Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
«Прикладная математика и информатика»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Исследовательский проект на тему «Spot the bot: семантические траектории текстов естественного языка»

Выполнила студентка группы БПМИ182, 4 курса,
Данг Куинь Ньы

Руководитель ВКР:
д.ф.-м.н., профессор ДАДИИ ФКН, Громов Василий Александрович

Москва 2022

Содержание

Аннотация

Аннотация

С развитием генеративных ботов и появлением таких моделей, как Generative Pretrained Transformers-3 (GPT-3), становится всё сложнее различить человеческие тексты от сгенерированных. В большинстве работах по обнаружению ботов используются методы обучения с учителем (supervised learning). При этом, поскольку боты распространены в социальных сетях, многие решения проблемы идентификации ботов опираются на мета-данные (взаимодействия аккаунтов, частота постов и т.п.). В данной работе мы фокусируемся на текстовых данных и проводим ряд эмпирических экспериментов с целью анализа семантического пространства текстов. На основе результатов мы выделяем наиболее важные признаки, которые можно использовать для обнаружения ботов. Для анализа используются алгоритмы кластеризации Уишарта и K-Means, а также метод построения казуальной плоскости энтропии-сложности. В качестве данных мы рассматриваем векторные представления, полученные с помощью сингулярного разложения (Singular Value Decomposition, SVD), а также Word2Vec векторов. В работе мы также экспериментируем с нечётким представлением данных, основанным на понятии нечёткого числа.

Ключевые слова — идентификация ботов, семантическое пространство, кластеризация, нечёткая логика, теория информации

Abstract

With the development of generative models like GPT-3, it is increasingly more challenging to differentiate generated texts from human-written ones. There is a

large number of studies that have demonstrated good results in bot identification. However, the majority of such works depend on supervised learning methods that require labelled data and/or prior knowledge about the bot-model architecture. In this work, we propose a bot identification algorithm that is based on unsupervised learning techniques and does not depend on a large amount of labelled data. By combining findings in semantic analysis by clustering (crisp and fuzzy) and information techniques, we construct a robust model that detects a generated text for any type of bot. We find that the generated texts tend to be more chaotic while literary works are more complex. We also demonstrate that the clustering of human texts results in fuzzier clusters in comparison to the more compact and well-separated clusters of bot-generated texts.

1 Введение

С развитием технологий обработки естественного языка, появилось большое количество ботов, тексты которых становится все сложнее и сложнее отличить от человеческих. Современные state-of-the-art решения используют методы, требующие размеченные данные, при этом неспособные адаптироваться к изменениям среди ботов. На данный момент существует не так много работ, использующих методы обучения без учителя (unsupervised learning), и, как правило, такие работы рассматривают конкретные виды ботов. Наша задача — провести подробное исследование семантических траекторий сгенерированных ботами текстов и художественной литературы с целью нахождения алгоритма, который сможет различить человеческие тексты от сгенерированных без каких-либо данных о самом боте.

В настоящий момент существует большое количество интересных подходов к решению задачи обнаружения ботов. В статье [1] авторы анализируют так называемые семантические характеристики эмоциональных окрасок твитов. Большое количество работ по этой задаче обучают нейронные сети на размеченных данных, так, например, в работе [2] авторам удалось построить хорошо работающую модель для обнаружения спама с ложным мнением. Модель достигает высоких результатов, однако, авторы указывают на ограничения методов обучения с учителем и необходимость исследования методов обучения без учителя. Вдохновленные результатами вышеприведенных работ, мы продолжаем изучение семантического пространства текстов уже с использованием методов, не требующих размеченных данных.

Данная работа дает общее представление о том, как тексты, написанные людьми, и тексты, созданные ботами, различаются на семантическом уровне. На основе результатов анализа мы получаем алгоритм обнаружения произвольных ботов: как простых (например, генерирующих тексты посимвольно), так и более сложных (как GPT). Проведенный нами анализ показывает, как

разные методы выделяют разные характеристики семантического пространства. Например, методом построения плоскости энтропии-сложности мы увидели, что написанные людьми тексты более хаотичные и сложные. Кластеризация показала, что для ботов получаются более компактные и разделимые кластера.

В следующем разделе мы даём постановку задачи работы. В третьем разделе проведен обзор существующих методов решения проблемы обнаружения ботов. Четвертый раздел содержит теоретическое описание используемых нами методов анализа. В пятом разделе описаны проведенные эксперименты и в шестом разделе мы предоставляем полученные результаты. В седьмом разделе представлены выводы. Дополнительные материалы находятся в Приложении.

2 Постановка задачи

Боты способны автоматически выполнять простые процессы намного быстрее человека и тем самым значительно упрощают и ускоряют работу Интернет-серверов. Однако нередко можно встретить использование ботов во вредоносных целях: с помощью них генерируют ненастоящие отзывы, направленные на повышение/понижение рейтинга того или иного продукта, распространяют спам, ложную информацию и т.д.. Большинство работ, направленных на решение проблемы обнаружения ботов, опираются на метаданные и их результаты применяются в конкретным видам ботов из социальных сетей. Намного менее исследованным является задача идентификации ботов через многочисленные текстовые данные, генерируемые ими. С распространением моделей GPT, способных имитировать естественный язык, появляется необходимость исследования самих текстов, их структуру, тонкости. В данной работе мы проводим ряд вычислительных экспериментов и на основе резуль-

татов анализа классифицируем семантические траектории на два класса: человеческих текстов и сгенерированных ботами текстов. Классификация проводится на уровне значимости в 5%, статистическая значимость оценивается тестом Уилкоксона [3]. Преимущество данного подхода в том, что он не требует большого количества размеченных данных и основывается на семантических свойствах текстовых данных. Поскольку тексты могут генерироваться бесконечным множеством людей и ботов и постоянно изменяться, при решении задачи идентификации ботов важно исследовать не конкретные боты, а произвольные.

По этой причине в работе исследуются художественная литература и сгенерированные LSTM и GPT моделями тексты (т.е. присутствуют как простые, так и сложные боты). С помощью метода сингулярного разложения [4] и моделей Word2Vec [5] каждому тексту сопоставляется последовательность из m-мерных векторных представлений слов $x = \{x_i\}_{i=1}^{\ell}$. Здесь и далее будем называть такие последовательности *семантическими траекториями*. В данной работе рассматриваются короткие семантические траектории, а именно подпоследовательности длины n. Векторные представления для них получаются конкатенацией m-мерных представлений входящих в подпоследовательность слов.

Анализ семантических траекторий проводится с использованием следующих двух подходов:

- с помощью кластеризации проводится анализ семантических траекторий и их взаиморасположения. Основной гипотезой является более компактная структура текстов ботов — мы предполагаем, что кластера ботов должны получиться более разделимыми и компактными;
- второй подход использует теорию информации и исследует расположение текстов на плоскости энтропии-сложности [6]. Мы предполагаем, что тексты, написанные людьми, соответствуют хаотическим процес-

сам, а семантически более простые тексты ботов находятся в областях детерминированных или шумовых процессов.

Исследование проводится для языков из разных групп: вьетнамский (вьетская группа), русский (восточнославянская группа) и английский (немецкая группа) — с предположением, что результаты будут отличаться для разных языков.

3 Обзор литературы

Социальная сеть играет незаменимую роль в современном обществе и является одним из основных источников информации. С одной стороны, появление Интернета и социальных сетей привело к доступности знаний практически любому человеку. С другой, огромное количество поступаемой информации не проходит верификацию и человек часто получает недостоверные знания. В худшем случае, информация может быть не просто непроверенной, но и умышленно ложной. Проблема идентификации ботов, используемых во вредоносных целях, существует уже более десятка лет. На данный момент наблюдается "пандемия социальных ботов": по оценкам на 2017-й год среди всех аккаунтов в Twitter около 15% являются ботами, в Facebook на 2019-й год — 11% [7]. Такие боты нередко вредоносны: распространяя ложную информацию, они могут влиять на людей при выборе покупок, избрании политических кандидатов и т.п., поэтому крайне важно уметь их обнаруживать.

Большое количество работ сосредоточено на идентификации именно социальных ботов на таких платформах, как Твиттер или Facebook. Основные методы решения задачи идентификации можно разделить на следующие группы: 1) выделяющие статистические признаки, 2) использующие графовые подходы (обычно проводится анализ взаимодействий разных аккаунтов социальных сетей), 3) комбинации данных подходов.

Большинство работ в основном используют алгоритмы обучения с учителем и сосредоточены на построении признаков, которые затем используются для построения моделей-классификаторов. Существуют различные методы построения таких признаков. Канг и др. [8] используют простые лексические и синтаксические характеристики, такие как частота букв или средняя длина слова. Хейдари и др. [9] определяют тональность английских и голландских твитов, вычисляя их полярность. Кардайоли и др. [10] моделируют пользователя Твиттера с помощью набора стилистических признаков и отличает аккаунты ботов от людей, анализируя постоянство стиля их сообщений. Чакраборти и др. [11] сочетают выделение текстовых признаков с графовыми подходами. В статье [1] авторы представляют архитектуру SentiBot, также сочетающей методы графового и семантического анализа.

На данный момент основные state-of-the-art решения имплементируют методы обучения с учителем (supervised learning), однако такой подход требует большого количества размеченных данных и не адаптируется к изменениям в данных. Поэтому еще одним важным направлением в решении проблемы идентификации ботов является применение методов обучения с частичным привлечением или без учителя (semi-supervised и unsupervised learning). В статье [12] авторы предлагают решать задачу выявления аномалий и показывают, что такой подход позволяет выявлять ботов в Твиттере с точностью в 90%, при этом приспосабливаясь к изменениям в поведении ботов.

Существуют также и решения, использующие концепции из теории информации. Чу и др. [13] характеризуют различия между действиями ботов и людей в Твиттере, вычисляя энтропию статистики активности аккаунтов. Они обнаружили, что у людей энтропия выше, чем у ботов, что подчеркивает их более сложное поведение во времени. В нашей работе мы применяем аналогичные идеи к текстовым данным вместо метаданных.

4 Методология

4.1 Сбор данных

В качестве данных из открытых источников были собраны тексты художественной литературы на вьетнамском, русском и английском языках. Для русского и английского языка большая часть текстов взята из проекта Гутенберга. Детали корпусов приведены в таблице 4.1.

язык	**размер корпуса**	**средний размер текста**
вьетнамский	1071	54532
русский	12692	1000
английский	11008	21000

Таблица 4.1: Характеристики корпусов художественной литературы

4.1.1 Генерация текстов

В качестве корпусов бота рассматриваются тексты, сгенерированные с помощью двух моделей — Long short-term memory recurrent neural network (LSTM) и Generative Pretrained Transformers (GPT). Модель LSTM генерирует тексты посимвольно [1]. Модель обучается на части корпуса художественной литературы (10000 эпох, размер батча — 16, последовательности длиной в 256 знаков). В качестве GPT моделей были взяты предобученные модели с huggingface[2].

Тексты генерируются следующим образом: из художественного произведения последовательно для каждого 500-го (условное количество слов одной страницы книги) слова генерируется "страница" текста с этим словом вначале.

[1] На основе https://github.com/alexyar88/char-lstm-text-generation

[2] Модели GPT: вьетнамский; русский; английский

4.2 Предобработка данных

Прежде чем приступить к анализу данных, тексты необходимо предобработать, а именно токенизировать (выделить из текста отдельные токены) и лемматизировать (привести слова-токены к начальным формам). Эти процедуры отличаются для разных языков. Например, для английского и русского языков токенизацию можно провести нахождением символов пробела в тексте, тогда как во вьетнамском такое выполнение некорректно. Слова во вьетнамском языке в основном являются составными и состоят из нескольких слов, разделённых между собой пробелами. Например, слова *môi* и *trường* по-отдельности обозначают “губы” и “школа”, тогда как слово *môi trường* в переводе — “окружающая среда”. Разбиение предложения на отдельные слова должно производиться в зависимости от контекста, поэтому разделение слов по символам пробела, как это делается для английского и русского языков, некорректно. С другой стороны, во вьетнамском языке не требуется проводить лемматизацию, поскольку все слова имеют только одну форму, в отличии от английского и русского языков.

