УДК 004.02

СРАВНЕНИЕ МЕТОДОВ ВЕКТОРИЗАЦИИ ТЕКСТОВ С СОХРАНЕНИЕМ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ

Бородаенко Д.В.

Магистрант 2 курса,

Московский Государственный Технический Университет имени Н. Э. Баумана,

Москва, Россия

Погудина А.С.

Магистрант 2 курса,

Финансовый Университет при Правительстве РФ,

Москва, Россия

Аннотация

В статье рассматриваются основные подходы к предварительной обработке текстовых данных и векторизации. В качестве методов векторного представления слов описаны самые популярные и практически наиболее применимые – Word2Vec и fastText. На практике продемонстрировано применение алгоритма Word2Vec с целью сохранения семантической близости текстов из одной научной сферы.

Ключевые слова: векторизация, нейронная сеть, Word2Vec, fastText классификация.

COMPARISON OF TEXT VECTORIZATION METHODS WITH SEMANTIC PROXIMITY

Borodaenko D.V.

Student,

Bauman Moscow State Technical University,

Moscow, Russia

Pogudina A.S.
Student,
Financial University under the Government of the Russian Federation
Moscow, Russia

Annotation
The article defines the main approaches to the preliminary processing of text data and vectorization (Word2Vec and fastText). Word2Vec is used to demonstrate the preservation of the semantic closeness of texts from one scientific field.
Keywords: vectorization, neural network, Word2Vec, fastText, classification.

При обработке текстов естественного языка, информация может быть получена из различных источников и содержать значительное количество шума. В отличие от человека, который легко узнает разные слова в различных текстовых массивах, машина на современном этапе развития техники не способна легко это выполнять. Проблема усугубляется тем, что количество слов, которыми пользуются люди значительно отличается от количества слов, которые содержатся в различных словарях. Например, только в современных словарях европейских языков содержится около 340 тысяч слов. При этом с разговорной речи люди пользуются значительно большим количеством слов (до 1 миллиона). Такая большая разница возникает из-за постоянного возникновения новых слов на базе существующих. При этом человек легко распознает новые слова в отличие от машины. Тексты могут включать жаргонизмы, слова-паразиты. Более того, в обиходе люди используют различные слова, которые обозначают торговые марки и т.п. По этой причине многие задачи, связанные с обработкой текстов, по настоящее время могут быть выполнены только человеком. Однако все большее количество задач,

связанных с обработкой текстовых массивов, удается реализовать при помощи современных интеллектуальных технологий. Например, уже имеются достаточно успешные примеры реализации таких задач, как краткое изложение информации, создание нового текста на базе имеющегося, определение схожести информации в различных текстах.

Для того, чтобы реализовать многие, на первый взгляд не решаемые при помощи интеллектуальных технологий задачи с использованием современных технологий, прибегают к предварительной подготовке данных. Уже разработаны и опробованы на практике различные подходы, среди которых можно отметить токенизацию, стемминг, лемматизацию, обработка словосочетаний.

Токенизация представляет собой способ предварительной обработки текста, при котором текстовый массив переводится в нижний регистр, разбивается на отдельные слова, именуемые токенами. После этого производится удаление любых символов, отличных от цифр, букв и пробела. Частицы, предлоги обычно также удаляют, потому что они встречаются в любых текстах независимо от смысла. Подобный вид обработки можно производить на любом языке. Однако подобная обработка недопустима в ряде областей (биохимия), в которых символы, отличающиеся от цифр, букв или пробела могут иметь специфическую информационную составляющую.

Стемминг. Морфология множества языков может быть описана через некоторое конечное количество правил. Например, могут быть описаны подходы к определению времени глаголов, к использованию родов. На основании подобных правил через выполнение обратных действий, определенных правилами конкретного языка, возможной становится вычленение основы слова. Это может быть осуществлено через удаление суффиксов приставок и т.п. Данный подход позволяет существенно облегчить автоматизированную обработку текстов, включая работу с их смысловой составляющей. Однако при работе с такими сложными флективными языками,

как русский, результат далеко не всегда оказывается хорошим. Отдельные слова при урезании могут образовать совершенно другую сущность (ВОДитель – ВОДа). В русском языке также имеются слова, которые при изменении изменяются полностью (иду – шел).

Лемматизация представляет собой приведение формы слова к ее лемме, т.е. к словарной форме этого слова. Например, леммой для существительных является именительный падеж единственного числа. Метод лемматизации в качестве предварительной обработки текстов используется с целью уменьшения размера словаря и получения более актуальной статистики по частоте встречаемости отдельных слов.

Методы стемминг и лемматизация относятся к методам нормализации при предварительной обработке. Нормализация – это обобщающий термин, описывающий процесс приведения слов к начальной форме.

Обработка словосочетаний может производится различными способами. Наличие информации о том, как слова встречаются относительно друг друга может существенно повысить качество автоматизированной интеллектуальной обработки. Например, интересным представляется выявление коллокаций. Этим термином называются словосочетания, которые обладают признаками целостной единицы. При использовании коллокаций выбор одного компонента производится по смыслу, а второго зависит от первого. Примеры коллокаций: внести изменения, дождь идет, выносить приговор. Коллокации широко распространены во многих языках, включая русский. Сходная ситуация возникает с идиомами (подложить свинью). Для поиска коллокаций или идиом рассматриваются слова в рамках одного предложения, которые отстоят друг от друга не более, чем на 5-6 слов.

Векторным представлением слов (word embedding) именуют совокупность подходов к моделированию языка и обучению представлений при обработке языка, имеющие целью сопоставление словам из определенного словаря векторов. Векторное представление слов представляет

собой вещественный вектор в каком-либо пространстве, имеющий невысокую фиксированную размерность. Слова-векторы являют собой численное представление слов.

Модели, включающие векторное представление слов, строятся с использованием машинного обучения на базе значительных текстовых массивов. В частности, модели могут достаточно хорошо обучаться на основе новостных сайтов или статей с портала «Википедия. Свободная энциклопедия» [3].

Векторное представление слов позволяет использовать математические операции. Геометрические отношения между получаемыми векторами отражают смысловые семантические связи между соответствующими словами.

