



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## ОТЧЕТ

по лабораторной работе № 7  
по курсу «Основы искусственного интеллекта»  
на тему: «Кластеризация»

Студент ИУ7-13М  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Орду М. А.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Строганов Ю. В.  
(И. О. Фамилия)

2025 г.

# СОДЕРЖАНИЕ

<b>1</b>	<b>Теоретическая часть . . . . .</b>	<b>3</b>
1.1	Постановка задачи . . . . .	3
1.2	Предобработка текстовых данных . . . . .	3
1.2.1	Выделение словоформ . . . . .	3
1.2.2	Выделение начальных форм слов . . . . .	3
1.3	Векторизация документов . . . . .	4
1.3.1	One-Hot Encoding . . . . .	4
1.3.2	Term Frequency-Inverse Document Frequency . . . . .	4
1.3.3	N-граммы . . . . .	5
1.4	Методы кластеризации . . . . .	5
1.4.1	Метод k-средних . . . . .	5
1.4.2	Метод c-средних . . . . .	6
1.4.3	Метод Гат-Гевы . . . . .	7
1.5	Метрики оценки качества кластеризации . . . . .	8
<b>2</b>	<b>Практическая часть . . . . .</b>	<b>9</b>
2.1	Цель и задачи эксперимента . . . . .	9
2.2	Описание набора данных . . . . .	9
2.3	Предобработка текста и векторизация . . . . .	9
2.4	Создание векторных представлений . . . . .	10
2.5	Результаты . . . . .	10

# 1 Теоретическая часть

## 1.1 Постановка задачи

В данной лабораторной работе решается задача кластеризации текстовых документов. Для заданного набора текстов необходимо:

1. Преобразовать текстовые документы в векторное представление;
2. Применить методы кластеризации: k-средних, с-средних и Гат-Гевы;
3. Проанализировать качество кластеризации при различном количестве кластеров, определить среднее внутрикластерное расстояние и среднее межкластерное расстояние для каждого рассматриваемого случая.

## 1.2 Предобработка текстовых данных

### 1.2.1 Выделение словоформ

Словоформа — это конкретная грамматическая форма слова, встречающаяся в тексте. Процесс выделения словоформ из текста включает:

- Разбиение текста на отдельные слова;
- Приведение к нижнему регистру;
- Удаление служебных частей речи;
- Удаление пунктуации и специальных символов.

В данном подходе слова сохраняются в том виде, в котором они встречаются в тексте.

### 1.2.2 Выделение начальных форм слов

Лемматизация — процесс приведения слова к его начальной форме (лемме). В рамках этой работы применялась библиотека `py morphology3`, которая:

- Определяет часть речи слова

- Учитывает морфологические характеристики (род, число, падеж и др.)
- Возвращает нормальную форму слова

Преимущество лемматизации можно назвать уменьшение размерности пространства признаков за счет объединения различных форм одного слова. В нашем случае, лемматизация приводит к уменьшению размерности вектора текстового документа.

## 1.3 Векторизация документов

После предобработки тексты преобразуются в числовые векторы. Преобразовать текст можно различными способами. В рамках работы рассмотрим статические методы векторизации.

В основе статистических подходов лежит парадигма «Мешка слов», которая абстрагируется от порядка слов и рассматривает текст как неупорядоченную коллекцию терминов. Эти методы основаны на подсчёте частот слов, что приводит к созданию векторных представлений высокой размерности. Несмотря на свою простоту и интерпретируемость, они обладают общими недостатками, такими как семантическая слепота, пренебрежение порядком слов и проблема разреженности данных.

### 1.3.1 One-Hot Encoding

Самый просто и примитивный метод векторизации текста, результатом которого является матрица с единицами и нулями внутри. 1 говорит о том, что какой-то текстовый элемент встречается в предложении (или в нашем случае документе). 0 говорит о том, что элемент не встречается в предложении.

### 1.3.2 Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency или сокращенно TF-IDF — один из наиболее распространённых и эффективных статистических методов. Вес слова вычисляется как произведение двух компонент: количество раз, когда слово встретилось в документе и натурального логарифма от количества документов деленное на количество документов содержащее

этот символ. Формула для расчета представлена в (1.1).

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{df_i} \quad (1.1)$$

где:

- $tf_{i,j}$  — количество раз, когда слово  $i$  встретилось в документе  $j$ ;
- $df_i$  — количество документов содержащее символ  $i$ ;
- $N$  — общее количество документов.

Логарифмическая мера снижает влияние служебных частей речи и повышает значимость слов, характерных для конкретного документа.

### 1.3.3 N-граммы

Для учёта локального контекста и фиксации устойчивых словосочетаний применяется расширение классического подхода — N-граммы. Элементами векторизации становятся не отдельные слова (униграммы), а последовательности из N соседних слов или символов. Это позволяет моделировать такие выражения, как «машинное обучение», в виде единого семантического токена. Однако использование N-грамм приводит к комбинаторному росту размерности пространства признаков, что создаёт серьёзные вычислительные сложности и требует больше данных для обучения моделей.