Кроме токенизации и лемматизации также выделяются и заменяются на специальные токены местоимения, предлоги, имена числительные и имена собственные. Для этого проводится извлечение именованных сущностей (Named entity recognition - NER) и разметка частей речи (Part of Speech (POS) tagging) .

Для обработки текстов на разных языках используются следующие библиотеки:

- вьетнамский язык — pyvi [3]
- русский язык — natasha [4]

[3] https://github.com/trungtv/pyvi
[4] https://github.com/natasha/natasha

- английский язык — spaCy (с en_core_web_lg) [5]

4.3 Векторизация данных

4.3.1 Сингулярное разложение

Существует большое количество способов преобразования текстовых данных в числовые векторы. В работе [4] описывается решение с помощью сингулярного разложения матрицы W (документы $\times$ термы). В данной работе используется TF-IDF матрица, построенная для набора текстов $D = (d_1, \ldots, d_N)$ и набора встречаемых слов $T = (t_1, \ldots, t_M)$. Элементы матрицы определяются как $w_{ij} = TF(t_i, d_j) \cdot IDF(t_i, D)$,

$$TF(t, d) = \frac{n_t}{\sum\limits_{k \in d} n_k}, \quad IDF(t, D) = \frac{|D|}{|\{d \in D : t \in d\}|} \tag{1}$$

где n_t — число вхождений слова t в текст.

С помощью m-рангового сингулярного разложения матрицу можно представить как

$$W \simeq W' = U \Lambda V^T \tag{2}$$

.

Здесь матрицы U и V — ортонормальные матрицы размеров $M \times m$, $N \times m$ соответственно, а Λ — диагональная матрица с сингулярными значениями $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_m > 0$. Матрица W' является наилучшим приближением матрицы W по L_2-норме, сохраняет отражённые в W семантические связи и позволяет получить векторные представления для встречаемых слов:

$$t_i \mapsto u_i \Lambda, \quad u_i \text{ — } i\text{-ая строка } U \tag{3}$$

[5] https://spacy.io/

При этом, если получены вектора представлений слов для $m = m_1$, то для получения векторов размера $m = m_2 < m_1$ достаточно взять первые m_2 компонент из m_1-мерных векторов. Это позволяет существенно сократить вычислительные ресурсы и является преимуществом для данной работы, поскольку анализ проводится для разных значений m.

В работе для получения сингулярного разложения используется модуль *linalg* библиотеки *scipy*.

4.3.2 Word2Vec

Второй метод построения векторов слов, используемый в данной работе — Word2Vec-модели [5]. Векторные представления извлекаются из вероятностного распределения, получаемого в результате обучения данных нейронных сетей. С использованием библиотеки *gensim*[6] мы обучаем модели на собранных корпусах художественной литературы. Рассматриваются оба варианта Word2Vec — skip-gram (модель предсказывает слово по соседним словам) и Continuous Bag of Words (CBOW, предсказание по слову соседних ему слов).

Word2Vec-модели хорошо выделяют структурные отношения между словами, близко располагая векторные представления синонимичных слов, из-за чего этот метод был выбран для анализа текстов.

4.4 Кластеризация

Для кластеризации тексты разбиваются на n-граммы. Векторные представления для n-грамм получаются конкатенацией векторов входящих в него слов.

В данной работе анализ проводится с помощью двух алгоритмов кла-

[6] https://radimrehurek.com/gensim/models/word2vec.html

стеризации — Уишарта и K-Means. Поскольку мы заведомо не знаем особенностей структуры семантического пространства, важно исследовать его с помощью методов, подходящих к разным видам данных. K-Means хорошо работает с данными, в которых кластера можно представить гиперсферами. В свою очередь, алгоритм Уишарта является плотностным и хорошо выделяет кластера произвольной формы, при этом находя шумовые объекты.

4.4.1 Алгоритм Уишарта

Алгоритм Уишарта [14] является плотностным алгоритмом и применяет концепции из теории графов.[7]

Кластеризируем объекты $x_1, \ldots, x_n$. Для отдельного объекта x посчитаем её значимость (saliency) как $p(x) = \frac{k}{V_k(x)n}$, где $V_k(x)$ — объём гиперсферы радиуса $d_k(x)$ до k-ближайшего соседа. Алгоритм начинает присваивать объектам метки кластеров из наиболее плотных областей, выделяя шумовые объекты и оценивая значимость кластеров (кластер c будем считать значимым, если для него выполняется $\max\limits_{x_i, x_j \in c} \{|p(x_i) - p(x_j)|\} \geqslant h$).

Кластеризация включает в себя следующие шаги:

1 Найдём для каждого объекта радиус $d_k(x)$ до k-ближайшего соседа. Отсортируем объекты по возрастанию посчитанных расстояний.

2 Инициализируем $G(V, E) = G(\{x_1\}, E_1)$. Через E_i будем обозначать множество рёбер, соединяющих i-ый объект с его k ближайшими соседями: $E_i = \{(x_i, x_j) : d(x_i, x_j) \leqslant d_k(x_i)\}$

3 Для i от 1 до n рассмотрим граф $G(V, E)$.

[7] В работе используется имплементация из https://github.com/Radi4/BotDetection/blob/master/Wishart.py.

- если вершина x_i в графе является изолированной, то создадим новый кластер с объектом x_i
- если x_i соединена только с вершинами из кластера ℓ и при этом кластер является завершённым, то отнесём объект к шуму. Если же кластер не является завершённым, то отнесём объект к этом кластеру
- если объект соединён с вершинами из кластеров $l_1, ..., l_q,$, то:
  - если все кластера являются завершёнными, то объект x_i отнесём к шумовым
  - иначе, если среди кластеров есть 2 и более значимых клатеров (относительно порога h) или если l_1 — шумовой кластер, то x_i и все точки из незначимых кластеров отнесём к шуму
  - иначе объединим кластера $l_1, l_2, \ldots, l_q$ в один кластер l_1 и отнесём к нему объект x_i.
- Обновим граф: $V \leftarrow V \cup \{x_i\}$; $E \leftarrow E \cup E_i$.

4.4.2 Алгоритм K-Means

Алгоритм K-Means разбивает объекты на K кластеров, минимизируя внутрикластерные расстояния до центроидов.

Кластеризация производится следующим образом:

1 Случайным образом генерируются K центроидов $c_1, \ldots, c_K$.

2 Каждый объект относится к кластеру с наиближайшим центроидом:

$$C_i = \{x : ||x - c_i||^2 \leqslant \min_{j=1,...,K} ||x - c_j||^2\} \tag{4}$$

3 Для посчитанных кластеров пересчитываются центроиды:

$$c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{5}$$

Шаги 2-3 повторяются до тех пор, пока разбиение не перестанет меняться.

В работе используется имплементация алгоритма из библиотеки *sklearn*.

4.5 Нечёткая логика

Как правило, рассматриваемые в различных реальных задачах данные являются неточными. Нечёткая логика предоставляет способ моделирования данных, учитывающий их неточности. Эта концепция применима и к текстовым данным: с помощью нечётких множеств удобно моделировать такие качественные понятия как “близкий”, “хороший” и т.д..

Нечёткость также можно ввести и в сами алгоритмы анализа данных. Далее рассмотрим два подхода: нечёткие числа и нечёткие алгоритмы кластеризации.

4.5.1 Нечёткие числа

Для моделирования данных введём понятие нечёткого LR-числа [16]. Для m-мерного объекта $x = (x_1, \ldots, x_m)$ рассмотрим функцию принадлежности μ:

$$\mu_j(x_j) = \begin{cases} L\left(\frac{m_{1j} - x_j}{l_j}\right), & x_j \leqslant m_{1j} \\ 1, & m_{1j} \leqslant x_j \leqslant m_{2j}\,, \\ R\left(\frac{x_j - m_{2j}}{r_j}\right), & x_j \geqslant m_{2j} \end{cases} \quad j = 1, \ldots m \tag{6}$$

Здесь m_1 и m_2 — левый и правый центры соответственно, L, R — функции левого и правого склонов графика функции, а l, r соответственные им длины (см. рис. 4.1).

Наиболее используемая функция принадлежности — симметрическая трапезоидная, т.е. имеющая склоны $L(z) = R(z) = (1 - z)I\{z \in [0, 1]\}$, её же мы используем и в данной работе.

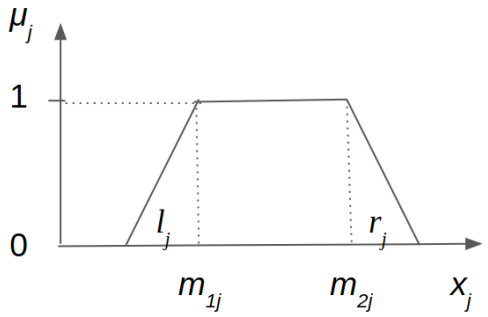

Рис. 4.1: Трапезоидная функция принадлежности

Для текстов нечёткие числа строятся следующим образом. Для каждого m-мерного вектора x определим значение функции принадлежности как

$$\mu_j(x_j) = \frac{n_j}{\max\limits_j n_j}, \quad j = 1, \ldots, m \tag{7}$$

, где n_j — число вхождений j-ой компоненты x в текст. Примем, что посчитанные $\mu_j(x_j)$ попадают на левый склон графика функции. Тогда по фиксированным значениям параметров l_j, r_j, а также $\Delta c = m_{2j} - m_{1j}$ — расстояние между центрами m_{1j}, m_{2j}, можно найти координаты центров и таким образом построить нечёткое число.

Поскольку в кластеризации мы рассматриваем n-граммы, введём следующую модификацию: для всех слов, встречаемых в n-грамме, получим вышеописанным способом нечёткие числа и возьмём их нечёткое пересечение этих чисел, а именно минимум соответствующих функций принадлежности [16]. Например, для биграмма $(x, y), x, y \in \mathbb{R}^m$ функция принадлежности будет

определена следующим образом:

$$\mu((x,y)) = \{\min(\mu_j(x), \mu_j(y))\}_{j=1}^{m} \tag{8}$$

Код для фаззификации данных и вычисления нечёткого расстояния представлен в репозитории работы.

4.5.2 Нечёткая кластеризация C-Means

C-Means является нечётким аналогом алгоритма K-means и минимизирует расстояния между объектами и центроидами кластеров [17]. Алгоритм принимает на вход параметр K - количество кластеров. При инициализации каждому объекту x случайным образом присваивается коэффициент принадлежности кластерам $w_k(x)$ $(k = 1, \dots K)$. Одна итерация алгоритма включает следующие шаги:

1 Пересчёт центроидов (взвешенная средняя объектов кластера):

$$c_k = \frac{\sum\limits_x w_k^m(x)x}{\sum\limits_x w_k^m(x)} \tag{9}$$

2 Пересчёт коэффициентов:

$$w_k(x) = \left(\sum_{c \in C} \left(\frac{||x - c_k||}{||x - c||} \right)^{\frac{2}{m-1}} \right)^{-1} \tag{10}$$

, где $C = \{c_1, c_2, ..., c_K\}$ — множество центроидов кластеров.

После каждой итерации $i+1$ проверяется условие: $||w^{(i+1)} - w^{(i)}||^2 < \varepsilon$ или $||C^{(i+1)} - C^{(i)}||^2 < \varepsilon$, если оно выполняется, то алгоритм завершает работу.

В работе используется имплементация алгоритма из библиотеки *fuzzy-c-means*.