Среди методов векторного представления слов в настоящее время наибольшее распространение получили следующие подходы:

1. Word2Vec

Данный метод предполагает построение пространства векторов слов на основе нейронных сетей. При построении каждому слову из значительного текстового массива определяется векторное представление. Для этого сначала формируется словарь, на базе которого вычисляется векторное представление для каждого из слов. Отличительным свойством данного метода является то, что векторное представление слов базируется на контекстной близости слов. Если различные слова встречаются рядом с одинаковыми другими словами, то можно сделать вывод об их контекстной близости или наличии схожего смысла. И речь не идет только об однокоренных словах. Например, слово «скала» будет ближе к слову «гора» нежели к слову «пластмасса». Такие слова обладают высоким уровнем косинусного сходства (cosine similarity), которое представляет собой меру сходства между двумя отличными от нуля векторами внутреннего пространства произведений, которое измеряет косинус угла между ними:

$$\text{Similarity}(A, B) = \cos\theta = \frac{A*B}{||A||*||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \qquad (1)$$

Векторное представление слов находится в высокой степени зависимости от размера обучающего текста, его тематики, а также от расположения в нем слов. Важнейшим недостатком метода Word2Vec является невозможность представить слова, которые отсутствуют в обучающем текстовом массиве.

Подробнее метод векторного представления Word2Vec будет рассмотрен далее.

2. fastText. Данный метод векторного представления может быть эффективно применен для многих слов, которые отсутствовали в обучающем текстовом массиве. Основой метода является применение N-грамм символов. В каждом случае выбирается свое число N. Рассмотрим представление слова «исследование» в виде 3-грамм и в виде 4-грамм (Таблица 1).

Таблица 1 – Пример представления слова в виде 3-грамм и в виде 4-грамм

Слово	3-граммы	4-граммы
Исследование	Исс, ссл, сле, лед, едо, дов, ова, ван, ани, ние	Иссл, ссле, след, ледо, едов, дова, ован, вани, ание

Благодаря тому, что многие N-граммы встречаются не только в словах исходного текста, векторные представления могут быть сформированы и для слов, отсутствующих в обучающем текстовом массиве.

При помощи векторного представления слов могут быть решены многие фундаментальные и прикладные научные задачи. Например, на основе 3300 тысяч аннотаций к различным научным статьям за период с 1922 года по 2018 год в области материаловедения ученые из США сформировали векторное представление 500 тысяч слов.

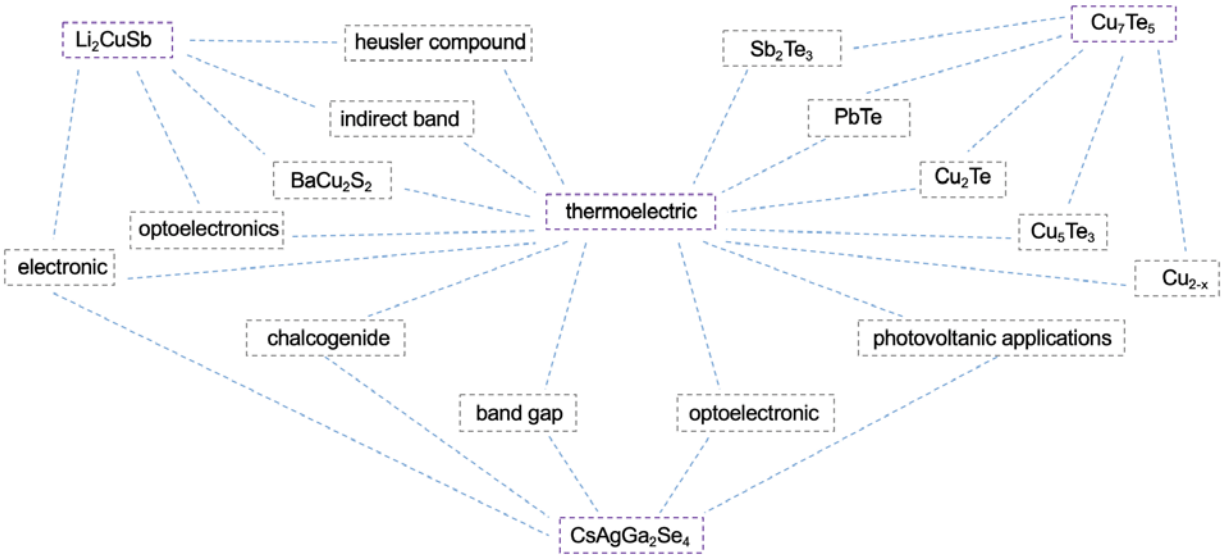

Рис. 1 – Визуализация векторного представления свойств материалов [8]

Работа позволила не только с высокой степенью достоверности описать известные свойства множества материалов, но и предсказать новые их свойства без теоретических знаний физики и химии (Рис.1). Векторное представление позволяет, например, определить понятие антиферромагнитизма через вычитание вектора из ферромагнетизма NiFe и прибавления IrMn.

Модели, базирующиеся на векторном представлении слов, оказываются весьма эффективными для развития интернет-поисковиков, а также при обучении различных голосовых помощников.

Создание модели на базе алгоритма Word2Vec предполагает решение следующих основных задач:

- перевод данных в one-hot кодировку;
- создание модели, получающей на вход и выход one-hot векторы;
- определение функции потерь, которая предсказывает правильное слово для целей оптимизации модели;

- определение качества модели через выявление того, что схожие слова обладают схожими векторными представлениями.

Рассмотрим такой обязательный компонент процесса, как embedding layer, который хранит в словаре вектора всех слов (Рис.2). В данном случае речь идет о громадной матрице, размер (embedding size) которой определяется как количество слов в словаре, умноженное на размерность пространства сжатого векторного представления слов. Отметим, что размер матрицы настраивается отдельно. Матрица образуется случайным образом, а затем настраивается в результате оптимизации.

