Санкт-Петербургский государственный университет

Кафедра системного программирования

Волков Григорий Валерьевич

Технология извлечения информации из последовательности текстов для определения их стиля

Курсовая работа

Научный руководитель:
профессор О. Н. Граничин

Санкт-Петербург
2018

Оглавление

Введение

Разбиение текстов на группы — классическая задача машинного обучения. В качестве признака, по которому производится классификация может выступать жанр, тематика или же авторство. Но классификация возможна лишь при наличии заранее известных жанров, тем, авторов и соответственно размеченных данных. Если же мы хотим формализовать понятие стиля письма, который является индивидуальной чертой писателя, следует решать в каком-то смысле более общую задачу — задачу кластеризации.

Это довольно молодая и малоизученная задача. Большинство статей [1][2], посвященных ей, опубликованы не ранее 2010 года.

Решение этой задачи позволяет не только определять авторство документа, но и получить новый способ его векторизации.

1. Цель работы

Задача заключается в построении прототипа алгоритма, его последующее использование для реалищации и тестирования. Были выделены пнукты, описывающие цель работы:

- Построить модель алгоритма, который векторизует текст так, что векторные представления текстов одного автора похожи.
- Выразить в описанных терминах существующие решения и предложить свои.
- Настроить тестирование моделей.

2. Модель

Предложенная модель алгоритма состоит из трёх частей:

1. Препроцессинг текста (text preprocessing).
2. Векторизация характеристик (feature vectorization).
3. Обучающийся алгоритм (learning algorithm).

2.1. Препроцессинг текста

Препроцессинг текста представляет из себя композицию отображений, называемых preprocessing item. Каждое такое отображение относится к одному из следующих видов:

- Из текста в текст.
  Например, перевод букв в нижний регистр.
- Из текста в список токенов
  Например, разбиение текста на слова.
- Из списка токенов в список токенов.
  Иными словами - фильтрация токенов. Например, удаление стоп-слов[1].

2.2. Векторизация характеристик

После препроцессинга, переводящего текст в список токенов, необходимо векторизовать документ. Иногда векторное представление зависит от контекста, т. е. от других текстов. Поэтому это преобразование имеет следующий вид:
Вход: множество текстов, представленных в виде множества токенов: $\{A_i \mid i = 1...n]\}$, где $A_i = \{a_j \mid j = 1, ...m_i\}$, а a_j — токен.
Выход: матрица V, где строка $V_{i.}$ интерпретируется как вектор, соответствующий тексту A_i.

[1] **Стоп-слова** — слова, несущие малую смысловую нагрузку

2.3. Обучающийся алгоритм

Алгоритм, на который накладывается несколько неформальных ограничений:

1. Он оперирует понятиями text preprocessing и feature vectorization.
2. Имеет параметры, от которых зависит ”поведение” алгоритма, и которые могут менятся в зависимости от обучающей выборки.
3. Имеет обязательный параметр - алгоритм кластеризации.