4.6 Построение плоскости энтропии-сложности

Другим подходом к изучению семантического пространства текстов является рассмотрение текста как временного ряда. В работе Мартино, Пластино, Россе [6] описан метод, позволяющий разделить хаотические ряды от простых детерминированных или стохастических процессов. В методе для ряда вычисляются две характеристики — сложность и энтропия, и расположение полученной точки на плоскости энтропии-сложности определяет тип ряда.

В качестве энтропии берётся энтропия Шеннона $S[P]$. Величина нормируется на максимальное значение $S_{\max} = S[P_e] = \ln N$ ($P_e = \{1/N, ..., 1/N\}$ — равномерное распределение на N значениях).

$$\begin{aligned} S[P] &= -\sum_{j=1}^{N} p_j \ln p_j \\ H_S[P] &= S[P]/S_{\max} \end{aligned} \tag{11}$$

Сложность вычисляется следующим образом:

$$C_{JS}[P] = Q_J[P, P_e] H_s[P] \tag{12}$$

Здесь, Q_J — дивергенция Дженсона-Шеннона между распредлениями P, P_e, где Q_0 — нормировочная константа ($0 \leqslant Q_J \leqslant 1$):

$$Q_J[P, P_e] = Q_0\{S[(P + P_e)/2] - S[P]/2 - S[P_e]/2\} \tag{13}$$

Рассмотрим одномерный временной ряд $\{x_t\}_{t=1}^{L}$ длины L. Для фикси-

рованного размера окна n и каждого момента времени t выделим n-грамм $(t = n, n+1, \ldots, L)$:

$$(t) \mapsto (x_{t-(n-1)}, x_{t-(n-2)}, \ldots, x_{t-1}, x_t) \tag{14}$$

В качестве распределения P рассмотрим распределение так называемых порядковых паттернов (ordinal patterns). Определим порядковый паттерн для рассматриваемого n-грамма $\pi = (r_0, r_1, \ldots, r_{n-1})$ — такую перестановку (0,1,. . . , $n-1$), что выполняется $x_{t-r_{n-1}} \leqslant x_{t-r_{n-2}} \leqslant \cdots \leqslant x_{t-r_1} \leqslant x_{t-r_0}$. Всего существует $n!$ таких перестановок. Распределение перестановок тогда определяется как частота их встречаемости:

$$p(\pi) = \frac{\{t | t \leqslant L - n + 1 : (t) \text{ соответствует перестановка } \pi\}}{L - n + 1} \tag{15}$$

Поскольку в нашей работе также исследуются векторные представления разных размерностей m, представим модификацию вышеописанного метода для многомерного случая.

Рассмотрим m-мерный временной ряд $\{x^t\}_{t=1}^{L}$, $x_t \in \mathbb{R}^m$. Для каждого момента времени t и каждой компоненты x_d^t $(d = 1, \ldots, m)$ рассмотрим n-граммы:

$$\begin{aligned}
(t_1) &\mapsto (x_1^{t-(n-1)}, x_1^{t-(n-2)}, \ldots, x_1^{t-1}, x_1^t) \\
&\vdots \\
(t_d) &\mapsto (x_d^{t-(n-1)}, x_d^{t-(n-2)}, \ldots, x_d^{t-1}, x_d^t) \\
&\vdots \\
(t_m) &\mapsto (x_m^{t-(n-1)}, x_m^{t-(n-2)}, \ldots, x_m^{t-1}, x_m^t)
\end{aligned} \tag{16}$$

Для каждого t_d получим перестановку π_d как в одномерном случае. Тогда определим общую перестановку как $\Pi = (\pi_1, \pi_2, \ldots, \pi_m)$ и будем рассматри-

вать распределение всевозможных перестановок Π (всего $(n!)^m$ вариантов).

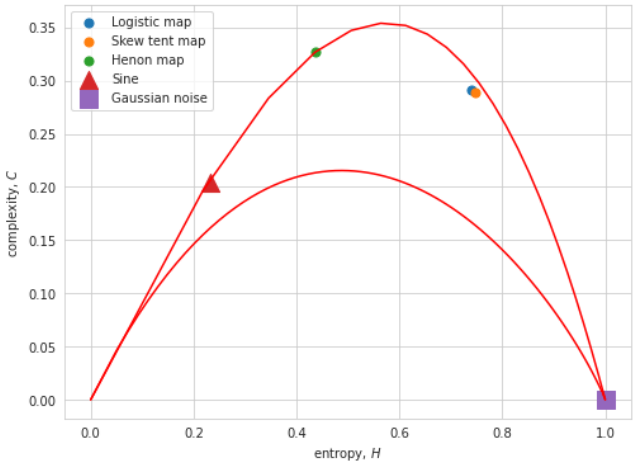

Рис. 4.2: Плоскость энтропии-сложности при $n = 4$. Треугольниками отмечены простые детерминированные процессы, точками — хаотические процессы, квадратами — простые стохастические процессы.

Таким образом, каждому тексту, как временному ряду, можно сопоставить точку на плоскости энтропии-сложности. При этом по расположению точки относительно теоретических границ [18] можно определить тип ряда. Простые стохастические процессы (например, гауссовский шум) попадают в правый нижний угол, простые детерминированные — в нижний левый угол (синусоида), точки, прилежащие к середине верхней границы, являются хаотическими [6]. На рис. 4.2 можно увидеть пример расположения различных временных рядов.

5 Эксперименты

четкая логика	**нечеткая логика**
алгоритм Уишарта	алгоритм Уишарта на нечетких числах
K-Means	Fuzzy C-Means

Таблица 5.1: Методы кластеризации

Для текстов из вьетнамской, русской и английской корпусов применяются вышеописанные методы кластеризации и построения плоскости энтропии-

сложности. Кластеризация применяется в четырех вариациях: по два алгоритма (плотностная кластеризация и K-Means) на каждый вариант представления данных (т.е. с применением четкой и нечеткой логики, см. таблицу 5.1).

Все эксперименты проведены для широкого набора значений параметров m и n, в работе представлены результаты для значений m и n в пределах от 1 до 16 и от 1 до 20 соответственно. На основе полученных результатов далее строятся простые классификаторы на основе метода опорных векторов для обнаружения ботов.

6 Результаты

Кластеризация

Прежде чем использовать кластеризацию для построения отдельных признаков для каждых текстов, были проведены эксперименты с корпусом, взятым полностью. Для каждого типа текста было взято по 3 миллионна уникальных n-грамм. Кластеризация всего корпуса как алгоритмом Уишарта, так и методом K-Means показал, что кластера ботов получаются более компактными: на рис. 6.1 (для кластеризации алгоритмом Уишарта), 6.2 (K-Means) видно, что для GPT и LSTM-текстов значение метрики RMSSTD[8], оценивающей компактность получаемых кластеров [19], получается значительно ниже, чем у литературных текстов. При этом значение RS, оценивающей разделимость кластеров между собой [19], для ботов ниже.

Кластеризация на отдельных текстах также показывает, что кластера ботов получаются более компактными. На рисунках 6.3-6.5 для вьетнамского, русского и английского языков соответственно изображены распределения метрики RMSSTD для кластеризации алгоритмами Уишарта и K-Means. За-

[8]Формулы метрик приведены в Приложении

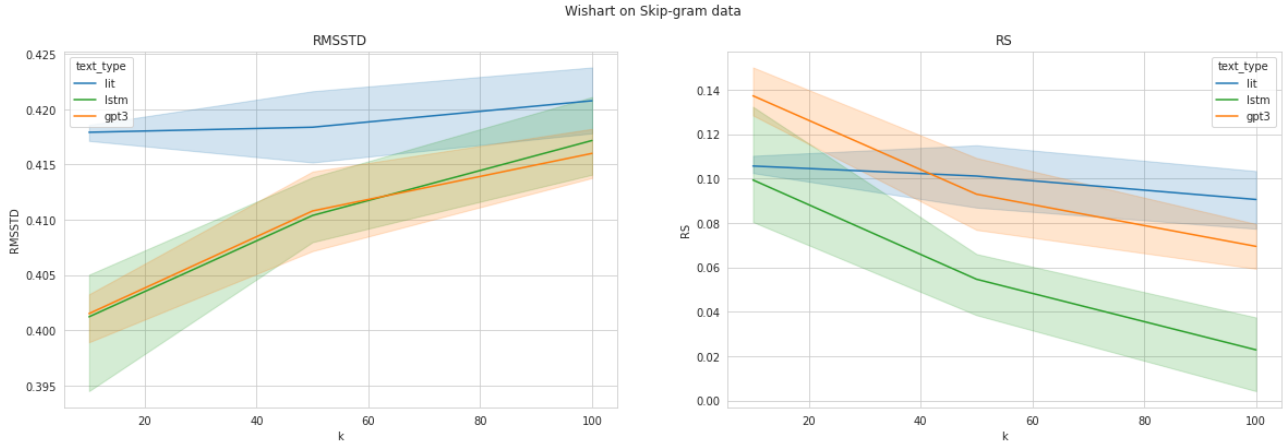

Рис. 6.1: Метрики разделимости кластеров (cohesion и separation) на всем корпусе вьетнамских текстов для алгоритма Уишарта на Skip-gram векторах

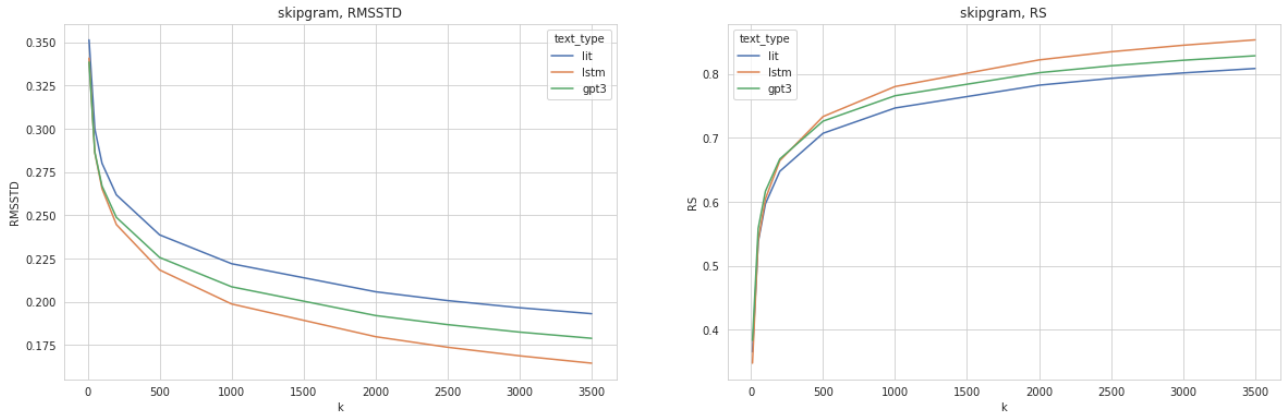

Рис. 6.2: Метрики разделимости кластеров (cohesion и separation) на всем корпусе вьетнамских текстов для кластеризации K-Means на Skip-gram векторах

метим, что для всех языков наименее компактными получаются кластера литературных текстов. Отдельно отметим, что для вьетнамского языка (рис. 6.3) для биграмм Skip-gram-векторов, тексты LSTM, получаются сильно компактнее, тогда как GPT-тексты находятся "ближе" к литературным. Тестом Уилкоксона [3] было выявлено статистически значимое различие распределение RMSSTD для литературных и текстов GPT: значение p-value получается равным 1.54e-2 (в среднем для разных наборов значений m, n), различие человеческих текстов от LSTM-текстов еще больше: p-value для них равно 0.0029. Для русского и английского языков распределения также статистически значимы (принятый нами уровень значимости — 5%): соответствующие значения p-value — 5.92e-3 и 2.29e-2.