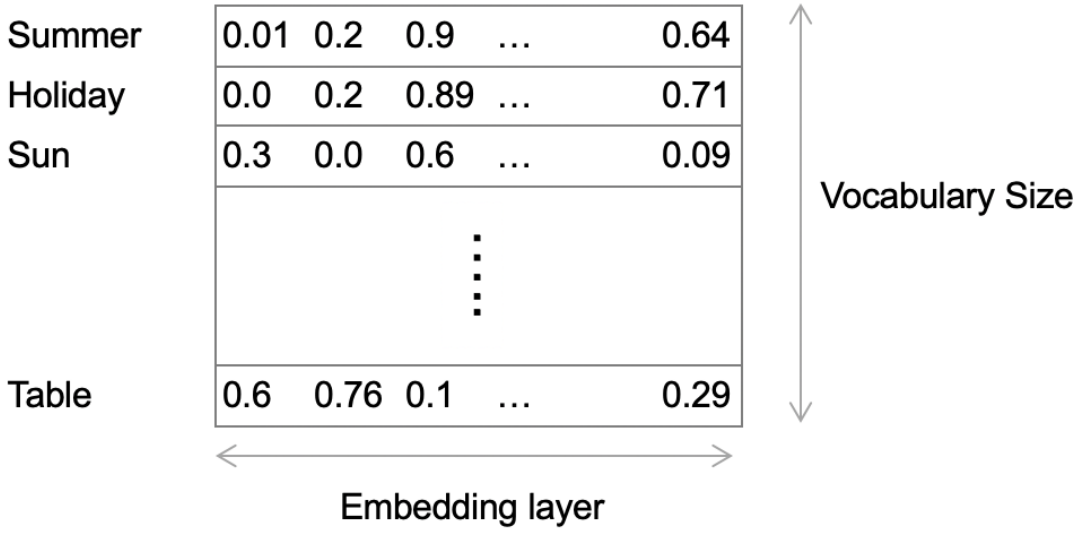

Рис. 2 – Пример матрицы embedding layer [Авторская разработка]

Нейросеть в рамках алгоритма обучается на входных данных. Изучение А. Кутузовым и И. Андреевым семантической близости в русском языке показало, что морфология русского языка не представляет собой препятствия для обучения моделей с использованием алгоритма векторного представления Word2Vec на русских корпусах. [7].

Чем больший размер имеет исходный текстовый массив, тем лучше получается модель, однако размер оказывает влияние на скорость алгоритма. Нейросеть, получив входной вектор слова, предсказывает результат, который представляет собой распределения вероятностей слов оказаться в контексте

входного слова. Реализуется это на общем множестве слов. На следующем шаге через использование функции потерь производится награждение модели за правильную классификацию либо накладывается штраф за неверную (Рис.3).

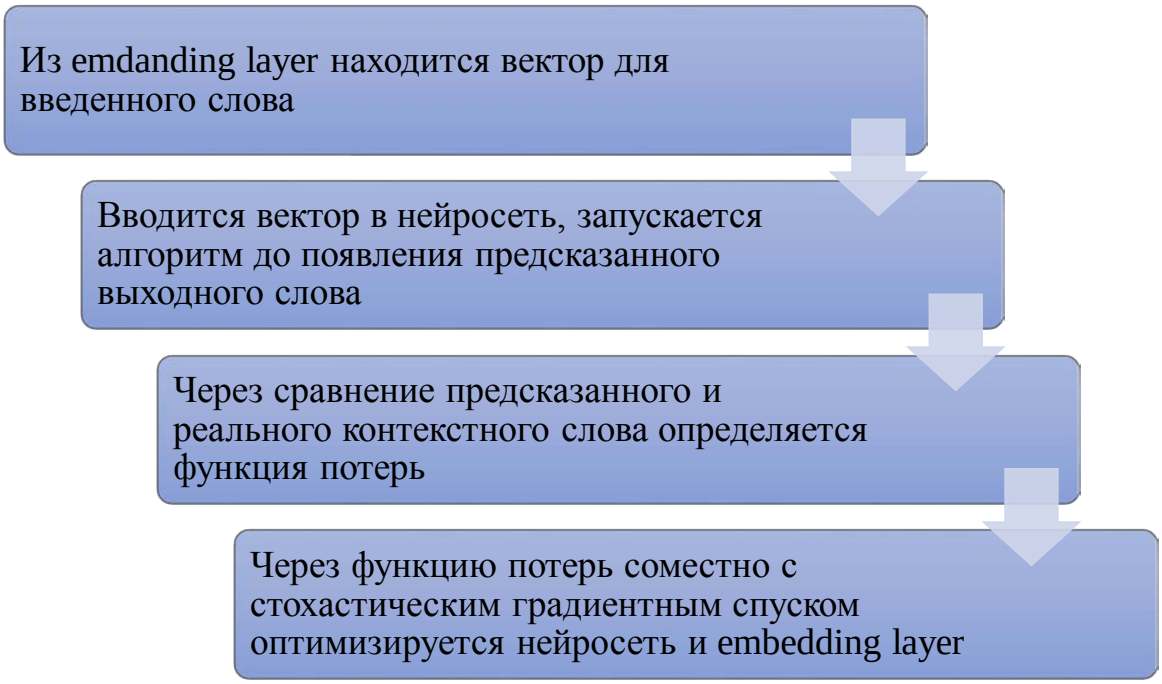

Рис. 3 – Обучение нейросети в рамках алгоритма Word2Vec [Авторская разработка]

В качестве функции потерь могут быть использованы различные подходы. Для задач классификации хорошо себя зарекомендовала стандартная функция перекрестной энтропии (softmax cross entropy loss). Однако в реальных ситуациях из-за колоссального размера словарей использование этой функции затруднительно [6]. Чаще на практике применяется модификация функции sampled softmax loss, при использовании которой сначала определяется стандартная функция перекрестной энтропии между реальным значением слова для целевого слова и значением для предсказанного слова. После этого вводится кросс-энтропийная потеря k негативных семплов, которая представляет собой целевое слово и слово вне контекстного окна. Негативные семплы предварительно отбираются в

соответствии с распределением шума. Функцию потерь можно определить на основании формулы:

$$L = \text{SigmoidCrossEntropy}(\text{Prediction}, \text{CorrectWord}) + \sum_1^K E_{\text{Noise ID}} \text{Sigmoid CrossEntropy}(\text{Prediction}, \text{Noise ID}) \quad (1)$$

где L- функция потерь, SigmoidCrossEntropy – представляет собой ошибку, которая определяется на выходном узле вне зависимости от остальных узлов [5, с.91].

В целом реализация алгоритма Word2Vec имеет, показанный на Рис.4.