2.4. Диаграмма модели

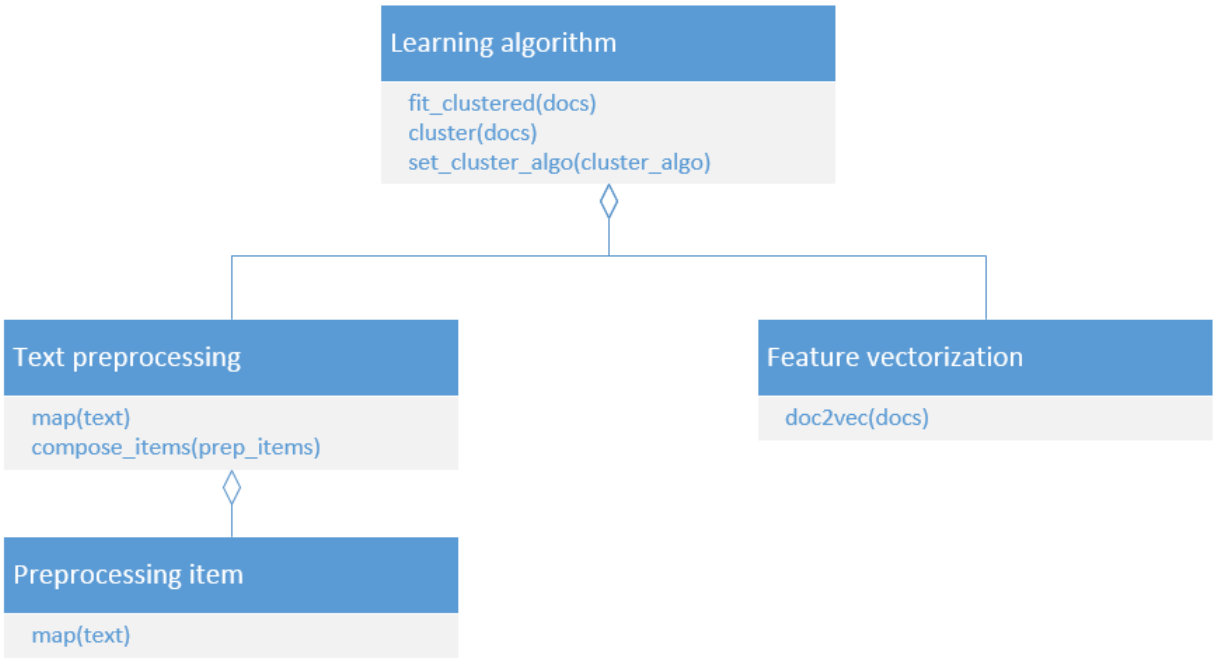

Рис. 1: Диаграмма модели

3. Инструменты

В данной секции описаны программные инструменты, которые использовались в процессе работы.

3.1. Язык

В качестве языка программирования был выбран Python по следующим причинам:

- Python широко используется для математических и статистических вычислений.
- Язык имеет множество удобных библиотек для научных вычислений.
- Простой синтаксис сильно облегчает прототипирование.

3.2. Библиотеки

В процессе работы были использованы следующие библиотеки:

- NumPy обеспечивает поддержку многомерных массивов и высокоуровневых математических функций.
- SciPy - библиотека для научных вычислений.
- NLTK (Natural Language Toolkit) - библиотека для символьной и статистической обработки естественного языка.

4. Тестирование

В тестировании являются ключевыми две вещи: выборки, представляющие из себя множество текстов, и метрика сходства полученного разбиения с эталонным.

4.1. Выборка

Были использованы следующие выборки:

1. Произведения литературы русскоязычных авторов:
   - М. А. Шолохов.
   - Л. Н. Толстой.
   - И. С. Тургенев.
   - А. С. Серафимович.
   - Ф. М. Достоевский.
2. Произведения литературы англоязычных авторов:
   - Д. Остин.
   - С. Кинг.
   - Д. Оруэлл.
   - Д. Толкин.
3. Статьи новостного агенства «Рейтер» (включают в себя по 50 статей 50 различных журналистов).

4.2. Метрики

В качестве метрик[1] качества кластеризации были выбраны:

1. Adjusted Rand index (ARI).

[1]Подробнее о метриках "успешности" кластеризации: `scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation`

2. Adjusted mutual information (AMI).

3. Fowlkes–Mallows index (FMI).

Все три обладают общим свойством: чем ближе значение к 1, тем полученное разбиение множества на кластеры ближе к истинному.

5. Реализация

В этом разделе указаны конкретные раелизации описанных ранее частей алгоритма.

5.1. Text preprocessing

5.1.1. Prep1

Перевод в нижний регистр → токенизация слов[1] → удаление стоп-слов[2] → удаление слов, содержащих символы, отличные от букв.

5.1.2. Prep2

Перевод в нижний регистр → токенезация слов.

5.2. Feature vectorization

5.2.1. NGram

Имеется алфавит Σ (например, русские буквы) и n - размер N-грамм Тогда множество $NGR = \{a_1 a_2 ... a_n \,|\, a_i \in \Sigma,\, 1 \leq i \leq n\}$ - все возможные буквенные n-граммы. Полученный вектор v имеет размер $|v| = |NGR|$. Пронумеровав элементы NGR, мы сможем каждому элементу вектора v сопоставить некоторую грамму. Пусть v_j - элемент вектора, а $gram$ - соответствующая ему грамма. Тогда v_j равен количеству встречаний $gram$ в тексте. Т.к. полученный вектор получается разреженным из-за N-грамм, встретить которые практически невозможно (например, "ъъ" в русском языке), в подсчете участвует m грамм, которые встречаются суммарно наибольшее количество раз во всех текстах. Полученный вектор v, который имеет m элементов, нормализуется. В противном случае векторное представление больших текстов и маленьких будет сильно отличаться.
Далее будем называть пару (n, m) *конфигурацией n-gram.*