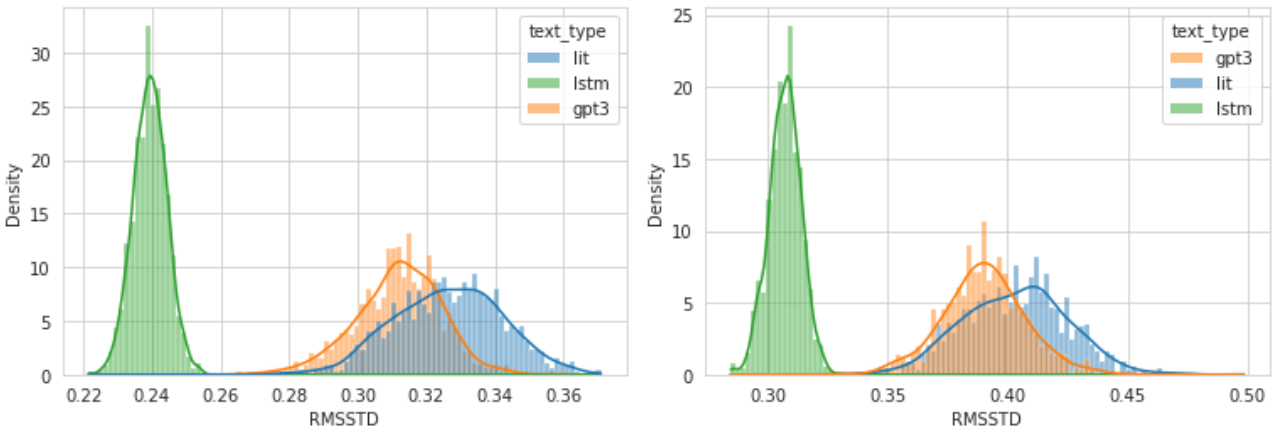

Рис. 6.3: Распределение значений RMSSTD для текстов вьетнамского языка; $m = 8, n = 2$, Skip-gram-вектора. Слева: для K-Means кластеризации, справа: для алгоритма Уишарта

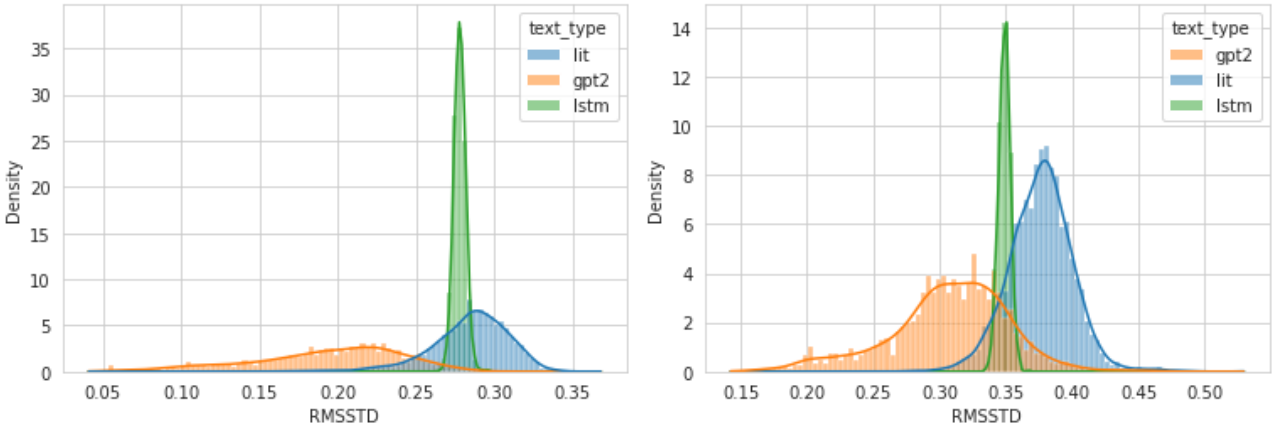

Рис. 6.4: Распределение значений RMSSTD для текстов русского языка; $m = 8, n = 2$, Skip-gram-вектора. Слева: для K-Means кластеризации, справа: для алгоритма Уишарта

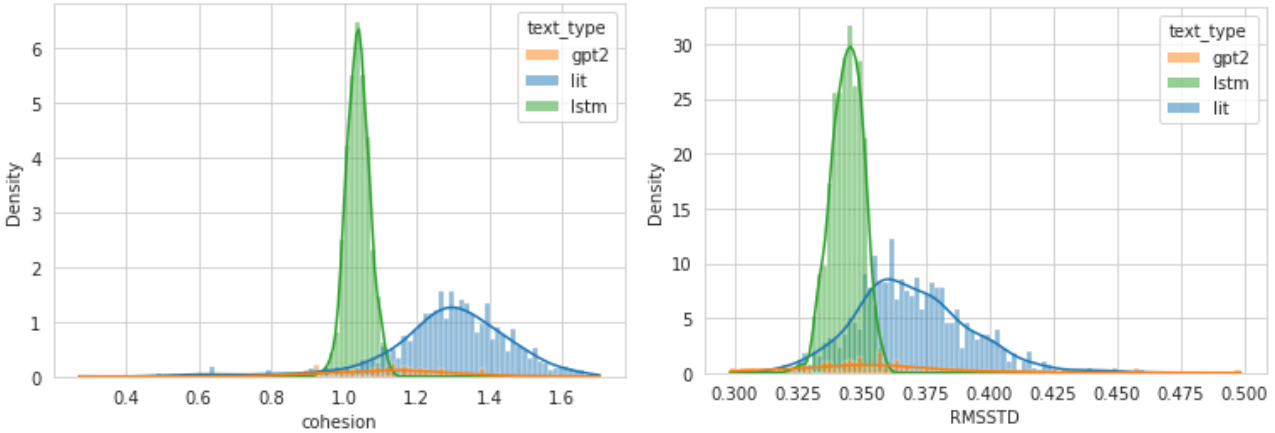

Рис. 6.5: Распределение значений RMSSTD для текстов английского языка; $m = 8, n = 2$, Skip-gram-вектора. Слева: для K-Means кластеризации, справа: для алгоритма Уишарта

Кроме компактности кластеров алгоритмом Уишарта можно выявить, насколько много выбросов присутствует в тексте. На графиках 6.6 (алгоритм Уишарта), 6.7 (алгоритм Уишарта на нечетких числах) для рассматриваемых языков отмечена доля шума в завимости от m. Заметим, что для всех языков меньше всего выделяется шум для представлений CBOW (кластеризация получается практически без шума при $m > 3$). При кластеризации текстов LSTM на всех языках выделяется больше всего шума, в свою очередь, тексты GPT по доле шума близки к литературным.

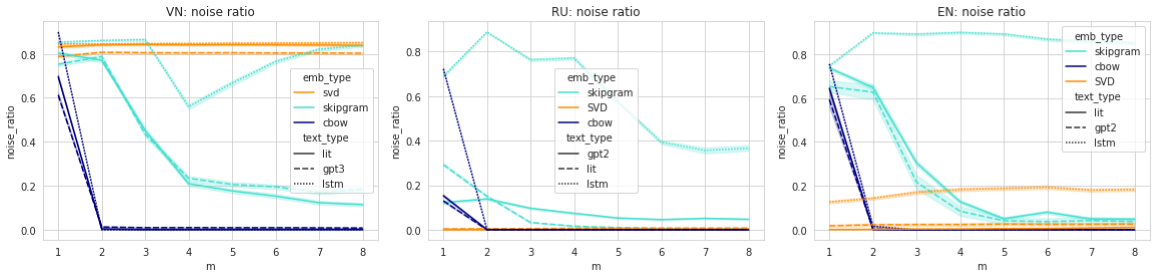

Рис. 6.6: Доля шума, выделенного алгоритмом Уишарта для вьетнамского (слева), русского (посередине) и английского (справа) языков.

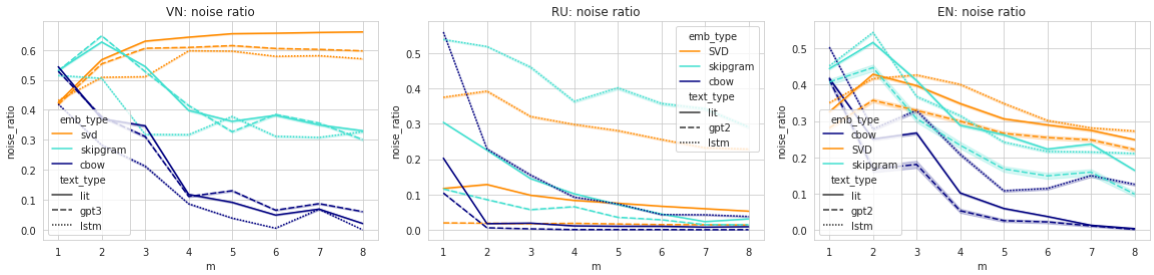

Рис. 6.7: Доля шума, выделенного алгоритмом Уишарта на нечётких числах для вьетнамского (слева), русского (посередине) и английского (справа) языков.

Таким образом мы подтверждаем гипотезу о том, что тексты ботов получаются более простыми (кластера более компактные) и достигают разнообразия за счёт генерации шума. Далее мы используем данный результат для построения классификаторов.

Классификация

Используя полученные результаты по кластеризации всего корпуса, мы переходим к применению кластеризации для построения характеристических признаков. На каждом тексте корпусов была запущена кластеризация с значениями параметров $m = 1 \dots 8$, $n = 2 \dots 10$. (Отдельно про подбор параметра для построения нечетких чисел см. в Приложении).

Поскольку по метрикам было выявлено, что кластера ботов получаются более компактными, в качестве признаков для построения классификаторов были выбраны внутрикластерные расстояния (усредненные, минимальные и максимальные). В качестве классификатора применяется метод опорных векторов с линейным ядром, параметр для L_2-регуляризации подбирается на обучающей выборке с кросс-валидацией. Кроме классификации всех типов текстов также отдельно были рассмотрены классификаторы для разных ботов по-отдельности. Для разных m, n строились отдельные классификаторы. Точность наилучших моделей приведены в таблицах 6.1- 6.3 для вьетнамского, русского и английского языков соответственно. Дополнительно также были посчитаны взвешенные значения F-меры, они получились близки к точности (выборки сбалансированны — для каждого типа текста бралось по 1000 примеров). Полные таблицы по всем парам значений m и n приведены в репозитории.

	все боты		**LSTM**		**GPT**	
Вид кластеризации	Train	Test	Train	Test	Train	Test
K-Means	0.862	0.903	1.0	1.0	0.887	0.881
Уишарт	0.902	0.896	1.0	1.0	0.893	0.900
C-Means	0.887	0.893	1.0	1.0	0.871	0.871
Уишарт на нечетких числах	**0.929**	**0.942**	1.0	1.0	**0.893**	**0.926**

Таблица 6.1: Точность классификаторов для вьетнамского языка

	все боты		**LSTM**		**GPT**	
Вид кластеризации	Train	Test	Train	Test	Train	Test
K-Means	0.912	0.934	0.999	1.0	0.871	0.916
Уишарт	**0.937**	**0.954**	0.999	1.0	**0.913**	**0.944**
C-Means	0.882	0.894	0.999	1.0	0.838	0.857
Уишарт на нечетких числах	0.882	0.913	0.991	1.0	0.904	0.911

Таблица 6.2: Точность классификаторов для русского языка

	все боты		**LSTM**		**GPT**	
Вид кластеризации	Train	Test	Train	Test	Train	Test
K-Means	0.947	0.975	1.0	1.0	0.903	0.881
Уишарт	**0.953**	**0.975**	1.0	1.0	0.904	0.881
C-Means	0.943	0.970	0.999	1.0	0.897	0.921
Уишарт на нечетких числах	0.945	0.947	1.0	1.0	**0.907**	**0.94**

Таблица 6.3: Точность классификаторов для английского языка

Заметим, что боты идентифицируются лучше всего по признакам от алгоритма Уишарта. Мы предполагаем, что алгоритм K-Means (и его нечеткая вариация C-Means) действуют хуже, поскольку данные, скорее всего, зашумленные, а кластера имеют произвольную форму (K-Means на таких данных работает не так хорошо, как плотностные алгоритмы кластеризации). При этом для вьетнамского и английского языка наблюдается улучшение качества от применения метода фаззификации данных: для вьетнамского и на всех ботах, и на GPT-боте отдельно точность классификации повышается, для английского точность для алгоритма Уишарта на нечетких числах выше всего при классификации GPT-текстов отдельно.

