ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

УДК _____

**Отчет об исследовательском проекте на тему**

# Поймай бота: семантические траектории текстов естественного языка

**Выполнен студентом:**

группы #БПМИ2223
Гришина Елена Романовна _____ (подпись) 25.05 (дата)

**Проверен руководителем проекта:**

Громов Василий Александрович _____ (подпись) _____ (дата)
Доктор физико-математических наук
Профессор
Профессор департамента анализа данных и искусственного интеллекта

Москва 2024

# Содержание

# Аннотация

Современный искусственный интеллект крайне ограничен в своих возможностях, и голосовые помощники могут произносить только заранее заготовленные анекдоты, но не способны придумывать свои собственные шутки. Однако люди, использующие голосовые помощники на основе искусственного интеллекта, желают от них большей человечности и, в частности, умения придумать новую шутку в ответ.

На данный момент существует не так много работ, исследующих юмор на семантическом уровне. Наша задача — провести подробное исследование семантических траекторий текстов на естественном языке с целью нахождения алгоритма, который сможет отличать ‘смешные’ тексты от ‘несмешных’. В данной работе мы представляем n-граммы векторами и фокусируемся на анализе топологической структуры текстов. На основе этих сведений планируется выявить особенности юмористических текстов и научиться генерировать шутки, учитывая выявленные характерные свойства.

# Ключевые слова

Топологический анализ, обработка естественного языка, персистентная гомология, представители гомологий, boundary matrix.

# 1 Введение

Обработка естественного языка (NLP) - направление на стыке лингвистики, математики и машинного обучения, которое исследует и обрабатывает на семантическом уровне тексты, написанные людьми. В рамках проекта "Spot the bot"мы исследуем семантические пространства языков разных групп и сравниваем их между собой.

Кроме того, в настоящее время приближение поведения разговорного ИИ к человеческому – актуальная задача, и со стороны пользователей есть запрос на более человечное поведение чат-ботов. Именно поэтому важно научить бота распознавать юмор и придумывать новый смешной ответ.

Некоторые современные исследования направлены на генерацию определенных типов юмористических текстов на основе объемных размеченных данных или используют для этой цели преимущественно методы обучения с учителем. Такие модели нестабильны, слабо обобщаются на разные виды юмора и способны только воспроизводить известные сценарии шуток. Однако, насколько нам известно, пока не существует достаточно полных исследований семантических пространств юмористических текстов естественных языков. Нахождение структуры семантического пространства юмористических текстов может помочь в генерации новых шуток.

# 2 Обзор литературы

В книге [1] подробно описан алгоритм получения SVD-эмбеддингов на основе матрицы, хранящей информацию о похожести слов и частоте их встречаемости в текстах. Кроме того, в работе рассматриваются особенности этого метода и его применимость к различным задачам. Также автор подробно описал основные принципы и приложения латентного семантического анализа (LSA) и LSM - методов, которые применяются для анализа текстов естественного языка и выявления взаимосвязей между словами или n-граммами.

Авторы работы [2] развернуто представляют процесс сбора и предобработки текстов естественных языков и получения эмбеддингов слов. Для сравнения результатов они использовали 10 языков, а для вычисления эмбеддингов применяли методы Skipgram и CBOW.

В статье [3] автор представляет одно из первых приложений персистентной гомологии к обработке естественного языка. В частности, автор предлагает эффективный алгортм подсчета чисел Бетти. Предложенный алгоритм SIFTS (Similarity Filtration with Time Skeleton) выявляет дыры, которые могут быть интерпретированы как семантические "связи"в тексто-

вом документе. Автор сконцентрировался на подсчете дырок размерностей 0 и 1 и временах их жизни (временем рождения дырки он считал $\epsilon$, при котором она впервые появлялась в комплексе Вьеториса-Рипса, моментом смерти - $\epsilon$, при котором переставала существовать). Также для нескольких коротких текстов он построил персистентные диаграммы, которые содержат информацию о числах Бетти на их пространствах.

Среди существующих программ Eirene [4] справляется с вычислениями репрезентативности, постоянных гомологий и представительных циклов (representative cycles) эффективнее, чем стандартный алгоритм [5]. Однако он не использует структуру когомологии. Представители гомологии, вычисленные с помощью упомянутых подходов, обычно являются исходными представлениями, полученными в результате вычислений персистентной гомологии.

Авторы статьи [6] описали новый метод извлечения представителей устойчивой гомологии с помощью когомологии. Они предложили вычислять персистентную когомологию и использовать полученную информацию для значительного улучшения времени работы прямых вычислений устойчивой гомологии. Этот алгоритм, примененный к фильтрациям Рипса, в целом вычисляет представителей персистентной гомологии значительно быстрее, чем стандартные методы.

В исследовании [7] авторы сосредоточились на автоматическом распознавании юмора. Авторы анализировали существующие наборы данных, обучали классификаторы на каждом из них и тестировали остальные. Они выяснили, что обучение и тестирование на одном и том же наборе данных дают хорошие результаты, но обобщающая способность моделей сильно различается. Модели, обученные на наборах данных с шутками из разных источников, демонстрируют лучшую переносимость, в то время как объем обучающих данных оказывает меньшее влияние.

# 3 Определения

**n-грамма** - последовательность из n слов, стоящих в подряд в конкретном тексте.

**Эмбеддинги** - вектора вещественных чисел, соответствующие словам (или n-граммам) естественного языка и несущие в себе информацию о характеристиках конкретного слова. Эмбеддинги строятся так, чтобы *близким* (в зависимости от решаемой задачи - по смыслу, морфологическим или другим признакам) словам (или n-граммам) соответствовали векторы, более близкие в векторном пространстве языка.