[1]Разбиение текста на токены такое, что токен - слово

[2]Список стоп-слов для различных языков предоставляет NLTK

5.2.2. POSGram

NLTK дает возможность определения части речи слова. Использовав эту возможность, можно по аналогии с предыдущим пунктом определить N-граммы, где алфавитом являются части речи.

5.3. Learning algorithm

5.3.1. PTHG

PTHG (Partitioning Texts into Homogeneous Groups) [3] - алгоритм, который разбивает каждый текст на чанки, проводит над ними некоторые преобразования, кластеризует их и на основе полученной кластеризации дает ответ.

Параметры:

1. L - длина чанка.
2. T - количество чанков-предшественников.
3. Dis - метрика расстояния между двумя векторами.
4. Nconf - конфигурация n-грамм.
5. ClusterAlgorithm - алгоритм кластеризации.
6. K - количество кластеров.

Шаги алгоритма:

1. После text preprocessing токены каждого текста склеиваются в один текст.
2. Каждый текст разбивается на чанки длины L.
3. Чанки векторизируются и образуют список векторов - c.
4. Далее формируется матрица V, где $V_{i,j}$ - расстояние между c_i и c_j. Для вычисления V необходимо ввести несколько определений:

- Множество чанков-предшественников: $\Delta_{i,T} = \{c_{i-j}, j = 1, \ldots, T\}$
- $ZV_{T,Dis,L}(c_i, \Delta_{i,T}) = \frac{1}{T} \sum\limits_{c \in \Delta_{i,T}} Dis(c, c_i)$
- $DZV_{T,Dis,L}(c_i, c_j) = |A_{i,j} + A_{j,i} - A_{i,i} - A_{j,j}|$, где $A_{a,b} = ZV_{T,Dis,L}(c_a, \Delta_{b,T})$

Значения $DZV_{T,Dis,L}$ формируют матрицу V.
Замечание: пусть j - номер чанка c_i внутри текста, из которого он был взят.
Тогда c_i "участвует" в нахождении $V \iff j > T$.

5. Таким образом, в строке $V_{i,\cdot}$ содержатся расстояния от чанка c_j до всех остальных.
   Строка $V_{i,\cdot}$ рассматривается как новое векторное представление чанка c_j и все строки подаются на вход алгоритму кластеризации.

6. В конце, текст относится к тому кластеру, где оказалось большинство его чанков.

Важная часть работы - попытка улучшения алгоритма, описанная в разделе "Эксперименты"

5.3.2. Identity

Алгоритм не подразумевает обучения и наличия каких-либо параметров. По сути, его описание было дано в 5.3.

Шаги алгоритма:

1. Применение text preprocessing.

2. Векторизация текстов, используя feature vectorization.

3. Кластеризация полученных векторов.

6. Эксперименты

6.1. Identity биграммы

Наипростейшей реализацией модели является объединение Identity в качестве алгоритма, токенезацию слов в качестве препроцессинга и биграммы (2-граммы) в качестве способа векторизации текстов. Алгоритм кластеризации - K-means. На следующей гистограмме изображены результаты метрик для разного количества авторов (кластеров).