Таким образом, методом кластеризации было выявлена более простая структура ботов — их кластера получаются компактнее. При этом признаки, выделяемые на основе кластеризации, результируют в точные классификаторы. Дополнительным улучшением точности моделей является применение нечеткой логики — для вьетнамского и английского языка наблюдается улучшение качества классификации при кластеризации на нечетких числах.

Энтропия и сложность

Для вычисления метрик энтропии и сложности, а также теоретических границ была имплементирована библиотека *ordec*[9]. Код включает в себя реализацию вышеописанной модификации метода энтропии-сложности для многомерного случая, чего нет в других библиотеках. Дополнительно также были проведены эксперименты с другим вариантом применения метода в многомерном случае (а именно, применение частичного упорядочивания многомерных объектов для составления порядковых паттернов), однако такой способ оказался неподходящим для решения задачи работы, подробнее можно увидеть в Приложении А.

При определенных значениях параметров m и n тексты могут попадать в зоны шума или детерминированности, где сложно разделить разные типы текстов (пример можно увидеть на рис. 6.11, для SVD-векторов при $m = 1, n = 4$ все тексты на русском языке смешиваются). По этой причине особое внимание надо уделять таким значениям параметров m, n, при которых тексты попадают в хаотическую область. На рисунках 6.8 - 6.10 зеленым цветом выделена область значений параметров m, n, для которых литературные тексты попадают в область хаоса. При значениях ниже оранжевой границы тексты попадают в область шума, а выше синей границы — в область детерминированных процессов. Значения существенно отличаются для разных языков: для вьетнамского в рассматриваемую область входят более длинные подпоследовательности, при $m = 1$ значения n от 10 до 14, тогда как для русского — от 6 до 8, а для английского от 7 до 8. Отдельно выделяются SVD-представления для английских литературных текстов: на всех значениях m, n тексты попадают в область шума. Примеры плоскости энтропии-сложности для таких параметров представлены на рис. 6.13, где видно, что, например, для $m = 3, n = 4$ тексты попадают хаотическую область и хорошо

[9] https://github.com/quynhu-d/Spot-the-Bot/tree/main/ordec

разделимы.

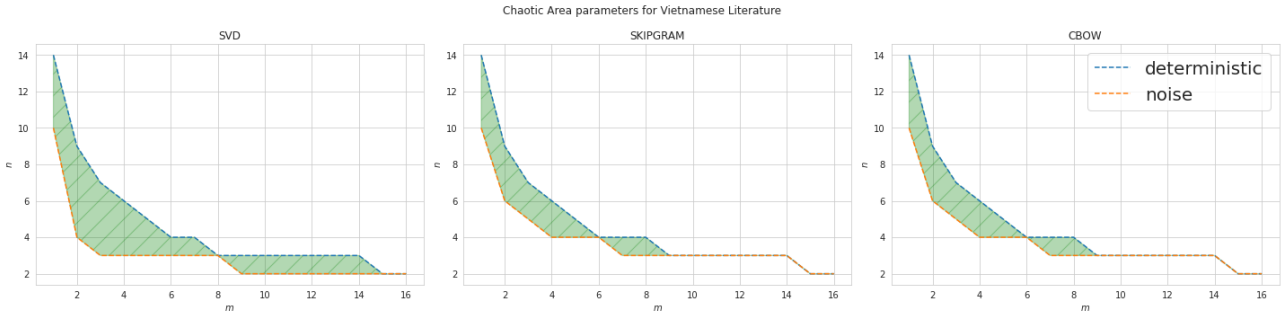

Рис. 6.8: Значения m, n, при которых тексты вьетнамской литературы попадают в область хаотических процессов.

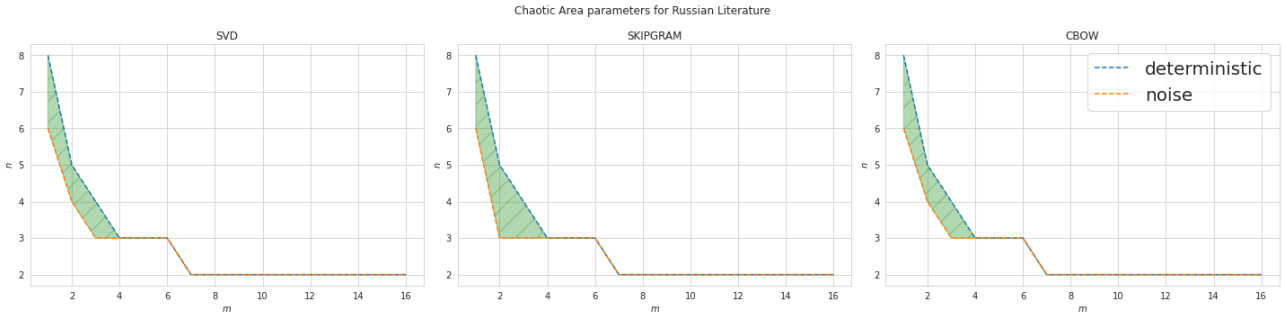

Рис. 6.9: Значения m, n, при которых тексты русской литературы попадают в область хаотических процессов.

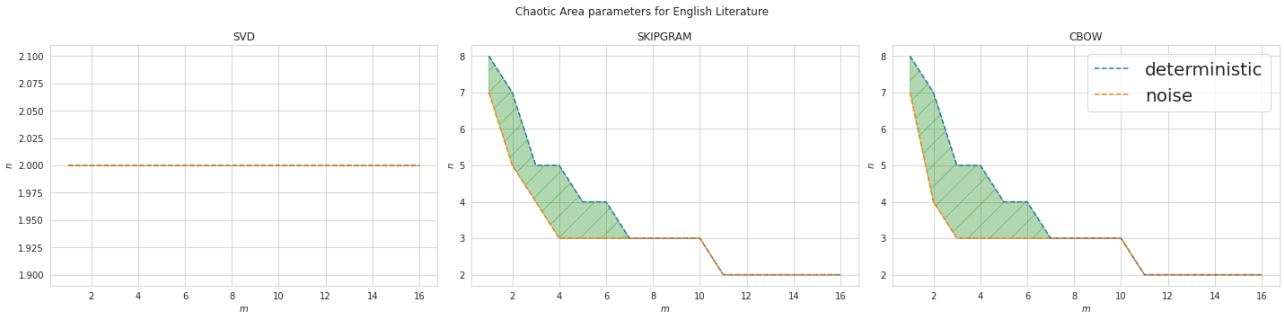

Рис. 6.10: Значения m, n, при которых тексты английской литературы попадают в область хаотических процессов.

На рис. 6.14 показана зависимость энтропии и сложности от значений m и n для SVD-векторов текстов вьетнамского языка. В среднем наиболее сложными получаются тексты художественной литературы. Аналогичные графики для SVD-векторов для русского и английского языка представлены на рис. 6.15, 6.16. Заметим, что для всех языков при $n = 2$ более "сложными"становятся тексты LSTM-ботов. Мы предполагаем, что это получается

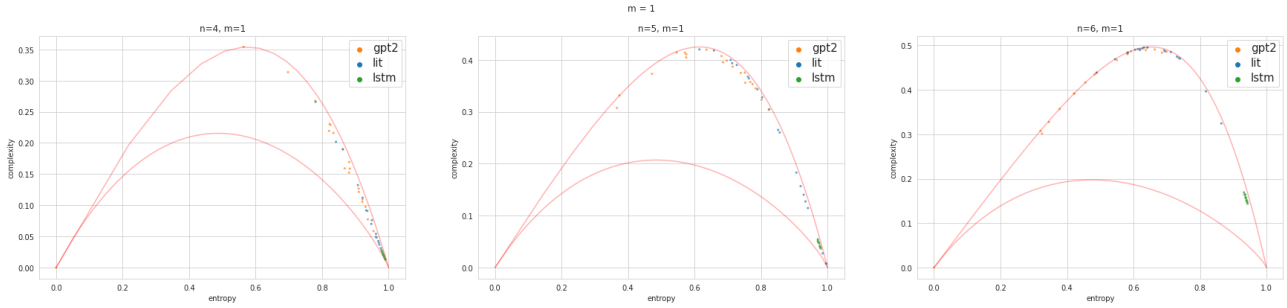

Рис. 6.11: Плоскости энтропии-сложности для русских текстов (с SVD-векторами). $m = 1, n = 4, \ldots 6$

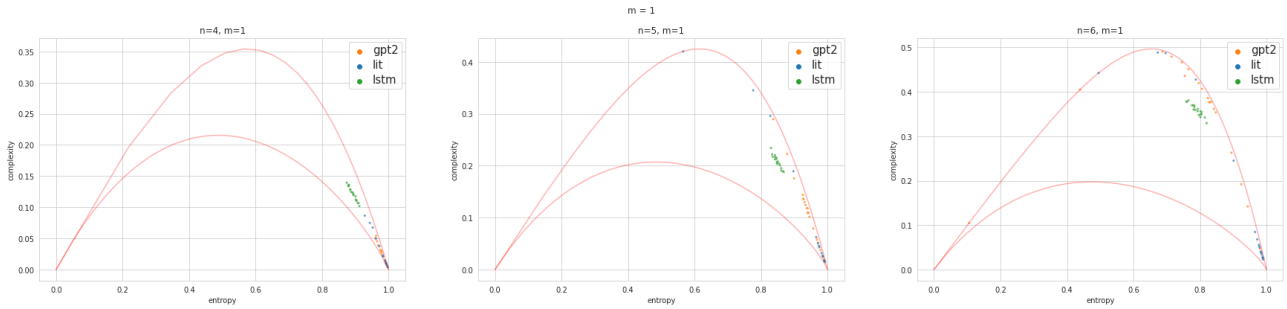

Рис. 6.12: Плоскости энтропии-сложности для английских текстов (с SVD-векторами). $m = 1, n = 4, \ldots 6$

из-за разнообразия биграмм: в текстах, сгенерированных моделями GPT, и художественной литературе может встречаться не так много пар словосочетаний, как в текстах, сгенерированных LSTM (модель крайне проста и не учитывает взаимосвязи между словами, из-за чего может сочетать любые слова). Похожая динамика получается и на Word2Vec представлениях, графики можно найти в Приложении (рис. A.4 - A.9).

Таким образом, для определенных значений m и n выявлено, что метрики энтропии и сложности значительно отличаются для разных типов текстов. Основываясь на данном результате, построим классификаторы, использующие метрики энтропии и сложности в качестве признаков. В качестве модели классификации применим метод опорных векторов с линейным ядром, параметр для L_2-регуляризации подбирается кросс-валидацией на обучающей выборке.

Первоначальным классификатором была взята модель, обученная на значениях энтропии и сложности для всех параметров m, n одновременно (т.е.