**Лемматизация** - процесс приведения слов естественного языка к леммам - началь-

ным (словарным) формам.

**TF-IDF** (TF - term frequency - частота слова, IDF - inverse document frequency - обратная частота документа) - характеристика, которая вводится в задачах NLP для оценки важности слова в наборе текстов. Вес конкретного слова по такой мере прямо пропорционален его частоте употребления в документе и обратно пропорционален его частоте употребления во всех анализируемых документах.

**SVD-разложение** матрицы $A \in \mathbb{R}^{m \times n}$ - это разложение вида $A = U\Sigma V^T$, где $U \in \mathbb{R}^{m \times n}$ и $V \in \mathbb{R}^{n \times n}$ - ортогональные матрицы, а $\Sigma \in \mathbb{R}^{n \times n}$ - диагональная матрица с вещественными числами $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$ на диагонали. Числа $\sigma_1, ..., \sigma_r$ называются сингулярными числами матрицы $A$.

**p-симплекс** - $(p+1)$-мерное обобщение треугольника (например, для $n = p+1$: $n = 0$ - точка, $n = 1$ - отрезок, $n = 2$ - треугольник, $n = 3$ - тетраэдр, ...). **p-цепь** - последовательность p-симплексов.

**Симплициальный комплекс (симплициальное пространство)** - топологическое пространство, являющееся объединением симплексов (т.е. с заданной на нем триангуляцией).

**Граница p-симплекса** (boundary of p-simplex) - набор $(p-1)$-мерных симплексов, являющихся его гранями. **Граница (boundary) для цепи** p-симплексов - это $+_2$ сумма $(p-1)$-мерных граней симплексов, входящих в эту цепь

**p-мерный цикл** (p-cycle) - p-цепь с пустой границей. **p-мерный граничный цикл** (p-boundary-cycle) - p-цикл, являющийся границей некоторой $(p+1)$-цепи.

**p-ое число Бетти** $\beta_p$ - это количество независимых p-мерных дырок симплициального пространства.

**$k$-ая граничная (boundary) матрица** $D_k$ - матрица размера $n \times m$, где n - количество симплексов размерности $k+1$, m - размерности $k$. Тогда $d_{i,j} = 1$, если симплекс $\sigma_i$ является гранью $\sigma_j$, и $d_{i,j} = 0$ иначе.

**Персистентная гомология** - это математический инструмент для анализа топологических данных, который выполняет многомасштабный анализ набора точек и выявляет в нем кластеры, дыры и пустоты. Эти топологические структуры дополняют стандартные представления признаков, что делает персистентную гомологию привлекательным инструментом извлечения признаков для искусственного интеллекта.

**Комплекс Вьеториса-Рипса** диаметра $\epsilon$ - симплициальный комплекс, в который входят симплексы, диаметр которых меньше $\epsilon$: $VR(\epsilon) = \{\sigma | diam(\sigma) < \epsilon\}$

# 4 Постановка задачи

Прежде всего, требуется получить эмбеддинги слов для естественных языков (русского, английского и кабильского). Для этого необходимо:

• Собрать корпус текстов на выбранном языке

• Произвести предобработку текстов (лемматизация, очистка от знаков препинания, замена некоторых слов на токены)

• Получить словарь языка (соответствие слово — вектор) на основе корпуса текстов

Далее, надо уметь окантовывать *'дыры'* в пространстве векторов n-грамм, полученных из текстов естественных языков, и идентифицировать их положение в пространстве и относительно других векторов. На первых этапах работы мы изучаем и разбираем существующие подходы в анализе топологической структуры таких пространств, адаптируем известные методы для нашей задачи и придумываем другие варианты решения этой задачи. Например, это можно делать, анализируя плотности в точках пространства. Другой способ получения персистентрной гомологии пространства описан в статье [6] и основан на рассмотрении boundary matrices.

Основные этапы работы:

• Разработка алгоритма для эффективного вычисления чисел Бетти на пространстве слов конкретного языка. Это тяжелая задача из-за огромного объема данных, которые необходимо обработать для получения результатов для естественного языка.

• Подбор текстов для обучения, получение и очистка корпусов языков

• Выявление персистентных гомологий в пространстве естественного языка и их представителей.

• Визуализация выделенных персистентных гомологий.

Предполагается, что шутки расположены на границах дыр в пространстве векторов, соответствующих n-граммам, в то время как нейтральные тексты находятся дальше от них. Для того, чтобы проверить эту гипотезу, требуется собрать информативный корпус текстов, который дает исчерпывающее представление о семантической структуре языка. Также необходимо научиться восстанавливать тексты по выделенным персистентным гомологиям, причем делать это так, чтобы фразы оставались юмористическими и были похожи на тексты естественного языка.

# 5 Обзор методов

## 5.1 Предварительная обработка языка

### 5.1.1 Сбор и предобработка текстов

Прежде всего, необходимо собрать тексты, написанные на естественном языке носителями. Как правило, словари эмбеддингов для языков строятся на основе литературных текстов, собранных из открытых источников, потому что именно художественная литература отражает структуру естественного языка. Для нас желаемое число исходных текстов для обработки языка - 5000-10000. Однако для ряда нераспространенных языков такое количество книг и другой художественной литературы, написанной на них, собрать невозможно, поэтому для построения их эмбеддингов в качестве текстов естественного языка могут использоваться страницы сайтов, статьи из газет и тексты новостей, которые не смешиваются с литературными текстами.

В открытом доступе может не существовать готовых решений (моделей или библиотек) для предобработки редких языков, поэтому для них необходимо самостоятельно собирать тексты для обработки, разрабатывать программы для их очистки и лемматизации. Основная задача этого этапа - очистить тексты от пунктуции и перевести формы слов в леммы: очистить тексты от падежей и склонений, привести все слова к начальной форме, а местоимения и числительные заменить на токены.