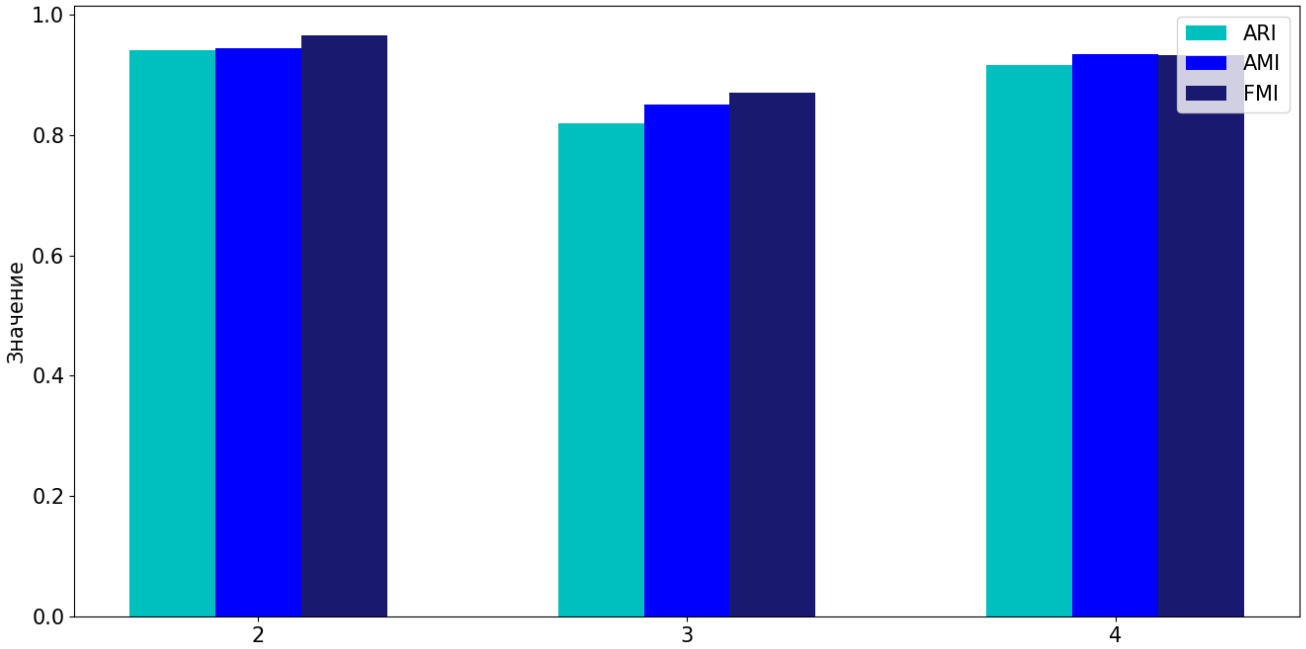

Рис. 2: Оценка качества кластеризации

6.2. Identity POS-граммы

Данная модель аналогична предыдущей. Единственное отличие - способ векторизации текстов: POS-граммы, описанные в 5.2.2. Следующая гистограмма аналогична гистограмме из предыдущего пункта.