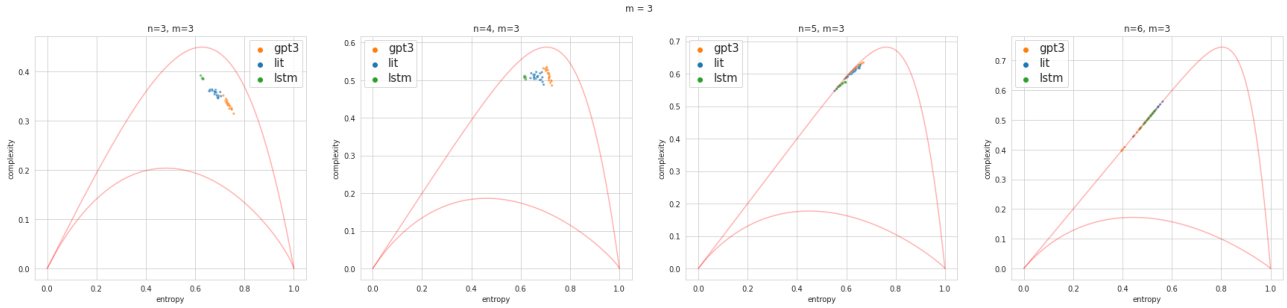

Рис. 6.13: Плоскости энтропии-сложности для вьетнамских текстов (с SVD-векторами). $m = 3, n = 3, \ldots 6$

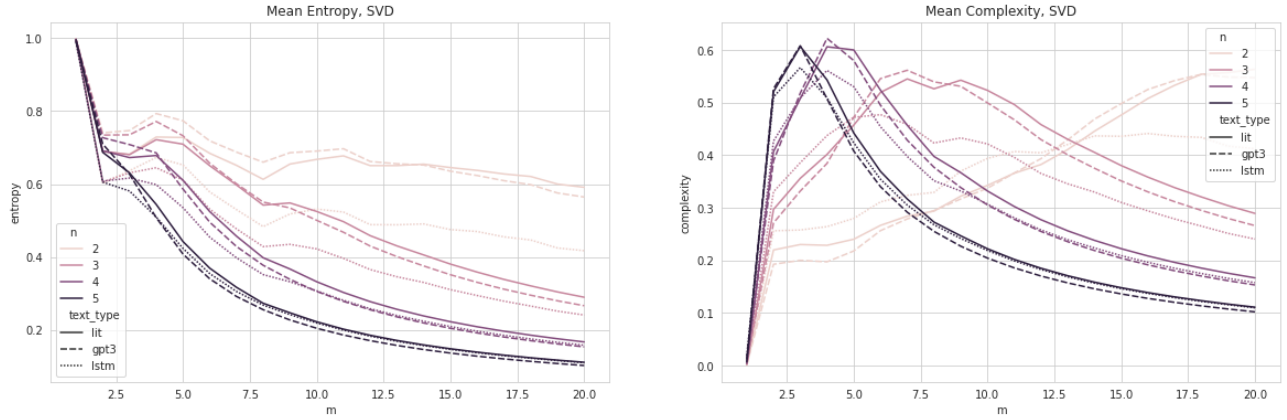

Рис. 6.14: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \ldots 5$ на SVD-векторах для вьетнамских текстов.

помимо метрик в выборке дополнительно присутствуют числовые признаки m и n). Точность такой модели получилась невысокой — всего 0.57 на обучающей и тестовой выборках. Значительно лучше себя показали классификаторы, обученные на значениях энтропии и сложности для разных наборов значений параметров m, n по-отдельности. Кроме классификации всех трех типов текстов (литературных, сгенерированных моделями GPT и сгенерированных моделями LSTM) также орассмотрены классификаторы для разных ботов по-отдельности. В таблицах 6.4 (точность), 6.5 (взвешенная F-мера) приведены наилучшие результаты классификаторов. Заметим, что метрики энтропии и сложности позволяют отличить простых ботов (LSTM) практически с 100-%-ной точностью и довольно точно для сложных ботов (GPT) — 90% на русском и 99% на вьетнамском и английском языках. Для вьетнамского лучший классификатор на всех ботах получился для SVD-представлений при $m = 3, n = 3$, для русского — Skip-gram, $m = 1, n = 8$, английского —

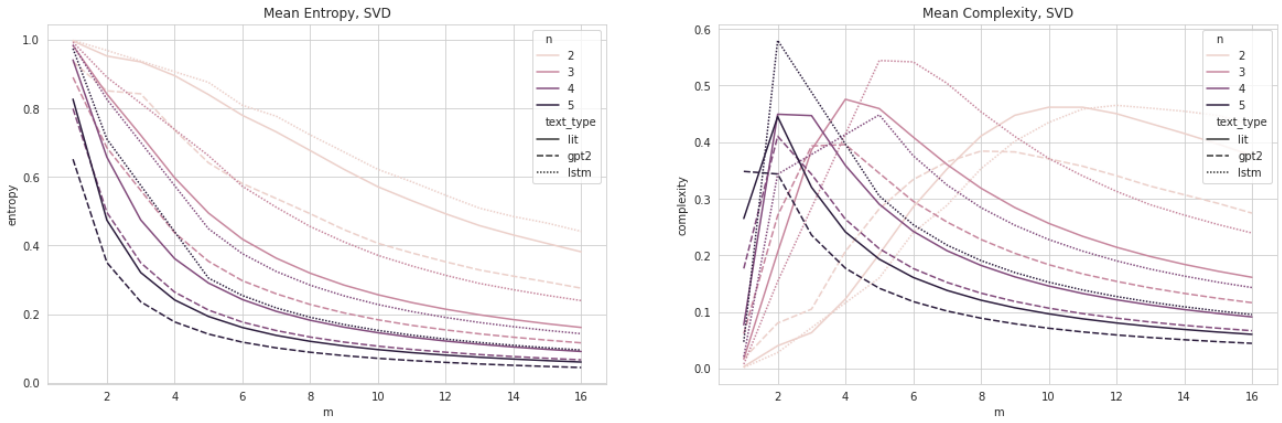

Рис. 6.15: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \dots 5$ на SVD-векторах для русских текстов.

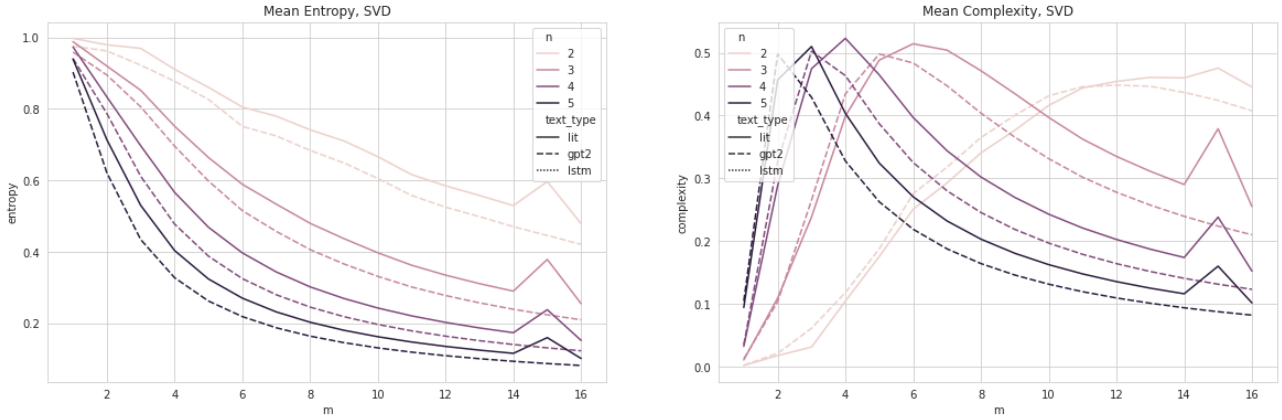

Рис. 6.16: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \dots 5$ на SVD-векторах для английских текстов.

Skip-gram, $m = 1, n = 3$. Полные таблицы результатов для разных вариантов векторных представлений и значений m, n приведены в репозитории.

	все боты		**LSTM**		**GPT**	
Язык	Train	Test	Train	Test	Train	Test
вьетнамский	0.981	0.989	1.0	1.0	0.991	0.995
русский	0.879	0.890	0.991	0.992	0.889	0.893
английский	0.937	0.965	0.999	1.0	0.997	1.0

Таблица 6.4: Точность классификаторов по метрикам энтропии и сложности

	все боты		**LSTM**		**GPT**	
Язык	Train	Test	Train	Test	Train	Test
вьетнамский	0.979	0.968	1.0	1.0	0.991	0.995
русский	0.828	0.844	0.991	0.992	0.878	0.881
английский	0.913	0.948	0.998	1.0	0.995	1.0

Таблица 6.5: F-мера классификаторов по метрикам энтропии и сложности

На рисунках 6.17, 6.18 представлена динамика изменения точности классификаторов в зависимости от изменения параметров m и n. На всех языках векторные представления Word2Vec (Skip-gram и CBOW) приводят к похожим результатам. В свою очередь, SVD-представления для вьетнамского в среднем лучше Word2Vec, а для английского, наоборот, хуже. Для русского результаты для SVD-вектора схожи с Word2Vec. Такую динамику и различие между видами векторных представлений можно связать с значениями параметров m, n, при которых тексты попадают в область хаотических процессов. Вышеописанные границы, представленные на рисунках 6.8-6.10 отличаются для SVD: для вьетнамского языка больше допустимых значений (наилучший классификатор как раз получился на SVD при $n = 3, m = 3$), тогда как для английского при всех значениях m, n тексты не попадали в область хаоса (при этом точность классификаторов на SVD существенно ниже). Для русского же область значений параметров m, n схожа, что также можно сопоставить с тем, что результаты классификаторов получились близки.

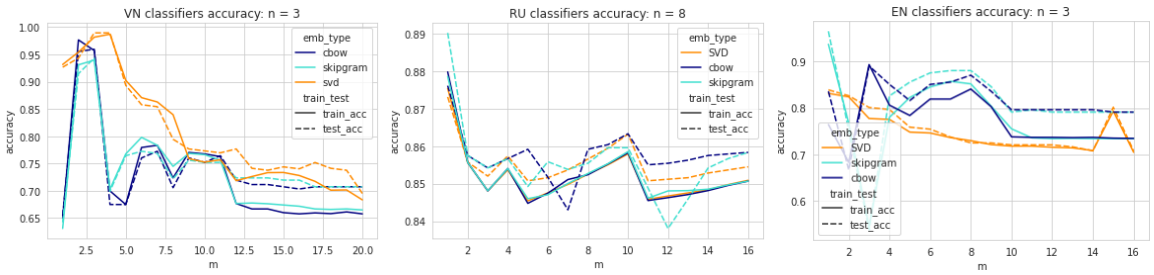

Рис. 6.17: Точность классификаторов в зависимости от m. Слева — для вьетнамского, $n = 3$, посередине — для русского, $n = 8$, справа — для английского, $n = 3$. При данных n точность классификации на тестовой выборке достигает максимума.

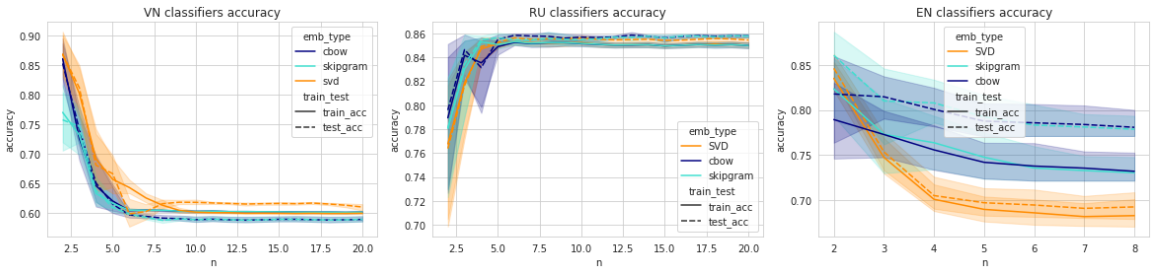

Рис. 6.18: Точность классификаторов в зависимости от n (усредненная по m). Слева — для вьетнамского, посередине — для русского, справа — для английского.