### 5.1.2 TF-IDF построение матрицы для набора текстов

Для работы с текстовыми данными требуется преобразовать слова в массивы чисел (эмбеддинги). Сначала для преобразования набора текстов естественного языка в матрицу $W$ используется метод TF-IDF, который позволяет оценить важность слова в документе относительно совокупности текстов.

Введем $tf(t,d) = \frac{n_{d,t}}{\sum_k n_{d,k}} = \frac{n_{d,t}}{n_d}$, где $n_{d,i}$ - число вхождений слова $i$ в отдельный документ $d$, а $n_d$ - общее число слов в документе $d$. Также пусть $idf(t,D) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}$, где $|D|$ - количество документов в наборе и $|\{d_i \in D | t \in d_i\}|$ - число документов, в которых встречается слово $t$. Тогда мерой TF-IDF называется произведение $tf(t,d) \cdot idf(t,D)$. TF (частота термина) измеряет, насколько часто слово встречается в документе, а IDF (обратная частота текста) - насколько уникально слово для набора текстов. Матрица $W$, построенная методом TF-IDF по корпусу собранных текстов, содержит в каждой клетке меру tf-idf для

пары слово-текст.

### 5.1.3 Получение SVD-эмбеддингов

После преобразования коллекции текстов в матрицу $W$ к ней применяется $K$-ранговое SVD-разложение. Этот метод описан в работе [1].

При применении SVD-разложения к матрице $W$ получается $W = U\Sigma V^T$, после чего векторное представление для $i$-ого слова $\hat{v_i}$ можно получить по формуле $\hat{v_i} = v_i\Sigma$, где $v_i$ - $i$-ая строка матрицы $V$.

Одно из главных примуществ использования SVD-разложения для получения эмбеддингов слов заключается в том, что достаточно один раз построить словарь векторных представлений. Тогда эмбеддинги меньших размерностей можно получать, оставляя несколько первых компонент более длинного эмбеддинга, не запуская процедуру построения эмбеддингов еще раз.

### 5.1.4 Методы понижения размерностей PCA и t-SNE

Один из основных способов понизить размерность данных с сохранением информации о пространстве - использование метода главных компонент (PCA, principal component analysis). Метод заключается в вычислении сингулярного разложения для набора исходных векторов, получении собственных значений и векторов и обрезании их до желаемой длины. Благодаря тому, что при построении эмбеддингов мы использовали метод SVD, в нашей задаче результат применения PCA будет соответствовать взятию нескольких первых компонент векторов изначального пространства.

Другой эффективный метод снижения размерностей - это t-SNE. Стохастическое вложение соседей с t-распределением (t-SNE, t-distributed Stochastic Neighbor Embedding) - это алгоритм нелинейного понижения размерностей векторов, использующийся для вложения многомерных пространств в пространства меньших размерностей (обычно 2 или 3). Неформально, метод отображает похожие объекты (расположенные рядом в исходном пространстве) так, чтобы в новых координатах они оставались рядом, а изначально далекие друг от друга точки с большой вероятностью оказывались на большем расстоянии.

## 5.2 Вычисление чисел Бетти

Наша основная задача - исследовать топологическую структуру пространства языка, поэтому мы будем рассматривать его различные характеристики. Начнем с чисел Бетти,

которые соответствуют количеству $p$-мерных 'дырок' в векторных пространствах.

Обозначим за $Z_p$ все $p$-мерные циклы, а $B_p$ - все $p$-мерные граничные циклы. Теперь пусть $H_p$ - факторгруппа $H_p = Z_p/B_p$. Напомним, что факторгруппа - группа (множество) $H_p = \{z * B_p | z \in Z_p\}$. Заметим, что циклы $B_p$ нас не интересуют, потому что они не окружают дырки, которые мы хотим считать и исследовать. Зато $H_p$ - базис p-мерных дыр пространства, то есть именно то, что мы и хотим искать и считать. Итак, p-ым числом Бетти $\beta_p$ называется $\beta_p = \mathrm{rk}(H_p) = \mathrm{rk}(Z_p/B_p) = \mathrm{rk}\, Z_p - \mathrm{rk}\, B_p$ - количество независимых p-мерных дырок пространства.

Для вычисления чисел Бетти на практике используются boundary матрицы. В p-boundary матрицах в соответствие столбцам ставятся p-симплексы комплекса, строкам - (p-1)-симплексы. Далее, на позиции $(i, j)$ в матрице положим $a_{i,j} = 1$, если $i$-ый (p-1)-симплекс является гранью $j$-ого p-симплекса. Тогда если $r_p$ - ранг p-boundary матрицы и $n_p$ - количество p-симплексов, то $\mathrm{rk}\, B_{p-1} = r_p$ и $\mathrm{rk}\, B_p = n_p - r_p$, поэтому для вычисления $Z_p$ и $B_p$ достаточно считать ранги boundary матриц.

