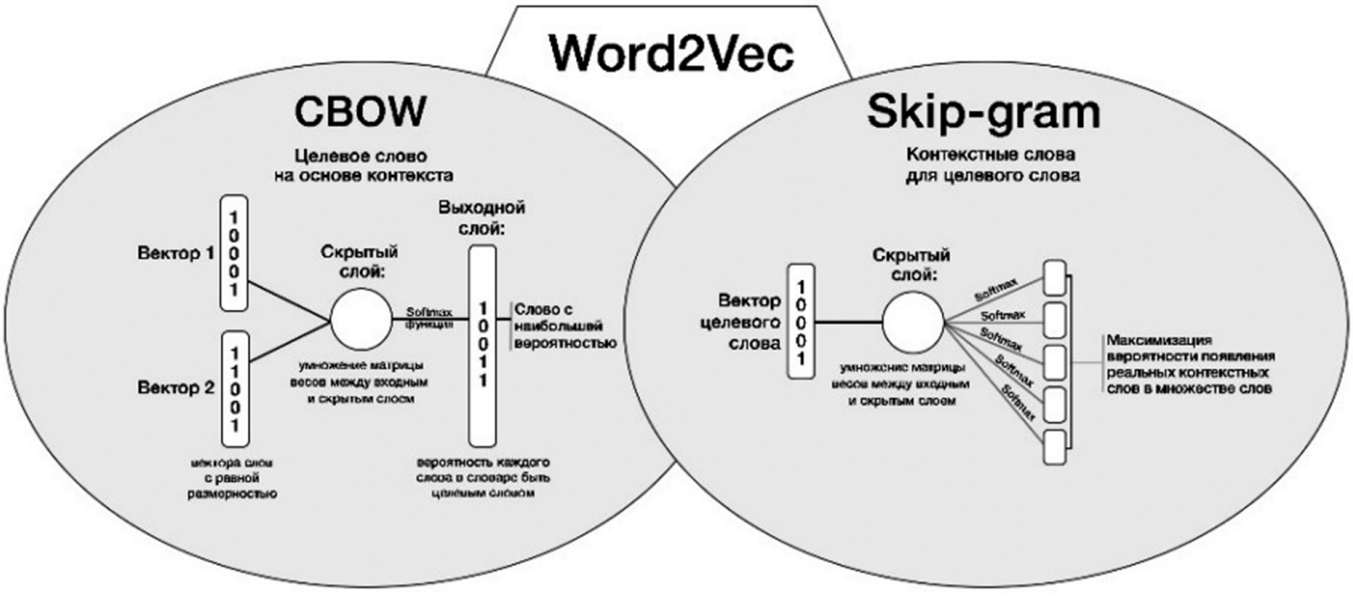


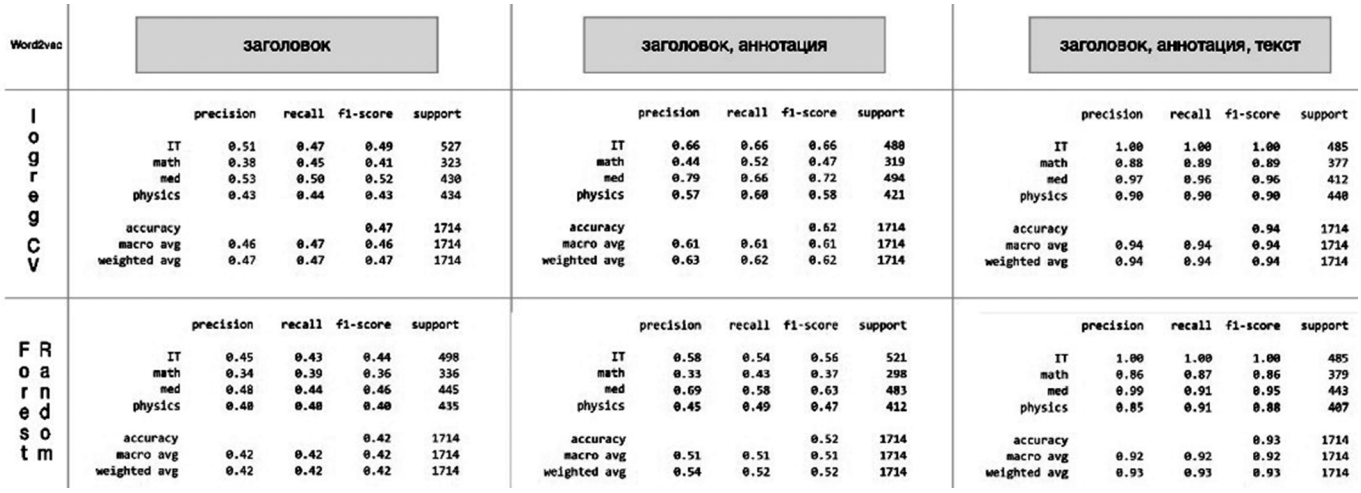
Рис. 6. Архитектура word2vec

```
word2vec_model = Word2Vec(sentences=df['title_tokens'], vector_size=100, window=5, min_count=1, workers=4)
```

```
# Функция для преобразования текста в усредненный вектор Word2Vec
def text_to_vector(tokens, model, vector_size=100):
    vectors = [model.wv[word] for word in tokens if word in model.wv]
    if len(vectors) > 0:
        return np.mean(vectors, axis=0)
    else:
        return np.zeros(vector_size)
```

```
# Преобразуем каждый заголовок в вектор
X = np.array([text_to_vector(tokens, word2vec_model) for tokens in df['title_tokens']])
```