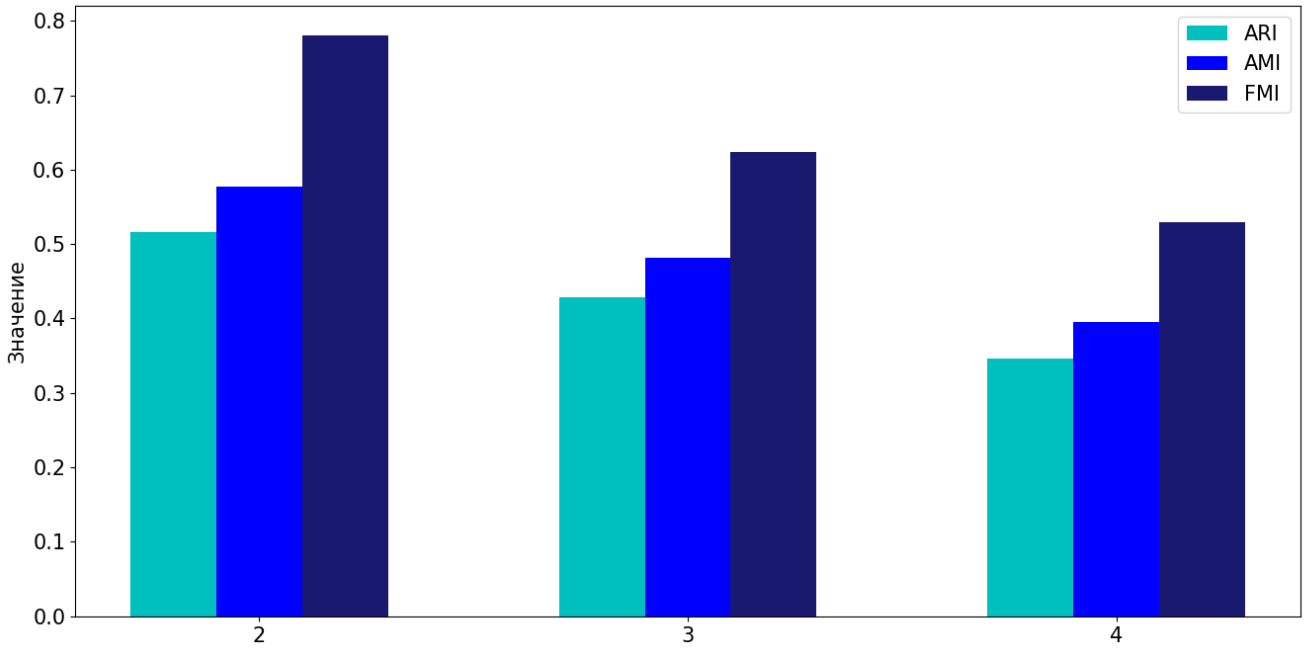

Рис. 3: Оценка качества кластеризации

6.3. Улучшение PTHG

Была использована выборка, содержащая произведения литературы. Эмпирическим путем было установлено, что оптимальные значения T и L - 15 и 1500 соответственно
Была предпринята попытка улучшения алгоритма за счет перебора других параметров с конкретными значениями:

- Алгоитм кластеризации:
  - K-medoids.
  - K-means.
  - Mini-batch k-means.
  - Agglomerative clustering.
  - Birch.
- Конфигурация N-грамм:
  - $n = 2$, $m = 50$.

- $n = 2$, $m = 100$.
- $n = 3$, $m = 50$.
- $n = 3$, $m = 100$.

- Метрика расстояния:
  - Canberra.
  - Euclidean.
  - Cosine.
  - Correlation.

В таблице представлены наборы параметров, показывающие наилучшие результаты по метрикам, описанным в 4.2.

	nconf	алгоритм кластеризации	метрика расстояния
1	(2, 100)	K-medoids	Canberra
2	(2, 100)	K-medoids	Cosine
3	(2, 100)	Mini-batch k-means	Cosine
4	(2, 100)	Agglomerative clustering	Cosine
5	(2, 100)	K-medoids	Euclidean

Таблица 1: Оптимальные параметры

На следующих графиках представлены средние метрики качества кластеризации для каждого из указанных в Таблице 1 набора параметров. На Рис. 4 - оценивается кластеризация чанков, а на Рис. 5 - исходных текстов.