**Algorithm 1:** Вычисление чисел Бетти

**Data:** Набор точек `points`
**Result:** Список `betti_nums` - чисел Бетти для набора точек
*Обозначим: rk2(M) - ранг матрицы M над полем* $\mathbb{Z}_2$
1. Выделить все симплексы в комплексе Вьеториса-Рипса, построенного на множестве точек для заданного $\epsilon$.
2. Построить список `all_subsets` размера `max_dim`, где `all_subsets[dim]` - список симплексов размерности `dim` $\leq$ `max_dim`. boundary_ranks = $[0]\times(max\_dim + 2)$

```
for n = 1, ...max_dim do
    cnt_nm1 = len(all_subsets[n-1])
    cnt_n = len(all_subsets[n])
    boundary_matrix=[[0 ×cnt_n] × cnt_nm1]; /* размера cnt_nm1 × cnt_n */
    for i=0,1,...,cnt_nm1; j=0,1,...,cnt_n do
        if all_subsets[n − 1][i] ∈ all_subsets[n][j] then
            boundary_matrix[i,j]=1
        end
    end
    boundary_ranks[n] = rk2(boundary_matrix)
end
for n = 0, ...max_dim do
    betti_nums[n] = len(all_subsets[n]) - boundary_ranks[n] - boundary_ranks[n+1]
end
return betti_nums
```

## 5.3 Выделение персистентных гомологий

Неформально говоря, персистентная гомология содержит информацию о ‘дырах’ пространства, включая время их появления и исчезновения, размерность и расположение. Также персистентная гомология отслеживает $\epsilon$, при которых появляются и исчезают дырки пространства комплекса

Задача построения гомологий рассматривается на заданном множестве точек в $p$-мерном пространстве с заданным на нем симплициальным комплексом. В этой работе был рассмотрен комплекс Вьеториса-Рипса.

Для получения персистентных гомологий на рассматриваемом пространстве вводится фильтрация симплициальных комплексов $K_1 \leq K_2 \leq ... \leq K_m = K$, где $K_i = K_{i-1} \cup \sigma_i$, то есть каждый комплекс $K_i$ получен добавлением нового симплекса $\sigma_i$ к предыдущему комплексу $K_{i-1}$. Тогда birth симплексом называется симплекс, при добавлении которого в комплекс $K_i$ возникает новый класс гомологии размерности $p$, а death симплексом - такой, что при его добавлении разрушается класс гомологии размерности $p-1$. Пары birth-death симплексов называются персистентными. Симплексы, которые не попали в персистентные пары, называются существенными.

Вместе с персистентными парами информативно рассмотрение представителей гомологий - $p$-цепей, которые никогда не ‘умирают’, то есть для которых не определен death симплекс. Алгоритм вычисления представителей персистентных гомологий предложен в статье [6]. Для применения алгоритма к рассматриваемому пространству на нем вводится фильтрация Вьеториса-Рипса, после чего строятся coboundary матрицы $d_k$, где $d_k = D_k^T$ - транспонированная $k$-ая boundary матрица. Далее алгоритм состоит из трех основных частей:

1. Сжатие coboundary матриц $d_k$, извлечение существенных и death симплексов. Для сжатия матриц здесь и далее используется алгоритм поиска базиса векторов матрицы над полем $\mathbb{Z}_2$, который проходит по столбцам слева направо.

2. Для каждого $k$ строятся $D_k$ - подматрицы boundary матриц, состоящие из столбцов, соответствующих существенным и death симплексам, которые были выделены на первом шаге алгоритма.

3. Сжатие матриц $D_k$ и извлечение из них представителей персистентных гомологий.

## 5.4 Разрежение пространства языка

Для разрежения пространства может быть использован Алгоритм 2. Он оставляет часть исходных точек пространства, учитывая каждую точку не более заданного числа раз

и оставляя в плотных областях те точки, которые ближе к центру скоплений.

**Algorithm 2:** Разрежение пространства языка

**Data:** Набор точек `points`
**Parameters:** max_neighbours, max_use, distance_upper_bound
**Result:** Набор точек `new_points` $\subseteq$ *points*
*Note: kdtree(points_in, points_query, k) возвращает список res размера len(points_in), где res[i] - список из $\leq k$ ближайших соседей для точки points_query[i], которые находятся на расстоянии $\leq$ distance_upper_bound от нее. Для этого используются KD-деревья.*

```
n = len(points)
cnt_used = [0 ×n]
main_kdtree = kdtree(points, points, max_neighbours)
for i = 0,..,n-1 do
    if cnt_used[i] ≥ max_use then
        continue
    end
    neighbours = ckdtree_res[i]
    for ind ∈ neighbours do
        cnt_used[ind] += 1
    end
    mean_neighbour = mean(neighbours)
    new_point = kdtree(neighbours, mean_neighbour, 1)
    new_points.append(new_point)
end
return new_points
```

# 6 Результаты

## 6.1 Предварительная обработка языка

Методология обработки естетственных языков была применена к русскому, английскому и кабильскому.

Для обработки кабильского - экзотического языка, входящего в берберскую группу - не существует готовых решений, поэтому мы полностью разрабатывали программу самостоятельно. Для этого мы собрали тексты на этом языке, написанные носителями языка, изучили грамматику кабильского, разработали лемматизатор для предобработки текста и создали словарь эмбеддингов методом SVD. Код написан на языке `Python` и доступен на github.

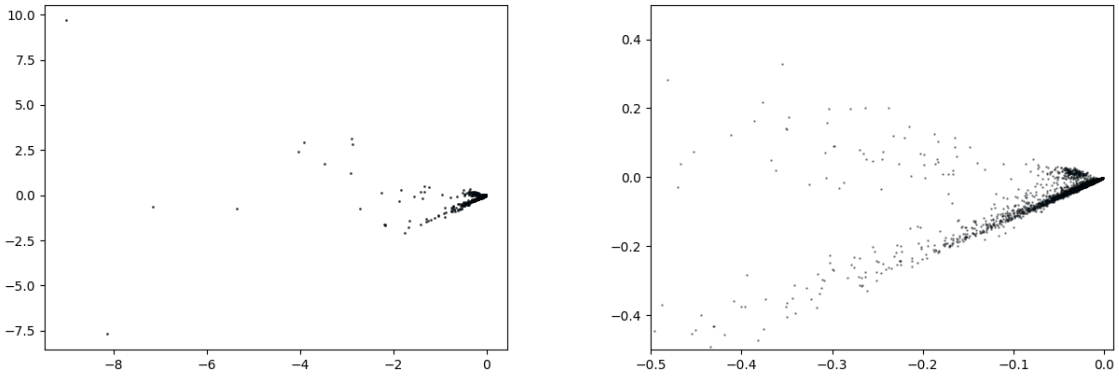
В связи с отсутствием достаточного количества текстов, написанных на кабильском языке его носителями, было принято решение строить эмбеддинги на основе 6447 статей из википедии на кабильском языке [8] и не смешивать их с литературными текстами. Отдельное