7 Заключение

В рамках исследования задачи идентификации ботов были применены два подхода — кластерный анализ и построение плоскости энтропии-сложности. Оба метода дали значимые результаты — точность классификаторов с применением этих методов получается от 90% и выше. В кластеризации наилушим получился алгоритм Уишарта, при этом применение нечеткости к данным также улучшает классификатор. В методе построения энтропии-сложности были выявлены различия в SVD-представлениях для разных языков, тем не менее, данные метрики успешно разделяют как простых, так и сложных ботов. Код и результаты экспериментов приведены в репозитории работы[10].

Дальнейшим направлением работы можно предложить применение описанных подходов к данным из социальных сетей или отзывов. Для этого необходимо исследовать зависимости от длин текстов и, возможно, модифицировать подходы, поскольку в данной работе исследование проводилось с длинными текстами. Кроме этого необходимо продолжить данное исследование с другими языковыми группами (например, языки романской группы), поскольку данная работа показала, что результаты могут отличаться для разных языков.

[10] https://github.com/quynhu-d/Spot-the-Bot/

Благодарность

Выражаю благодарность своему научному руководителю, Громову Василию Александровичу, профессору департамента анализа данных и искусственного интеллекта, ВШЭ, за ценные советы и непрерывную поддержку при написании настоящей работы.

Также выражаю благодарность отделу суперкомпьютерного комплекса НИУ ВШЭ за предоставленные ресурсы для проведения вычислительных экспериментов [20].

Список литературы и источников

1. Dickerson J. P., Kagan V. and Subrahmanian V. S.. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? // IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 620–627, 2014.

2. Yafeng R. and Ji D.. Neural networks for deceptive opinion spam detection: An empirical study. // Information Sciences 385, pp. 213–224, 2017.

3. Wilcoxon F. Individual comparisons by ranking methods. // Breakthroughs in statistics, Springer, pp. 196–202, New York, NY, 1992.

4. Bellegarda J. R.. Latent semantic mapping: principles and applications. // Synthesis Lectures on Speech and Audio Processing. Vol. 3, No. 1, pp. 1–101, 2007.

5. Mikolov T., Chen K., Corrado G. and Dean J.. Efficient estimation of word representations in vector space. // arXiv preprint arXiv:1301.3781, 2013.

6. Rosso O. A., Larrondo H. A., Martin M. T., Plastino A. and Fuentes M. A.. Distinguishing noise from chaos. // Physical review letters, vol. 99, no. 15, 154102, 2007.

7. Cresci S.. A decade of social bot detection. // Communications of the ACM 63.10 (2020): 72-83, 2020.

8. Kang A. R., Kim H. K. and Woo J.. Chatting pattern based game bot detection: do they talk like us? // KSII Transactions on Internet and Information Systems (TIIS), vol. 6, no. 11, pp. 2866–2879, 2012.

9. Heidari M., James Jr H. and Uzuner O.. An empirical study of machine learning algorithms for social media bot detection. // In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), pp. 1–5, IEEE, 2021.

10. Cardaioli M., Conti M., Di Sorbo A., Fabrizio E., Laudanna S. and Visaggio C. A.. It's a Matter of Style: Detecting Social Bots through Writing Style Consistency. // In 2021 International Conference on Computer Communications and Networks (ICCCN), pp. 1–9, IEEE, 2021.

11. Chakraborty M., Das S. and Mamidi R.. Detection of Fake Users in Twitter Using Network Representation and NLP. // In 2022 14th International Conference on COMmunication Systems and NETworkS (COMSNETS), pp. 754–758, IEEE, 2022.

12. Minnich A., Chavoshi N., Koutra D. and Mueen A.. BotWalk: Efficient adaptive exploration of Twitter bot networks. // In Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining, pp. 467-474, 2017.

13. Chu Z., Gianvecchio S., Wang H. and Jajodia S.. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? // IEEE Transactions on dependable and secure computing, vol. 9, no. 6, pp. 811–824, 2012.

14. Wishart D.. A numerical classification methods for deriving natural classes. // Nature 221, pp. 97–98, 1969.

15. MacQueen J.. Some methods for classification and analysis of multivariate observations. // In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14, pp. 281-297, 1967.

16. Novák V., Perfilieva I. and Mockor J.. Mathematical principles of fuzzy logic. // Springer Science and Business Media, vol. 517, 2012.

17. Bezdek J. C., Ehrlich R. and Full W.. FCM: The fuzzy c-means clustering algorithm. // Computers and geosciences 10.2-3, pp. 191-203, 1984.

18. Calbet X. and López-Ruiz R.. Tendency towards maximum complexity in a nonequilibrium isolated system. // Physical Review E, 63(6), p.066116, 2001.

19. Xiong H. and Li Z.. Clustering Validation Measures. // in: C. C. Aggarawal and C. K. Reddy (Eds.) “Data Clustering: Algorithms and Applications”, Chapman and Hall/CRC, pp. 571–605, 2014.

20. Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I. HPC Resources of the Higher School of Economics // Journal of Physics: Conference Series. 2021. Vol. 1740, No 1. P. 012050.

A Приложения

Примеры текстов ботов

LSTM-тексты

- **Вьетнамский**: *theo thì chiều thấy chuyện trong thuyền đến cho chỉ thì thấy thể cho chúng cho trong chàng chiều đi*
- **Русский**: *под собой в возветил в руках и стало сказал он на положить он признавал от нем высоко совсем конечно в небесной в странное вообще под вам все старика, страстных сердце с стеной получила в получила в сердце на столование*
- **Английский**: *and the body of the matter with the man in the way to the particularly wanting the state of the probably to the street with a word and with the beautiful ends and*

GPT-тексты

- **Вьетнамский**: *Trong tiếng Anh còn có các nghĩa: "Thực tế, vũ trụ phát triển vượt bậc trong thời đại ngày nay đồng nghĩa với việc nó sẽ tạo ra giá trị to lớn cho sự sống trên Trái Đất".*
- **Русский**: *послушайте, а что это за штука такая? — Это? — переспросил я плевательское отношение к людям, к их проблемам, к их нуждам*
- **Английский**: *As I became more intelligent at that time it took me much longer to get myself into what I actually believed in.*

Кластеризация

Метрики кластеризации

$$RMSSTD = \{\sum_i \sum_{x \in C_i} ||x - c_i||^2 / [m \sum_i (n_i - 1)]\}^{1/2}$$

$$RS = (\sum_{x \in X} ||x - c||^2 - \sum_i \sum_{x \in C_i} ||x - c_i||^2) /_{x \in X} ||x - c||^2$$

Здесь X — набор кластеризуемых объектов, C_i — i-ый кластер, $n_i = |C_i|$ — число объектов в i-ом кластере, m — размерность объектов.

RMSSTD оценивает компактность кластеров, чем меньше значение, тем компактнее кластера. RS оценивает разделимость кластеров, чем выше значение, тем дальше кластера расположены друг от друга.

Подбор параметров при построении нечётких чисел

Для построения нечётких чисел (описание метода в разделе 4.5.1) необходимо задать параметры ℓ, r и Δc. В качестве $Deltac$ было взято значение 1e-2, как стандартное отклонение в значениях координат в векторных представлениях рассматриваемых текстов. Тогда для подбора параметров ℓ, r рассматривается доля пересечения функций принадлежности: поскольку для n-граммов берутся минимумы по нечётким числам встречаемых слов, необходимо подобрать такие значения, чтобы при пересечении нечётких чисел не получились только нулевые функции принадлежности.

На рисунке A.1 можно увидеть, что доля пересечения, начиная примерно с $\ell = r = 0.1$, выходит на константу и равняется 0.5. Именно это значение и выбрано далее при построении нечётких чисел для кластеризации.

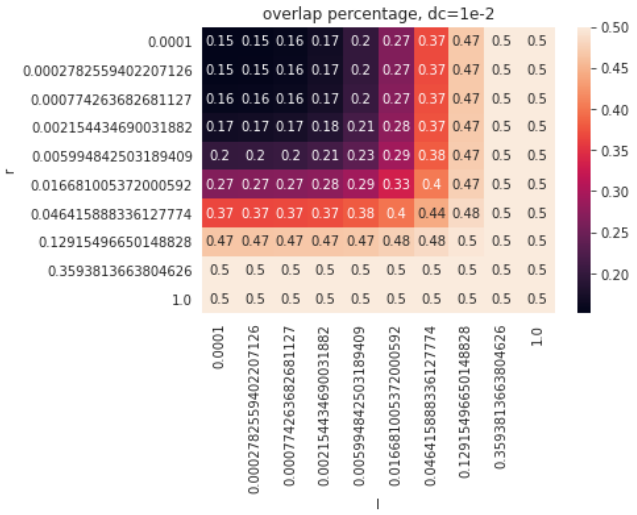

Рис. А.1: Доля пересечения нечётких чисел в зависимости от параметров ℓ, r.

Энтропия и сложность

Координатное упорядочение

Кроме описанного в разделе 4.6 модификации метода энтропии-сложности для многомерного случая также был рассмотрен следующий вариант: при построении порядковых паттернов будем упорядочивать m-мерные точки покоординатно. Сначала рассматривается первая координата, если для двух точек значение в этой координате равны, то они сравниваются уже по второй координате, и т.д.. Следующие результаты приведены для текстов на вьетнамском языке.

На рисунке А.2 можно заметить, что допустимые значения n при $m = 8$ (т.е. те значения, при которых литературные тексты не попадают в область шума/детерминированности) получаются высокими: от 9 до 11. При этом средние значения энтропии и сложности для разных типов текстов крайне близки.

На плоскости энтропии-сложности с данным вариантом упорядочения тексты разделяются плохо (см. рис. А.3), поэтому вместо него было принято решение использовать описанный в разделе 4.6 подход, рассматривающий

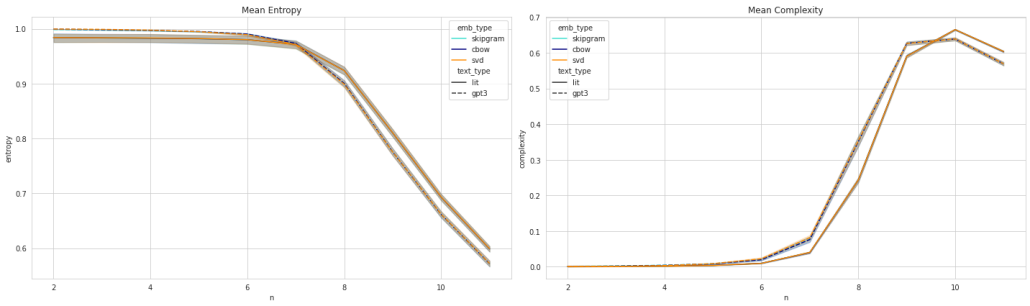

Рис. A.2: Изменение энтропии (слева) и сложности (справа) в зависимости от n, $m = 8$.

одновременно значения всех координат.