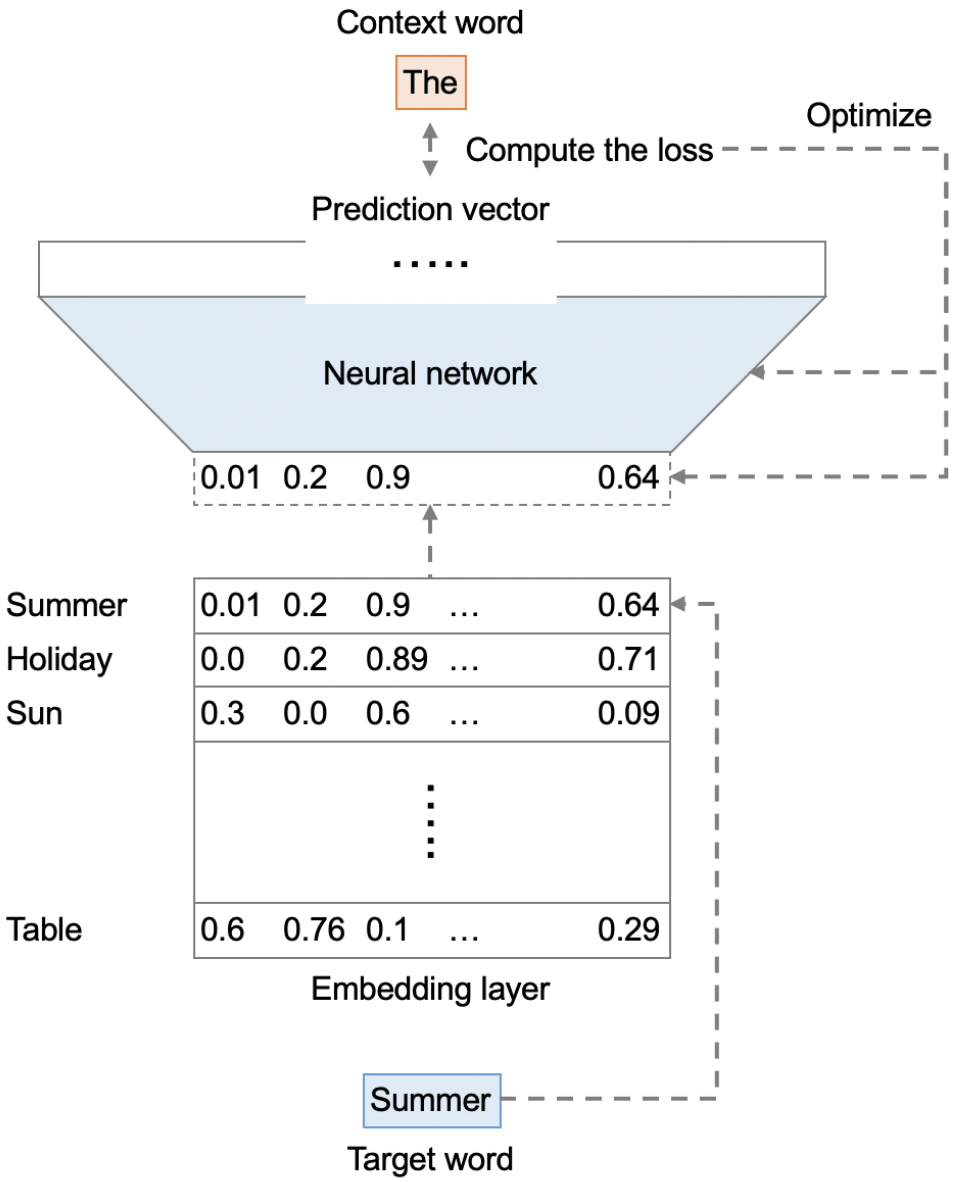

Рис. 4 – Схема работы алгоритма Word2Vec [Авторская разработка]

Алгоритм Word2Vec может использовать два основных подхода к обучению. При этом результат выполнения Word2Vec всегда сильно зависит от контекста. Один из двух используемых подходов получил название “непрерывного мешка со словами” (CBOW, continuous bag-of-words model). В данной модели по имеющемуся контексту производится подбор наиболее вероятного слова (Рис.5).

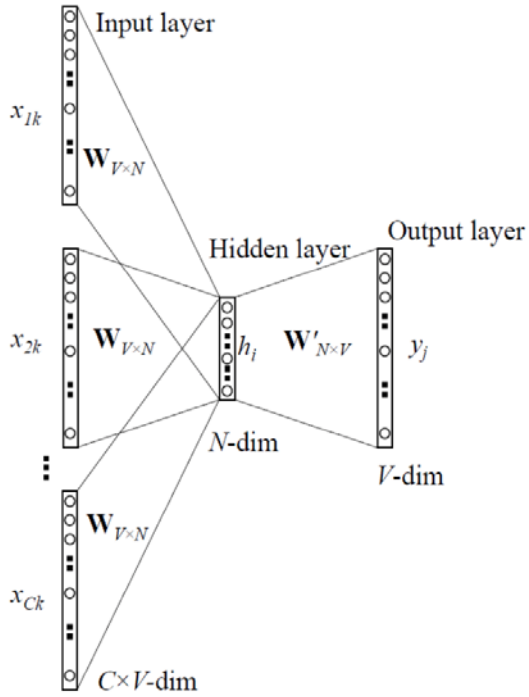

Рис. 5 – Схема реализации алгоритма CBOW [1, с.211]

Второй подход к обучению в рамках алгоритма Word2Vec называется Skip-gram. В рамках данной модели на основании имеющегося слова производится предсказание слов из контекста. (Рис.6)

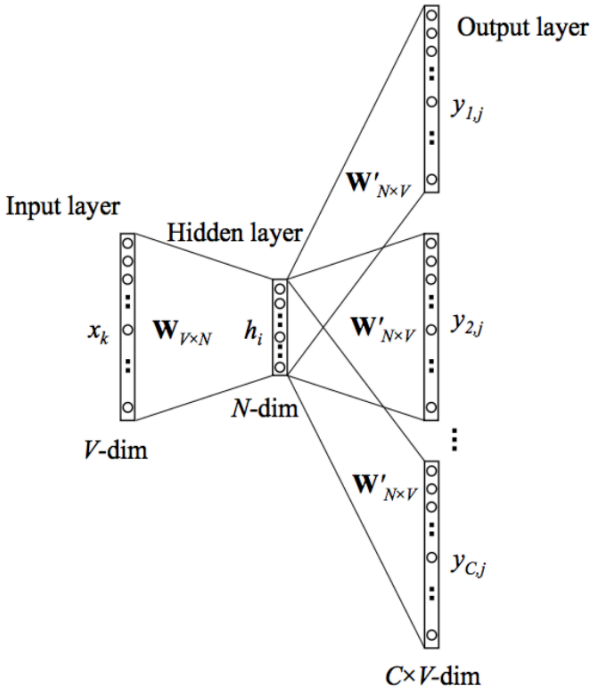

Рис. 6 – Схема реализации алгоритма Skip-gram [1, с.213]

Алгоритм векторного представления Word2Vec напрямую не даёт семантические отношения между словами. Тем не менее наиболее часто употребляемые в исходном текстовом массиве слова присутствуют в ассоциативном ряду, который алгоритм может возвратить в качестве подходящих слов к главному слову.