Рис. 7. Пример использования модели word2vec из библиотеки gensim для заголовков



Word2vec

Logregcv

	заголовок				заголовок, аннотация				заголовок, аннотация, текст			
	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
IT	0.51	0.47	0.49	527	0.66	0.66	0.66	488	1.00	1.00	1.00	485
math	0.38	0.45	0.41	323	0.44	0.52	0.47	319	0.88	0.89	0.89	377
med	0.53	0.50	0.52	430	0.79	0.66	0.72	494	0.97	0.96	0.96	412
physics	0.43	0.44	0.43	434	0.57	0.60	0.58	421	0.90	0.90	0.90	440
accuracy			0.47	1714			0.62	1714			0.94	1714
macro avg	0.46	0.47	0.46	1714	0.61	0.61	0.61	1714	0.94	0.94	0.94	1714
weighted avg	0.47	0.47	0.47	1714	0.63	0.62	0.62	1714	0.94	0.94	0.94	1714

Forest Random

	заголовок				заголовок, аннотация				заголовок, аннотация, текст			
	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
IT	0.45	0.43	0.44	498	0.58	0.54	0.56	521	1.00	1.00	1.00	485
math	0.34	0.39	0.36	336	0.33	0.43	0.37	298	0.86	0.87	0.86	379
med	0.48	0.44	0.46	445	0.69	0.58	0.63	483	0.99	0.91	0.95	443
physics	0.48	0.48	0.48	435	0.45	0.49	0.47	412	0.85	0.91	0.88	407
accuracy			0.42	1714			0.52	1714			0.93	1714
macro avg	0.42	0.42	0.42	1714	0.51	0.51	0.51	1714	0.92	0.92	0.92	1714
weighted avg	0.42	0.42	0.42	1714	0.54	0.52	0.52	1714	0.93	0.93	0.93	1714

Рис. 8. Метрики качества разных моделей кластеризации с разным количеством данных для word2vec векторизации

применяем обученную модель для заголовков, аннотаций и текстов. Пример использования модели векторизации и вспомогательные функции на рисунке 7.

После этого можно обучить модели классификации и получить метрики. Результат на рисунке 8.

Качество модели стало хуже по сравнению с векторизацией текстов, основанных на статистических подходах.

Заключение

Сводный результат проведённого эксперимента показан в таблице 3.

В рамках проведённого исследования были проанализированы различные методы векторизации текстов для многозадачной классификации научных статей. Основной целью работы было сравнение метрик методов векторизации в зависимости от объёма входных данных и выбранных моделей классификации.

Наибольшая точность классификации (до 97 %) достигается при использовании полного текста статей вместе с заголовками и аннотациями. Это подтверждает, что увеличение объёма данных значительно улучшает качество моделей. Мешок слов (CountVectorizer) и TF-IDF показали схожие результаты, с небольшим преимуществом TF-IDF в задачах классификации. Оба метода демонстрируют высокую точность при использовании полного текста статей. Word2Vec, несмотря на учёт семантической близости слов, показал более низкие результаты по сравнению с методами, основанными на статистических подходах.

Также проведённое исследование позволяет сделать вывод про качество моделей классификации, несмотря на то что это не являлось основной задачей: у модификация логистической регрессии метрики лучше по сравнению со случайным лесом в большинстве сценариев.