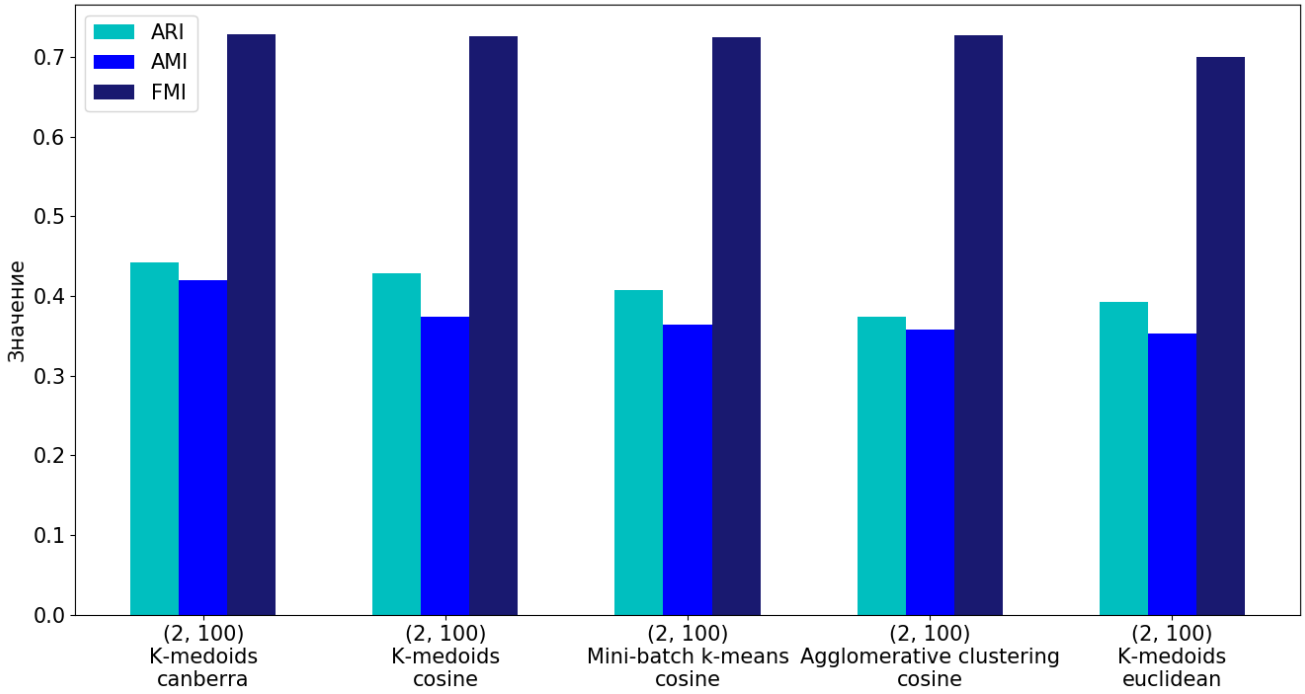

Рис. 4: Оценка качества кластеризации чанков

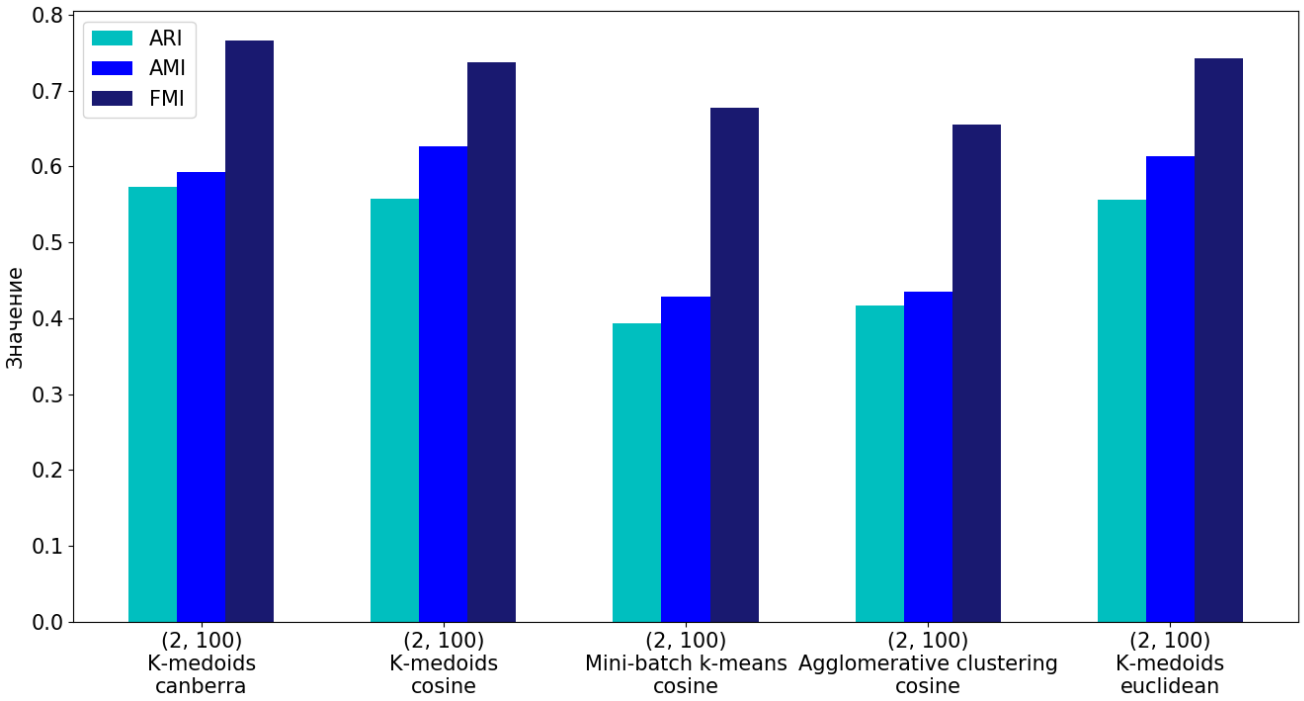

Рис. 5: Оценка качества кластеризации текстов

6.4. Кластеризация статей «Рейтера»

На графике представлена зависимость Adjusted rand index от количества авторов (кластеров) для трех вышеизложенных подходов. В качестве выборки выступали статьи новостного агенства «Рейтер».