Напомним, что формирование ассоциативного ряда при использовании алгоритма Word2Vec для главного слова производится на основании косинусного сходства. При этом позиция выбранного слова в автоматически полученном ассоциативном ряду не является симметричной относительно позиции исходного главного слова в ассоциативном ряду, сформированном для выбранного ранее слова [4]. Иными словами, позиции слов, для которых предполагаются семантические отношения, несмотря на одинаковый обучающий текстовый массив, не совпадут. Позиции будут зависеть от того, какое слово выбиралось в качестве главного при запуске алгоритма.

Векторное представление слов являются может быть применено для решения различных задач. В настоящее время алгоритм Word2Vec активно используется в следующих направлениях:

- моделирования языков;
- тэгирование частей речи;
- машинный перевод;
- вопросно-ответные системы;
- поиск опечаток;
- поиск синонимов;
- ранжирование;
- чат-боты;
- обнаружение именованных сущностей;
- кластеризация;
- анализ тональности текста.

К настоящему времени уже имеется достаточное количество исследовательских работ, направленных на изучение особенностей алгоритма Word2Vec. Например, исследователями Wang C., Cao L., Zhou B., была проделана экспериментальная работа по реализации автоматического поиска синонимов в области медицины [10]. Работа однозначно подтвердила возможность применения представления слов в виде векторов на базе алгоритма Word2Vec в различных предметных областях.

Векторное представление слов, реализуемое при помощи алгоритма Word2Vec, несмотря на уже накопленное количество разнообразной информации по его применению, предполагает дальнейшее проведение обширной исследовательской работы в направлении лингвистики и статистики [10]. Накопление данных, выявление закономерностей через исследования будет способствовать росту качества извлекаемых сущностей. Например, опытным путем уже определено, что при решении задач

классификации типа «род-вид» с помощью алгоритма векторного представления слов Word2Vec выявляются следующие особенности:

- При оценке расстояния от главного слова до слова-кандидата, представляющего место в ассоциативном ряду, ранжированному по косинусной мере, получается, что наименьшее расстояние возникает в ситуации, когда главное слово является видом, а слово-кандидат – родом [4]. В остальных случаях расстояние оказывается большим.
- В абсолютном большинстве случаев частота встречаемости слов, представляющих собой «род» значительно превышает частоту слов, представляющих «вид». Например, среди пар, сформированных исследователями на основании Викисловаря, такие пары составили 88% от общего количества пар типа «род-вид» [4]. Оставшиеся пары, по мнению исследователей М. С. Каряевой, П. И. Браславского и В. А. Соколова, следует относить к исключениям, так как они отображают слабую связь типа «род-вид» (Рис.7).

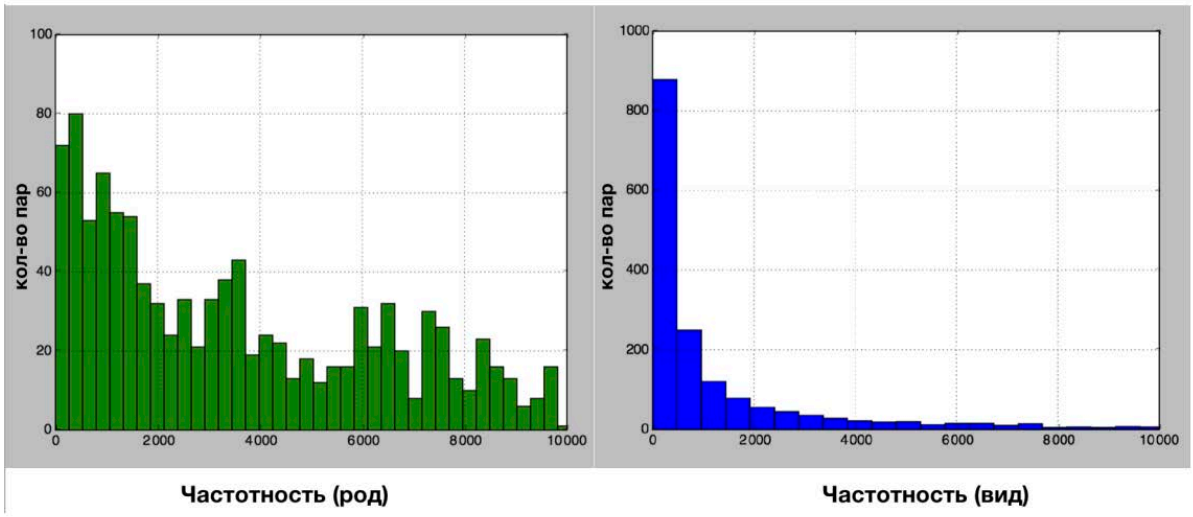

Рис. 7 – Распределение частоты встречаемости [4]

Теперь рассмотрим практическую реализацию алгоритма и проверим, возможно ли с помощью алгоритма Word2Vec сохранить семантическую близость текстов из одной научной сферы. В качестве входных данных используем статьи по трем темам: физике, биологии и экономике. Основной

задачей является проверка возможности кластеризации статей по тематикам после перехода от обычного текста к эмбедингам.

В связи с тем, что обучение модели с помощью алгоритма Word2Vec требует большой вычислительной мощности и колоссальных объемов данных, было принято решение взять предобученные модели. В библиотеке genism встроены таковые, однако они актуальны для текстов на иностранном языке. По этой причине были выбраны модели, подготовленные командой «RusVectōrēs» [9]. Для сравнения их качества и выбора наилучшей были использованы две модели, обученные по алгоритмам skip-gram на корпусе «Тайга» и cbow на корпусе «НКРЯ». Тайга – это открытый и структурированный веб-корпус русского языка, снабженный морфологической и синтаксической разметкой (подкорпус поэзии не использовался). НКРЯ - Национальный Корпус Русского Языка в полном объёме.