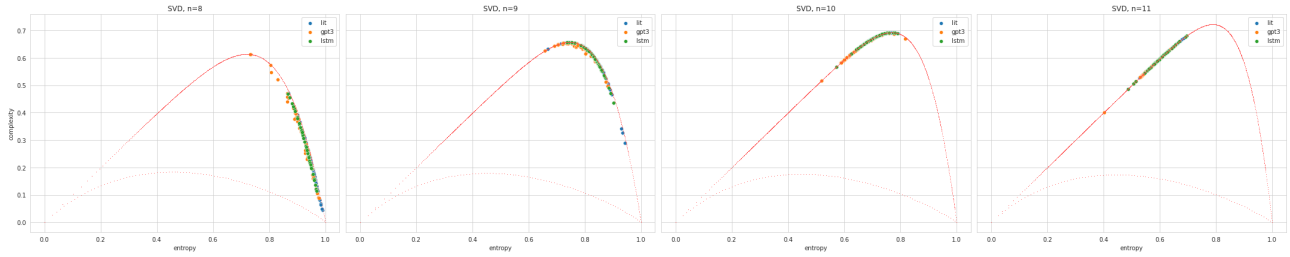

Рис. A.3: Плоскость энтропии-сложности, $m = 8$, $n = 8, \ldots, 11$

Графики изменения энтропии и сложности на Word2Vec представлениях

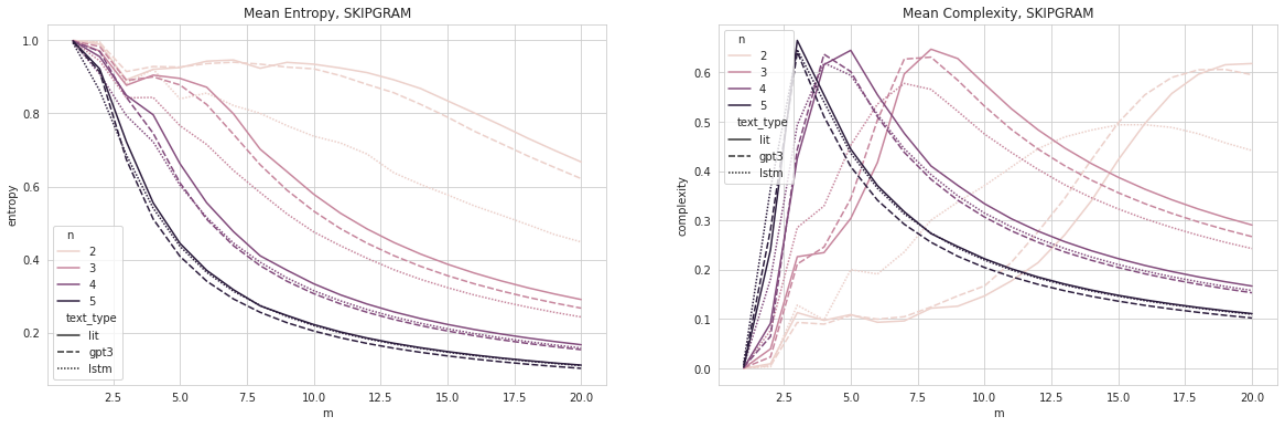

Рис. A.4: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \ldots 5$ на Skip-gram-векторах для вьетнамских текстов.

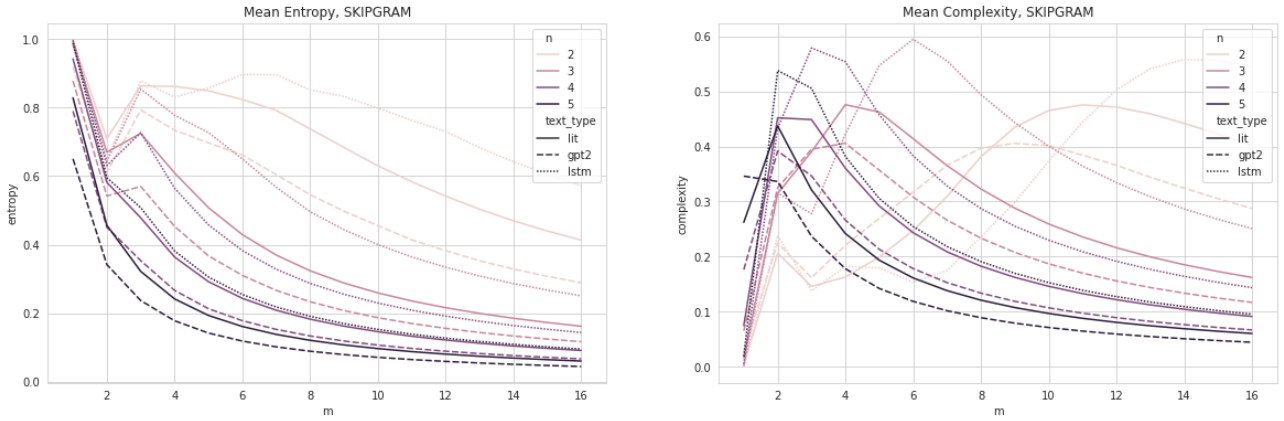

Рис. A.5: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \ldots 5$ на Skip-gram-векторах для русских текстов.

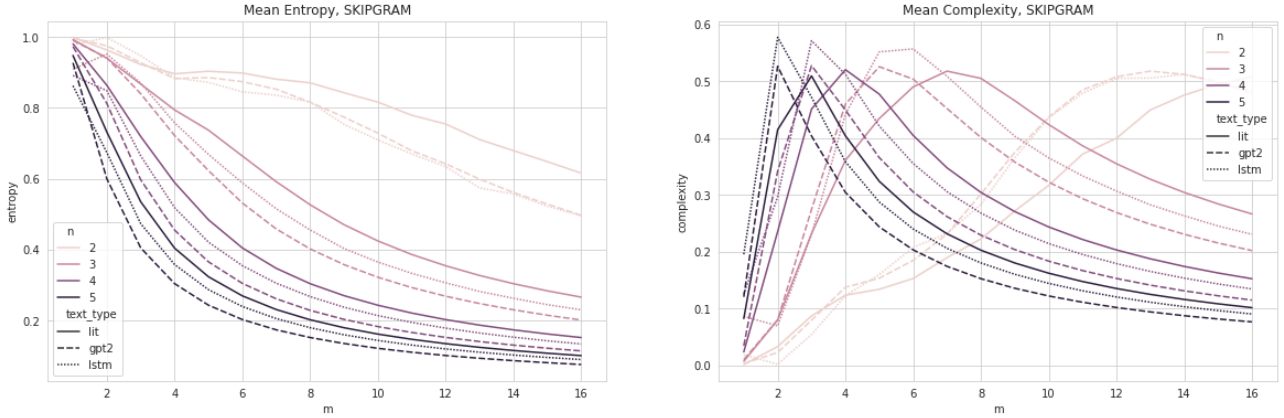

Рис. A.6: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \ldots 5$ на Skip-gram-векторах для английских текстов.

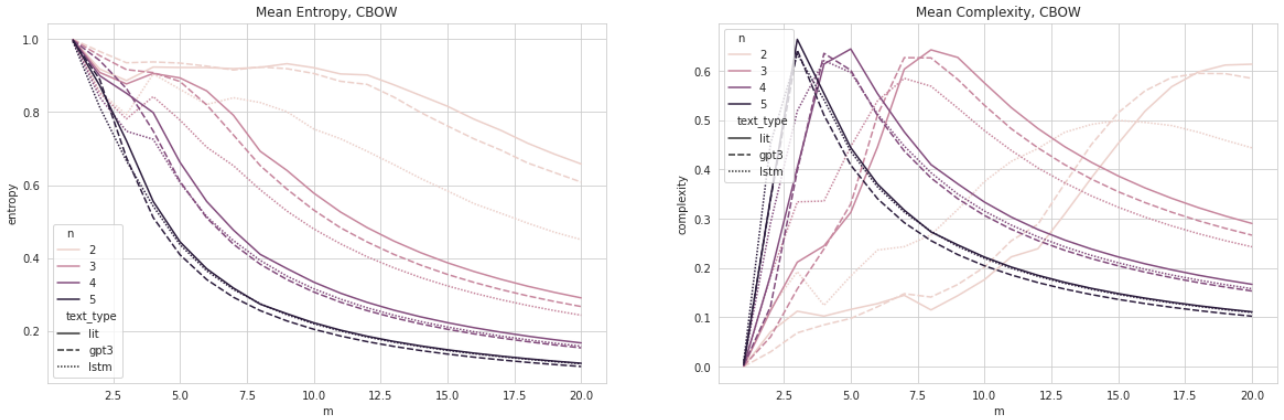

Рис. A.7: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \ldots 5$ на CBOW-векторах для вьетнамских текстов.

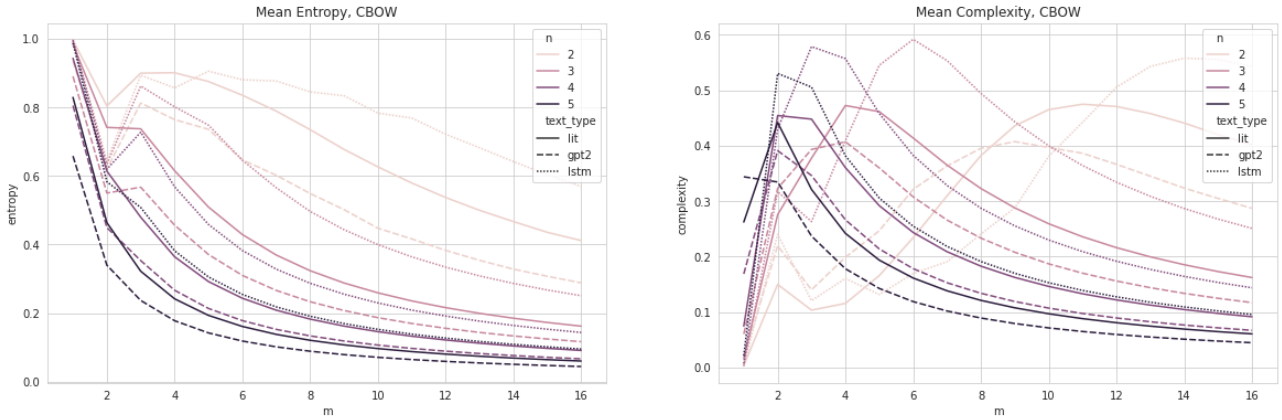

Рис. A.8: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \ldots 5$ на CBOW-векторах для русских текстов.

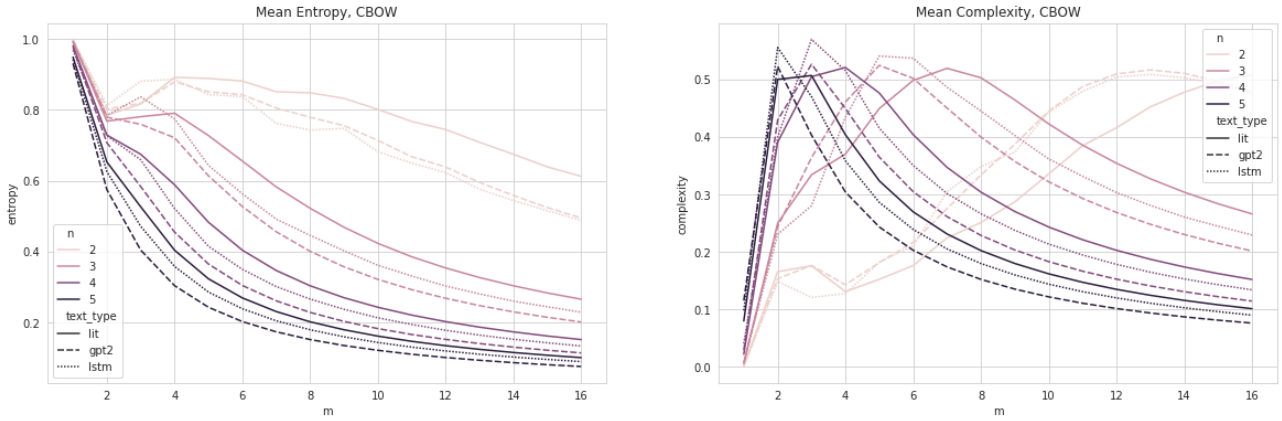

Рис. A.9: Зависимость энтропии (слева) и сложности (справа) от значения m при $n = 2, \ldots 5$ на CBOW-векторах для английских текстов.