внимание в процессе парсинга Википедии уделялось тому, чтобы не сохранять в качестве текстов отдельные слова или словосочетания, которые могли выступать в статьях в качестве заголовков, а выделять на страницах сайта абзацы текста. Эта особенность связана с тем, что для получения SVD-эмбеддингов необходимы цельные тексты, поэтому корпусы, состоящие из отдельных фраз или предложений, не подходят.

На следующем этапе обработки кабильского языка мы изучали его грамматику и консультировались с носителем, проверяя правильность разработанных правил лемматизации. В результате этого этапа были получены предобработанные тексты на кабильском языке.

Далее мы преобразовали набор текстов естественного языка в матрицу методом TF-IDF и применили к ней SVD-разложение для получения эмбеддингов. В качестве ранга разложения $r$ было использовано $r = 1024$ (большое число, но меньшее количества уникальных слов (6424) и текстов (6447)). Этот подход выгоден потому, что вектора размерностей $d < r$ легко получать, взяв первые $d$ компонент в уже посчитанных векторах и не запуская процедуру вычисления еще раз. Таким образом, мы получили словарь эмбеддингов для кабильского языка - список уникальных слов из собранных текстов и векторов для них.

После получения эмбеддингов для слов кабильского языка мы визуализировали их в 2d. Оставив первые 2 координаты 1024-мерных векторов эмбеддингов в качестве $x$ и $y$ были получены графики, приведенные на Рисунке 6.1.




(a) SVD визуализация

(b) Увеличенная SVD визуализация

Рис. 6.1: SVD визуализация кабильского

Видно, что в данном случае визуализация с применением метода PCA для редукции размерностей не отражает семантической структуры языка: точки эмбеддингов слишком скучены и ‘склеены’. По этой причине мы использовали метод t-SNE с последующей визуализацией. Отметим, что алгоритм t-SNE является неустойчивым, поэтому визуализации одного и того же языка могут отличаться от запуска к запуска, однако основная семантическая структура не меняется. На рисунке 6.2 показаны несколько разных визуализаций

кабильского языка с применением t-SNE.

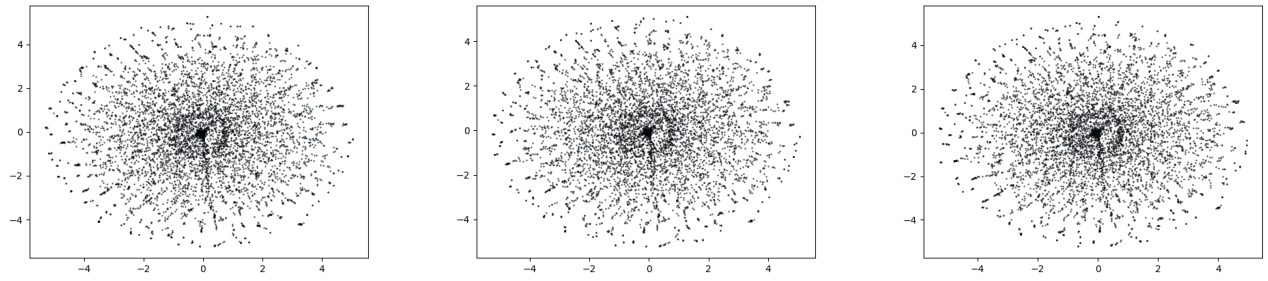
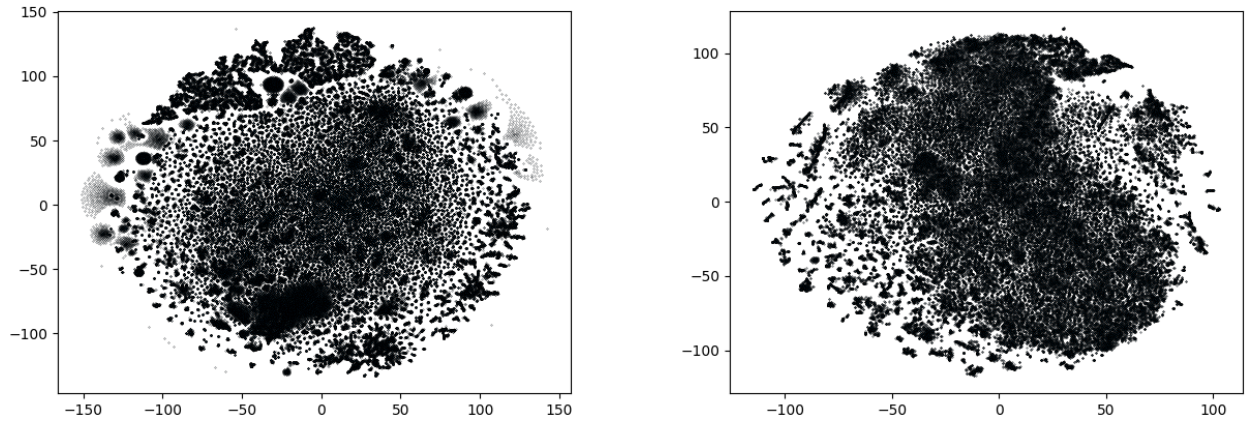


Рис. 6.2: Визуализация кабильского с применением t-SNE

Результаты визуализации русского и английского языка представлены на Рисунке 6.3.