Перед обучением оба корпуса были токенизированы, разбиты на предложения, лемматизированы и размечены по частям речи при помощи UDPipe. Тэги частей речи соответствуют формату Universal PoS Tags (например, «печь_NOUN»). Стоп-слова (союзы, местоимения, предлоги, частицы) были удалены. Некоторые устойчивые и частотные словосочетания из двух слов (биграммы) были объединены в один токен через спецсимвол «::» (относится к именам собственным).

Для того, чтобы произвести векторизацию слов в тексте, необходимо произвести предварительную обработку. Для каждого из выбранных разделов было подготовлено по три научные статьи, которые были разбиты на абзацы. Пример обработки одного из них представлен в Таблице 2.

Таблица 2 — Пример исходного и обработанного текста

Исходный текст	Обработанный текст

Хотя все диффенбахии похожи между собой, существует много гибридов этого растения. Наиболее популярны такие разновидности, как диффенбахия прелестная (d. Amoena) с очень крупными (до 50 см в длину) листьями и диффенбахия сегуина (d. Seguina, больше похожая на пятнистую, но с более широкими листями.	хотя_SCONJ весь_DET диффенбахия_NOUN похожий_ADJ между_ADP себя_PRON существовать_VERB много_ADV гибрид_NOUN этот_DET растение_NOUN наиболее_ADV популярный_ADJ такой_DET разновидность_NOUN как_SCONJ диффенбахия_NOUN прелестный_ADJ d_X amoena_PROPN с_ADP очень_ADV крупный_ADJ до_ADP xx_NUM сантиметр_NOUN в_ADP длина_NOUN лист_NOUN и_CCONJ диффенбахия_NOUN егуить_NOUN d_X seguina_PROPN более_ADV похожий_ADJ на_ADP пятнистый_ADJ но_CCONJ с_ADP более_ADV широкий_ADJ лист_NOUN'

Алгоритм Word2Vec векторизует только отдельные слова (токены), поэтому сначала необходимо векторизовать их по отдельности и только после этого объединить вектора слов в один, который и будет является вектором целого абзаца. Самым оптимальным вариантом является подсчет среднего вектора из набора.

После проведения этих операций для каждого абзаца из статей получаем набор из векторов, готовых к дальнейшему анализу. По причине того, что размерность векторов равна тремстам, визуализировать подобные представления достаточно сложно. Воспользуемся алгоритмом понижения размерности, чтобы вектора можно было отразить на двумерной плоскости. Для этого был выбран алгоритм t-SNE (t-distributed Stochastic Neighbor Embedding) - Стохастическое вложение соседей с t-распределением [2, с.156].

Стоит отметить, что при векторизации целых абзацев, не все слова можно векторизовать, так как они отсутствуют в корпусах, на которых обучались модели. Так как корпус «Тайга» в несколько раз превышает корпус «НКРЯ», словарь такой модели получился примерно на 25% больше. По этой причине при обработке текстов первой моделью было удалено всего лишь 8 абзацев против 24 - у второй, которые состояли полностью из неизвестных для модели слов. Для удобства анализа вектора, характеризующие статьи разных научных сфер, отображаются различными цветами.

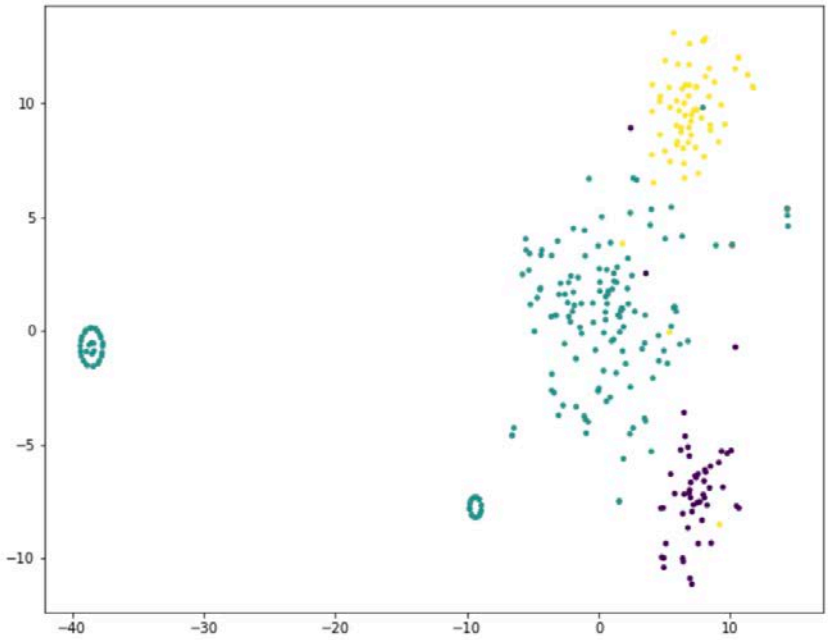

Рис. 8 – Визуализация векторного представления с помощью модели skip-gram [Авторская разработка].

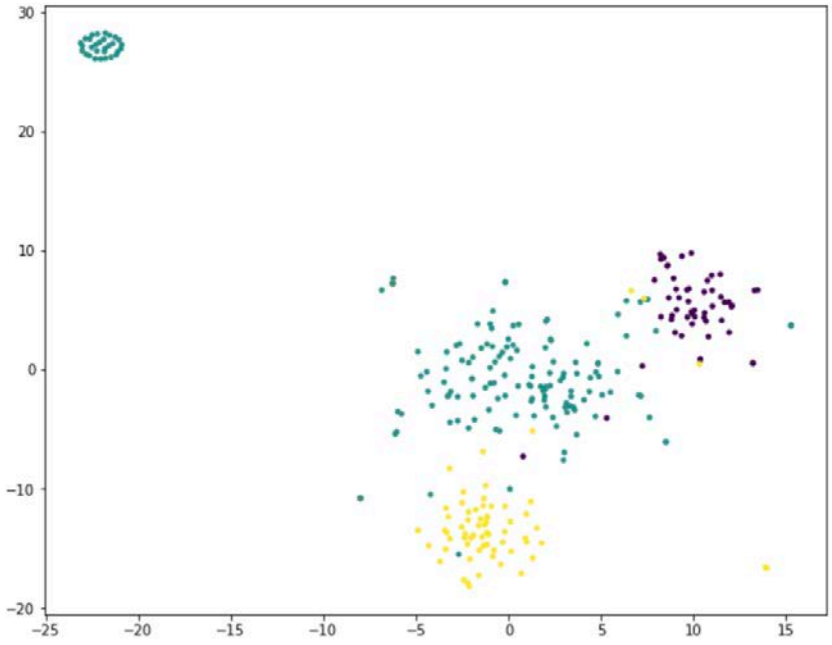

Рис. 9 – Визуализация векторного представления с помощью модели cbow [Авторская разработка].