Таким образом, для задач классификации научных текстов, где важна точность и интерпретируемость результатов, рекомендуется использовать методы векторизации, основанные на статистических подходах.

Таблица 3.

Метрики точности для многозадачной классификации научных текстов с использованием разных методов векторизации текстов

Векторизация	Данные	Модель	accuracy	Macro avg	Weighted avg
Мешок слов	заголовок	LogisticRedressionCV	0.77	0.77	0.77
		RandomForestClassifier	0.68	0.67	0.68
	заголовок, аннотация	LogisticRedressionCV	0.83	0.83	0.83
		RandomForestClassifier	0.73	0.73	0.73
	заголовок, аннотация, текст	LogisticRedressionCV	0.97	0.97	0.97
		RandomForestClassifier	0.96	0.96	0.96
TF-IDF	заголовок	LogisticRedressionCV	0.77	0.77	0.77
		RandomForestClassifier	0.70	0.70	0.71
	заголовок, аннотация	LogisticRedressionCV	0.83	0.83	0.83
		RandomForestClassifier	0.72	0.72	0.73
	заголовок, аннотация, текст	LogisticRedressionCV	0.97	0.97	0.97
		RandomForestClassifier	0.96	0.96	0.96
Word2vec	заголовок	LogisticRedressionCV	0.47	0.46	0.47
		RandomForestClassifier	0.42	0.42	0.42
	заголовок, аннотация	LogisticRedressionCV	0.62	0.61	0.62
		RandomForestClassifier	0.52	0.51	0.52
	заголовок, аннотация, текст	LogisticRedressionCV	0.94	0.94	0.94
		RandomForestClassifier	0.93	0.92	0.93

ЛИТЕРАТУРА

1. Горигорян Э.Г., Паршин М.Н., Методы NLP для предобработки текстовых данных и выделения признаков // Научный журнал «Бизнес и общество» №3(31) — 2021, стр. 1–8
2. Булыга Филипп Сергеевич, Курейчик Виктор Михайлович // Сравнительный анализ методов векторизации текстовых данных большой размерности // Известия ЮФУ. Технические науки. 2023. №2 (232).
3. Корюкин, А.В. Исследование влияния настроек TF-IDF векторизации текста на результаты бинарной классификации тональности / А. В. Корюкин // Математические методы в технологиях и технике. — 2021. — № 5. — С. 126–130. — DOI 10.52348/2712-8873_MMTT_2021_5_126. — EDN DBBXDL.
4. Лыченко Н.М., Сороковая А.В., Сравнение эффективности методов векторного представления слов для определения тональности текстов // Математические структуры и моделирование 2019. №4(52) — Бишкек, Кыргызстан, стр. 97–110
5. Краснянский М.Н., Обухов А.Д., Соломатина Е.М., Воякина А.А., Компьютерная лингвистика и обработка естественного языка // Вестник ВГУ, Серия: Системный анализ и информационные технологии №3 — 2018, стр. 173–182
6. Майкл А. Лоунс, Как избежать ловушек машинного обучения: руководство для академических исследований // Школа математики и компьютерных наук — Эдинбург, Великобритания, 2024, стр. 1–33 — DOI: https://doi.org/10.48550/arXiv.2108.02497
7. Формула средневзвешенного значения // Машинное обучение в Питоне [сайт]. 2024. URL: https://www.geeksforgeeks.org/weighted-average-formula/
8. Модуль sklearn, метрики, значение классов // Библиотека Scikit-learn : Машинное обучение в Питоне [сайт]. 2025. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
9. Модуль sklearn, мешок слов // Библиотека Scikit-learn : Машинное обучение в Питоне [сайт]. 2025. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
10. Модуль sklearn, векторизация TF-IDF // Библиотека Scikit-learn : Машинное обучение в Питоне [сайт]. 2025. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
11. Батраева И.А., Нарцев А.Д., Лезгян А.С., Использование анализа семантической близости слов при решении задачи определения жанровой принадлежности текстов методами глубокого обучения // Вестник Томского государственного университета, «Управление, вычислительная техничка и информатика» №50 — Томск, 2020, стр. 14–22
12. Gensim, модели, word2vec // Библиотека Gensim: Глубокое обучение в Питоне [сайт]. 2025. URL: https://radimrehurek.com/gensim/models/word2vec.html