## 1.4 Методы кластеризации

### 1.4.1 Метод k-средних

K-средних — классический метод четкой кластеризации. Четкая кластеризация — кластеризация с заранее известным количеством кластеров. Алгоритм можно описать следующим образом:

1. Начальный выбор координат центроидов  $k$  кластеров/ Выбор, как правило, носит случайный характер, однако существуют модификации метода, где начальный выбор выполняется на основе поиска максимально отдалённых друг от друга потенциальных центроидов кластеров;

2. Назначение каждого объекта ближайшему центроиду;
3. Пересчет центроидов кластеров;
4. Повторение шагов 2-3 до сходимости.

Целевая функция:

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (1.2)$$

### 1.4.2 Метод с-средних

Метод с-средних – метод нечёткой кластеризации, где каждый объект принадлежит всем кластерам с определённой степенью принадлежности от 0 до 1.

Метод минимизирует взвешенную сумму квадратов расстояний:

$$J_m(U, V) = \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m d_{ij}^2 \quad (1.3)$$

где:

- $d_{ij} = \|\mathbf{x}_j - \mathbf{v}_i\|$  – евклидово расстояние
- $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\}$  – множество центроидов
- $m > 1$  – параметр нечёткости.

Алгоритм метода с-средних можно описать следующим образом.

1. Пересчёт центров кластеров. Центр каждого кластера вычисляется как взвешенное среднее всех точек данных. Весом для точки служит её степень принадлежности к этому кластеру, возведённая в степень  $m$ . Таким образом, точки с высокой степенью принадлежности сильнее влияют на положение центра;
2. Для каждой точки данных заново вычисляется её принадлежность ко всем кластерам. Степень принадлежности обратно пропорциональна расстоянию от точки до центра кластера: чем точка ближе к центру, тем выше её принадлежность к этому кластеру. Для одной точки сумма всех принадлежностей равна 1;

3. Вычисления прекращаются, когда изменения в матрице принадлежностей между двумя последовательными итерациями становятся меньше заданного порога  $\varepsilon$ ;

### 1.4.3 Метод Гат-Гевы

Метод Гат-Гевы — модификация алгоритма с-средних, который используется адаптивная метрика расстояния (*англ. fuzzy maximum likelihood estimation, FMLE*) вместо стандартной евклидовой.

Алгоритм можно описать следующим образом:

1. Случайным образом задаются начальные степени принадлежности точек к кластерам;
2. Центроиды вычисляются как взвешенные средние всех точек, где веса — степени принадлежности в степени  $m$ ;
3. Для каждого кластера вычисляется ковариационная матрица, которая описывает:
  - Направление наибольшего разброса точек;
  - Степень вытянутости кластера;
  - Ориентацию кластера в пространстве.
4. Расстояние от точки до центроида кластера вычисляется с учётом его формы. Точка может быть ближе к центру вытянутого кластера по его длинной оси, даже если евклидово расстояние велико;
5. Степени принадлежности пересчитываются на основе новых расстояний;
6. Процесс повторяется, пока изменения в матрице принадлежностей не станут меньше заданного значения.

## 1.5 Метрики оценки качества кластеризации

**Среднее внутрикластерное расстояние** показывает насколько похожи объекты внутри одного кластера. Вычисляется по формуле (1.4):

$$W = \frac{1}{N} \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\| \quad (1.4)$$

где  $N$  — общее количество объектов,  $\mu_i$  — центроид кластера  $C_i$ .

**Среднее межкластерное расстояние** показывает насколько хорошо кластеры отделены друг от друга. Вычисляется по формуле (1.5):

$$B = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^k \|\mu_i - \mu_j\| \quad (1.5)$$

где  $k$  — число кластеров.



## 2 Практическая часть

### 2.1 Цель и задачи эксперимента

Целью работы является исследование эффективности методов кластеризации Гат-Гевы, К-средних и С-средних на размеченном наборе текстовых данных. Для этого необходимо:

1. Предобработать тексты и создать векторные представления документов двумя методами;
2. Провести кластеризацию с использованием указанных методов, варьируя количество кластеров  $k$ ;
3. Рассчитать метрики внутрикластерного и межкластерного расстояний для оценки качества разбиения;
4. Сравнить полученные кластеризации с экспертной разметкой.

### 2.2 Описание набора данных

Работа проводилась на предоставленном размеченном наборе текстовых данных. Набор содержит  $N = 50$  текстовых документов, относящихся к  $K = 7$  тематическим кластерам согласно экспертной разметке.

### 2.3 Предобработка текста и векторизация

Векторизация документов выполнялась в два этапа: предобработка текста и создание числового вектора.

В ходе первого этапа исходный текст очищался от знаков препинания, символов перевода строк и приводился к нижнему регистру. После этого каждый документ разбивался на отдельные слова (токены) с помощью простого разделителя по пробелам.

Для получения начальных форм слов использовалась библиотека `ru morphology3`. Каждый токен, полученный на предыдущем этапе, преобразовывался в свою нормальную форму (лемму).

## 2.4 Создание векторных представлений

После лемматизации для каждого из двух методов (мешок словоформ и мешок лемм) выполнялась векторизация. В результате каждый документ  $d_i$  был представлен как вектор  $\mathbf{x}_i$ , где каждая компонента соответствовала весу конкретного слова (словоформы или леммы) в документе. Служебные части речи при векторизации не учитывались.

## 2.5 Результаты

## ЗАКЛЮЧЕНИЕ