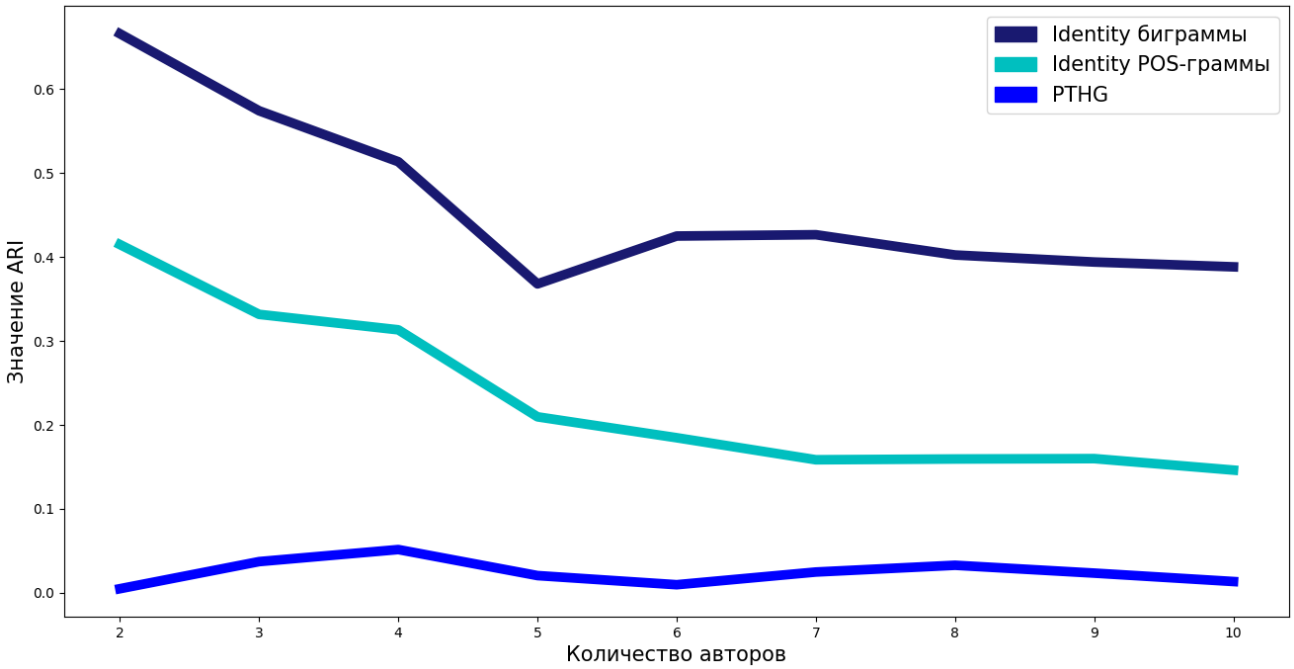

Рис. 6: Оценка качества кластеризации текстов

Заключение

В ходе работы были достигнуты следующие результаты:

- Разработана система для исследования алгоритмов кластеризации текстов по авторскому стилю.
- Были реализованы некоторые алгоритмы и улучшены существующие.
- Получены экспериментальные результаты.

Таким образом, был создан в каком-то смысле фреймворк для решения описанной задачи. Также было налажено тестирование и создано пространство для дальнейших исследований в теме идентификации автора.

Список литературы

[1] Bagnall Douglas. Text Clustering on Authorship Attribution Based on the Features of Punctuations Usage. — 2016.

[2] Jin Mingzhe1 Jiang Minghu. Authorship clustering using multi-headed recurrent neural networks. — 2012.

[3] Konstantin Amelina Oleg Granichin Natalia Kizhaeva Zeev Volkovichd. Patterning of Writing Style Evolution by means of Dynamic Similarity. — 2017.