Как видно из Рис. 8 и Рис. 9, в обоих случаях некоторые объекты сильно отделены от остальных и сгруппированы в плотные кластера. Как выяснилось в дальнейшем, это были абзацы из статей по физике, содержащие в себе числовые значения и формулы. Стоит отметить, что остальное облако точек, визуально разделяется на 3 образования, что демонстрирует сохранение близости между векторами из разных научных тем. Таким образом, можно наглядно убедиться в сохранении семантической близости.

У алгоритма t-sne есть один существенный недостаток – его невозможно спроецировать в обратную сторону, поэтому для того, чтобы сконцентрироваться на наибольшем облаке точек, было принято решение использовать алгоритм кластеризации dbscan (Density-based spatial clustering of applications with noise) - Основанная на плотности пространственная кластеризация для приложений с шумами. Этот алгоритм был выбран исходя из высокой плотности точек, отделенных от кластеров. С его помощью удалось выделить основное облако как шум, и оставить только нужные вектора (Рис. 10 и Рис.11).

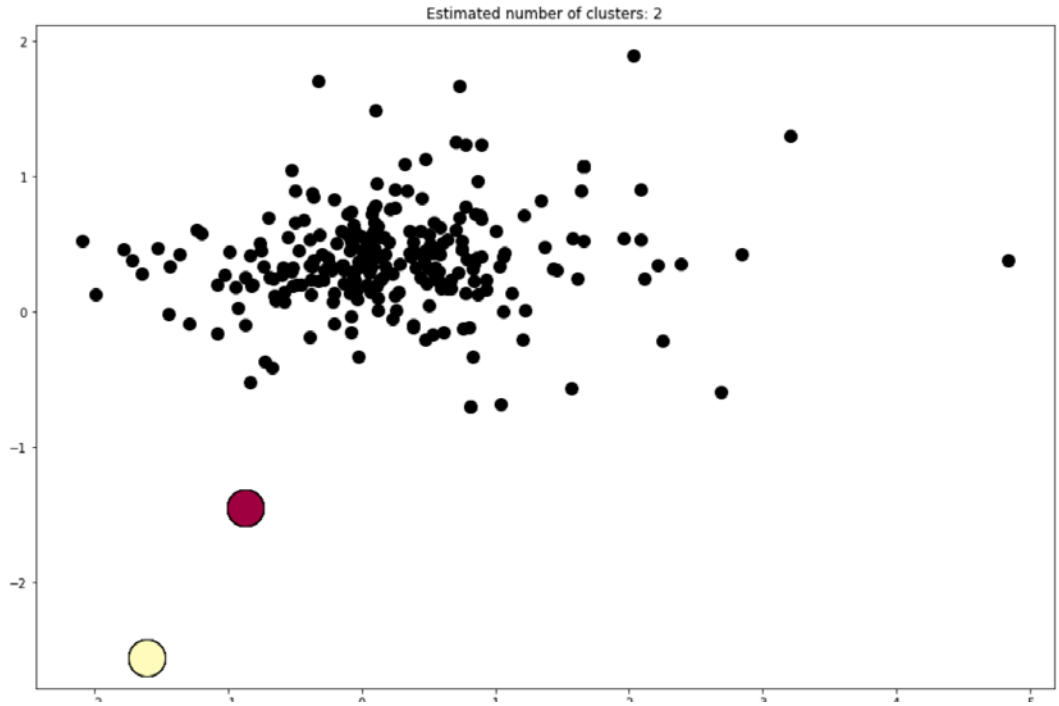

Рис. 10 – Визуализация кластеризации векторов с помощью dbscan на векторах алгоритма skip-gram [Авторская разработка].

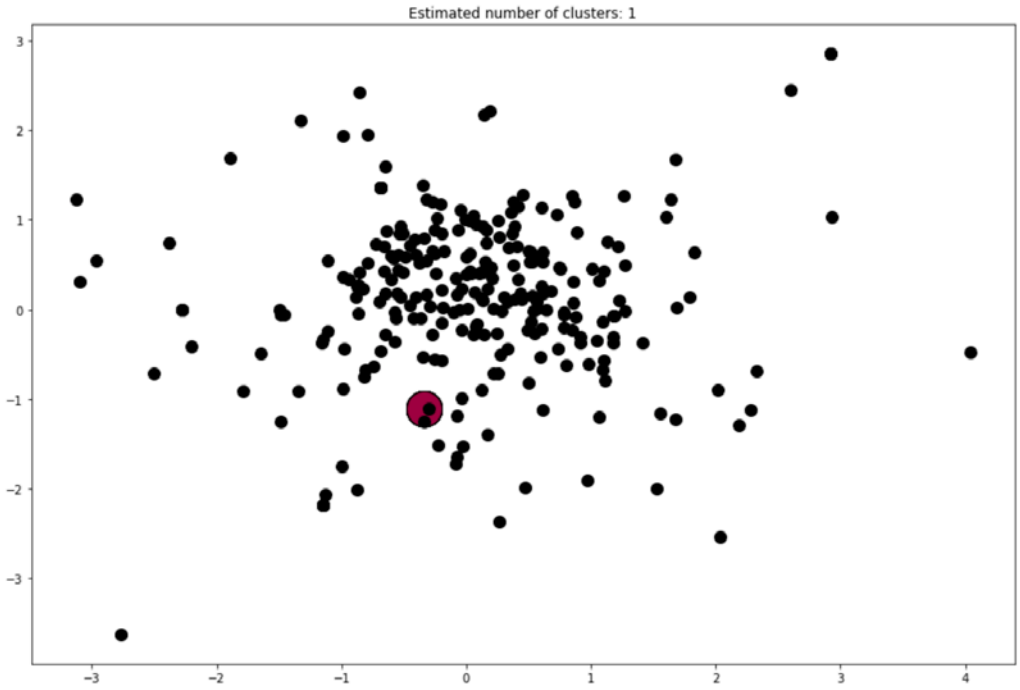

Рис. 11 – Визуализация кластеризации векторов с помощью dbscan на векторах алгоритма cbow [Авторская разработка].