(a) русский язык

(b) английский язык

Рис. 6.3: Визуализация языков с применением t-SNE

## 6.2 Вычисление чисел Бетти

Был имплементирован Алгоритм 1 для вычисления чисел Бетти. Чтобы проверить, что алгоритм работает правильно, он был применен к некоторым стандартным топологическим объектам, для которых числа Бетти известны: сфере, тору, тору ‘с ушками’. Полученные результаты приведены в таблицах 6.1 и 6.2.

Однако применить алгоритм к пространству языков не удалось из-за большого количества точек (например, для русского языка - $2 \cdot 10^5$).

## 6.3 Выделение персистентных гомологий

Мы стремимся исследовать ‘дырки’ всех размерностей (вплоть до размерности построенного пространства) для большого объема точек в пространстве (в нашем датасете 207928

| $\epsilon$ | $\beta_0$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|
| $\leq$0.6 | 100 | 0 | 0 |
| 0.7 | 98 | 0 | 0 |
| 1 | 86 | 0 | 0 |
| 1.5 | 1 | 5 | 0 |
| 1.6 | 1 | 1 | 0 |
| $[1.7, 5]$ | **1** | **0** | **1** |
| $\geq$5.1 | 1 | 0 | 0 |

Сфера, $n = 100, r = 3$

| $\epsilon$ | $\beta_0$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|
| 0.1 | 794 | 0 | 0 |
| 0.5 | 1 | 86 | 0 |
| 0.8 | 1 | 3 | 0 |
| 0.9 | 1 | 2 | 0 |
| $[1, 1.7]$ | **1** | **2** | **1** |
| 1.8 | 1 | 0 | 1 |
| $\geq$1.9 | 1 | 0 | 0 |

Тор, $n = 1000, r = 1, R = 2$

Таблица 6.1: Числа Бетти для точек на топологических объектах

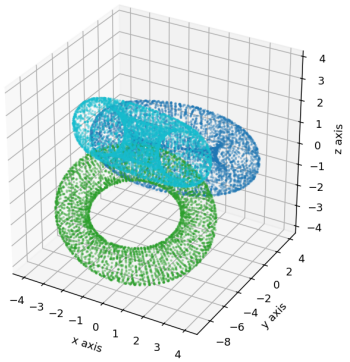



Рис. 6.4: Тор с ушками

| $\epsilon$ | $\beta_0$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|
| 0.5 | 8 | 252 | 1 |
| 0.6 | 1 | 177 | 2 |
| 0.9 | 1 | 23 | 2 |
| 1 | 1 | 6 | 2 |
| $[1.1, 1.2]$ | 1 | 3 | 3 |
| 1.3 | 1 | 3 | 4 |
| $[1.4, 1.5]$ | 1 | 2 | 3 |
| $[1.6, 1.7]$ | 1 | 2 | 2 |
| $[1.8, 2]$ | 1 | 2 | 1 |
| $[2.1, 2.3]$ | 1 | 2 | 0 |
| ... | | | |

Таблица 6.2: Числа Бетти для ‘тора с ушками’

слов русского языка, поставленные в соответствие 5-мерным векторам). Главная проблема, которая встает на нашем пути, связана с огромным количеством точек, которые необходимо обработать и на которых надо построить комплекс Вьеториса-Рипса для дальнейшего вычисления чисел Бетти. Однако нет необходимости строить полный комплекс - достаточно знать p-симплексы для $p < 5$ ($p <$ размерности нашего пространства). Кроме того, формирование boundary матриц занимает много времени (квадрат от количества симплексов соответствующих размерностей).

Другая проблема, возникшая при разработке алгоритма, заключалась в необходимости хранить большой объем данных. Заметим, что построенные boundary матрицы будут разреженными: в каждом столбце отдельной матрицы очень мало единиц, потому что в любой $p$-симплекс входит лишь $p$ $(p-1)$-симплексов. Это значит, что для оптимизации хранения boundary матрицы можно использовать разреженные матрицы, сохраняя лишь индексы единиц в них.

Для оптимизации вычисления персистентных гомологий метод был применен к частям языка и к разреженному пространству языка. На Рисунке 6.5 отображены результаты окантовки дыр на различных частях полного пространства русского языка, разными цветами

показаны различные выявленные дыры (представители персистентных гомологий).

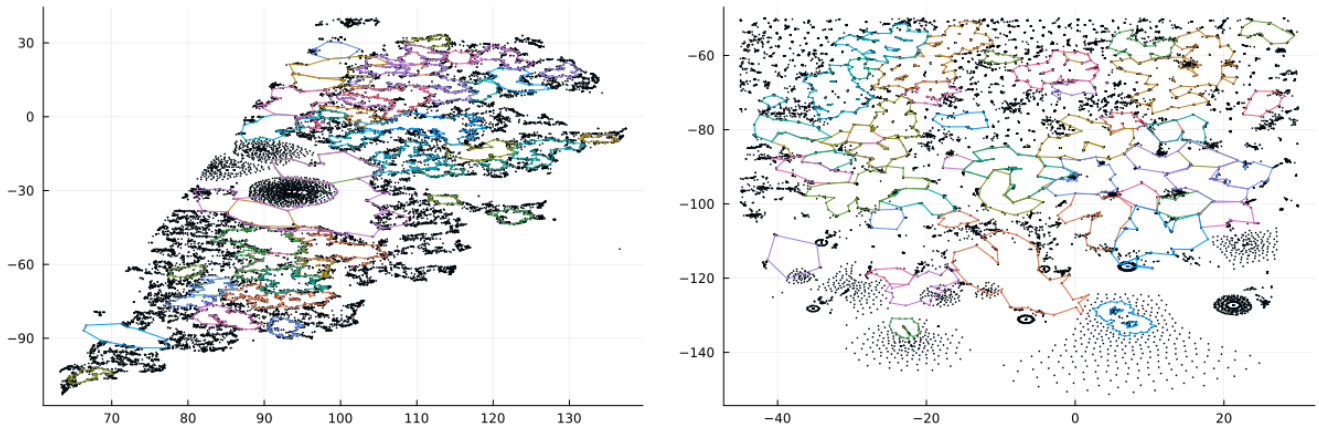
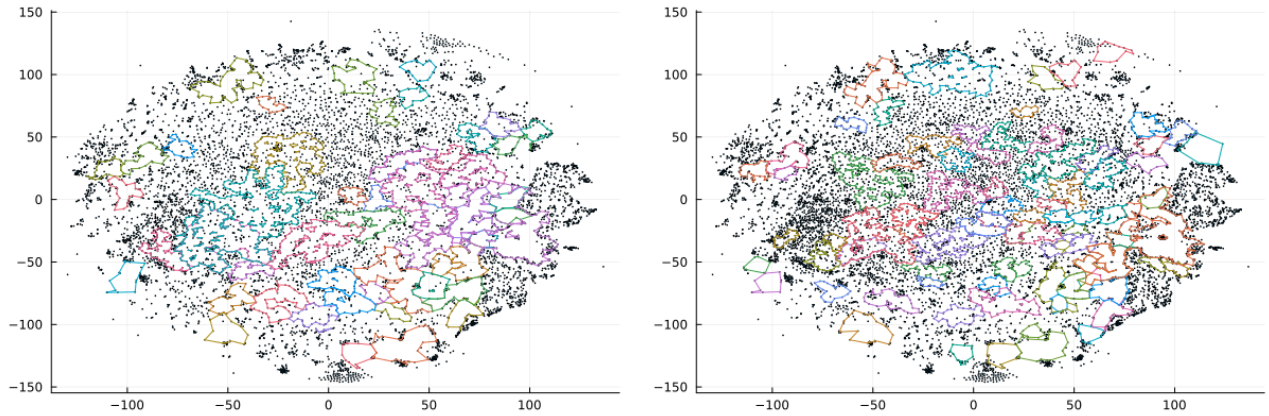



Рис. 6.5: Дырки на частях полного пространства русского языка

Также представители персистентных гомологий были найдены на разреженном пространстве русского языка, построенном методом, описанным в главе 5.4. Результаты показаны на Рисунке 6.6, разными цветами окантованы различные дыры пространства.




(a) 17890 точек, 50 дыр

(b) 25245 точек, 80 дыр

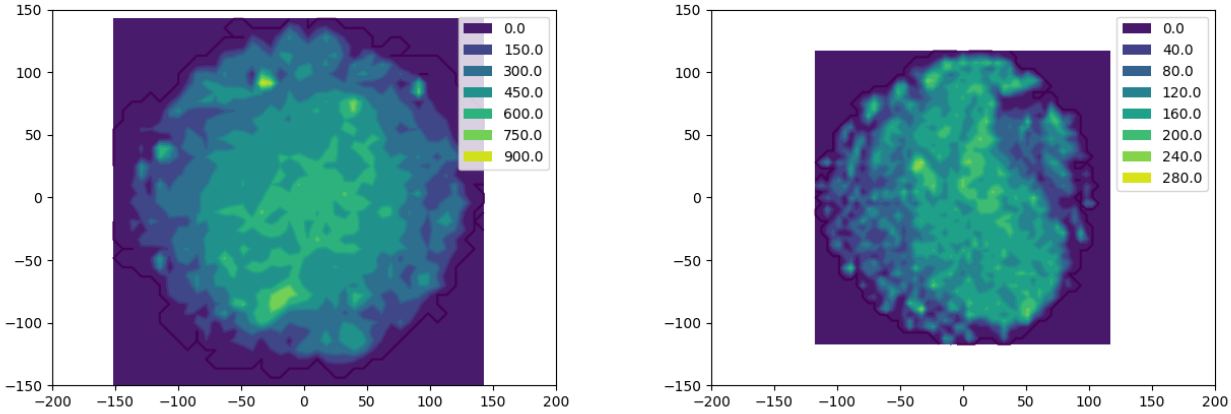
Рис. 6.6: Наиболее важные дыры на разреженном пространстве русского языка

Мы планируем сравнить дыры, выделенные на разреженном пространстве языка с дырами, полученными на частях полного пространства, и оценить, насколько полученные результаты отличаются.

## 6.4 Разрежение пространства языка

В двумерных пространствах русского и английского языка, которые были получены методом t-SNE, были насчитаны плотности. Для этого в пространстве были выделены точки на пересечении сетки размера $bins \times bins$, для каждой из которых было посчитано количество

точек пространства, попадающих в круг радиуса $r$ с центром в выбрынной точке с сетки. Для русского языка использовались параметры $r = 6, bins = 40$, для английского $r = 4, bins = 50$ Результаты представлены на Рисунке 6.7, в легенде указано количество точек, попадающих в $r$-окрестность точки.




(a) русский язык

(b) английский язык

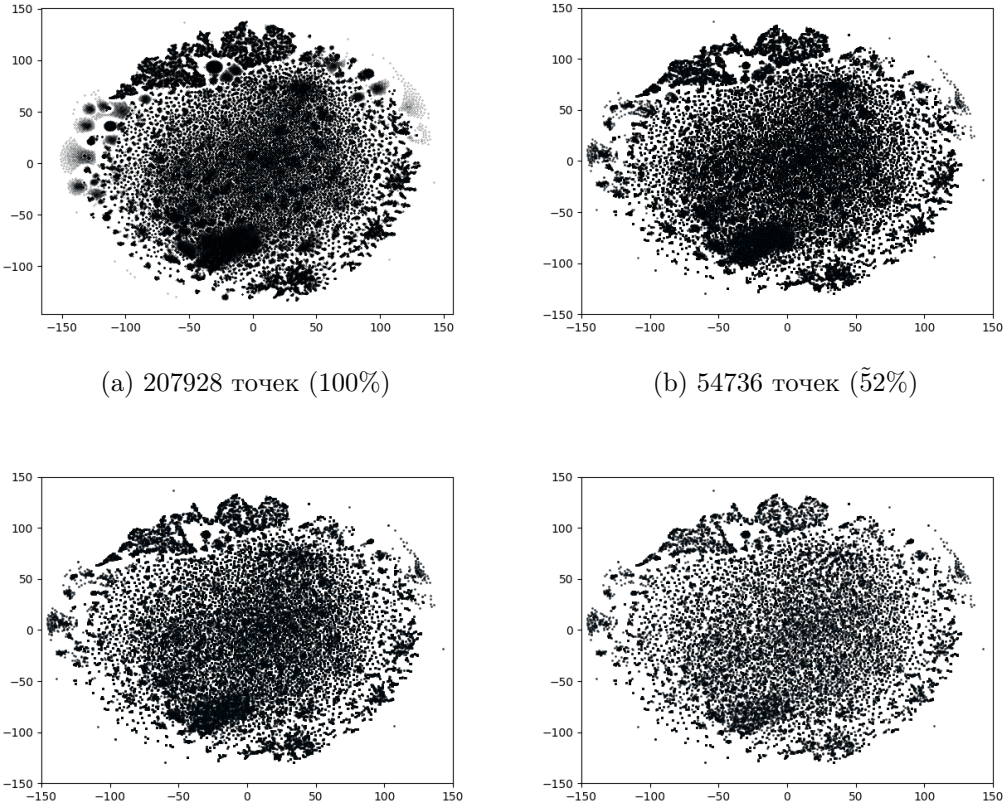
Рис. 6.7: Плотности пространств языков в 2d

В корпусе русского языка, обработанного методом, описанным в Главе 5.1, было получено 207928 ($2 \cdot 10^5$) различных слов. Из-за большого объема данных обработка пространства естественного языка является сложной задачей, требующей огромных ресурсов памяти и времени. Чтобы считать числа Бетти и выделять персистентные гомологии на пространстве, было решено уменьшить количество обрабатываемых точек Алгоритмом 2. Визуализации полученных разреженных пространств для русского языка представлены на Рисунке 6.8. Визуально семантическая структура пространства сохраняется.

При сжатии пространства точек Алгоритмом 2 точки скопления точек в среднем сохраняются, поэтому мы предполагаем, что персистентные гомологии на сжатом пространстве и на исходном будут близки. Чтобы проверить эту гипотезу, планируется применить методологию к меньши наборам точек и сравнить результаты для них. Также отдельного исследования требует подбор наиболее информативных комбинаций параметров `max_neighbours, max_use, distance_upper_bound` для разрежения пространств.

# 7 Направления дальнейшей работы

Мы планируем продолжить анализ пространства слов русского языка, провести исследования для других языков и сравнить результаты, полученные на пространствах слов различных естественных языков. Также на основе результатов, полученных из анализа то-



(a) 207928 точек (100%)

(b) 54736 точек (5̃2%)

(c) 54736 точек (2̃6%)

(d) 29267 точек (1̃4%)

Рис. 6.8: Визуализация полного и разреженных пространств русского языка

пологической структуры различных текстов, планируется генерировать шутки (фразы, расположенные около границ дыр пространства n-грамм), учитывая выявленные характерные свойства юмористических текстов.

# 8 Использованные ресурсы

Данное исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ [9].

# Список литературы


[1] Jerome R. Bellegarda. *Latent Semantic Mapping: Principles Applications*. Morgan Claypool Publishers, 2006.

[2] Edouard Gravea, Piotr Bojanowski, Prakhar Gupta, Armand Joulin и Tomas Mikolov. “Learning Word Vectors for 157 Languages”. В: ().

[3] Xiaojin Zhu. “Persistent Homology: An Introduction and a New Text Representation for Natural Language Processing”. В: (2013).

[4] Gregory Henselman и Robert Ghrist. “Matroid filtrations and computational persistent homology”. В: (2016).

[5] Letscher, Edelsbrunner и Zomorodian. “Topological persistence and simplification”. В: (2002).

[6] Matija Čufar и Žiga Virk. “Fast computation of persistent homology representatives with involuted persistent homology”. В: (2021).

[7] Alexander Baranov1, Vladimir Kniazhevsky1 и Pavel Braslavski. “You Told Me That Joke Twice: A Systematic Investigation of Transferability and Robustness of Humor Detection Models”. В: (2023).

[8] *Kabyle wikipedia*. URL: https://kab.wikipedia.org/wiki/Asebter_agejdan.

[9] Kostenetskiy P.S., Chulkevich R.A. и Kozyrev V.I. “HPC Resources of the Higher School of Economics”. В: *Journal of Physics: Conference Series* (2021). URL: https://doi.org/10.1088/1742-6596/1740/1/012050.