Применив данный алгоритм к векторам обеих моделей, можем выявить самый плотный кластер (отмечен красным цветом на Рис. 10) из всего облака

точек, чтобы в дальнейшем удалить его. При тех же настройках, для векторов skip-gram модели, удалось обнаружить сразу 2 кластера.

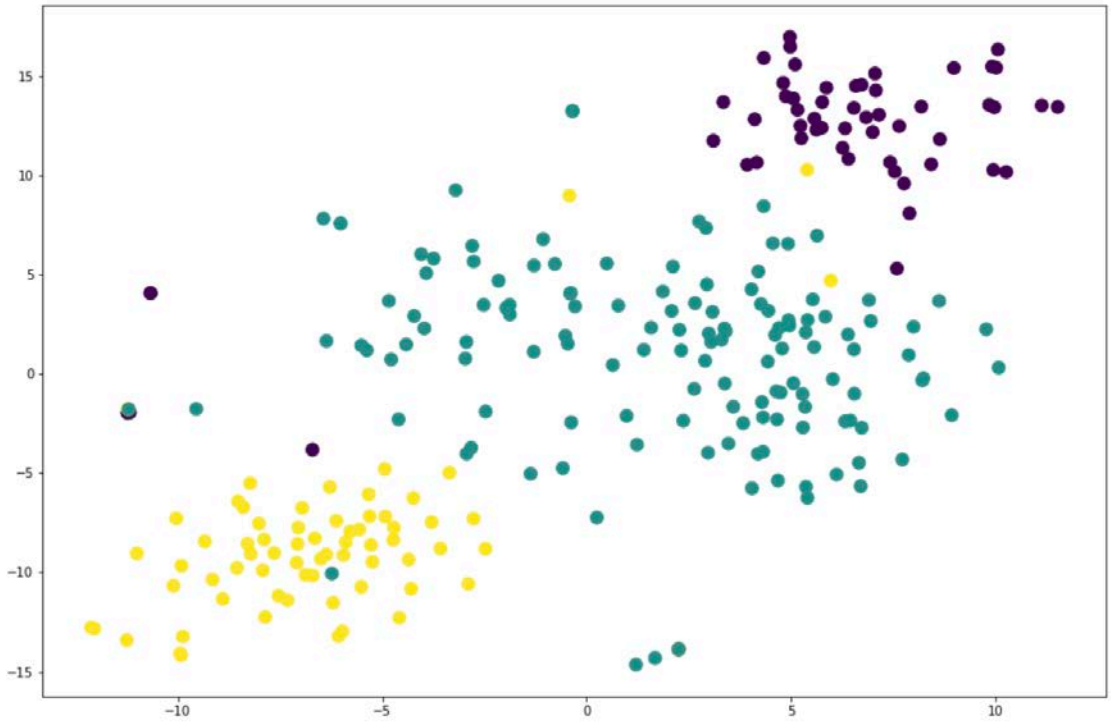

Рис. 12 – Визуализация векторного представления с помощью модели cbow после удаления лишнего кластера [Авторская разработка].

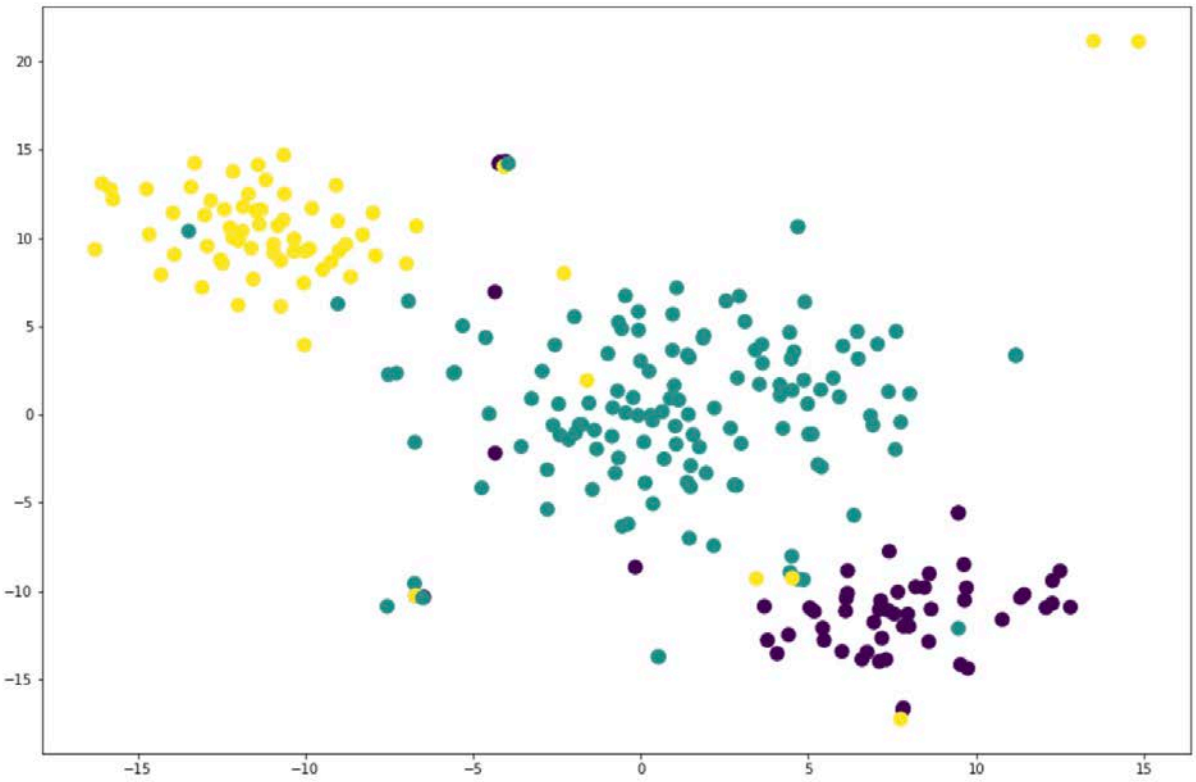

Рис. 13 – Визуализация векторного представления с помощью модели cbow после удаления лишнего кластера [Авторская разработка].

Для того, чтобы удостовериться в том, что эти кластера корректно разделяются, был применен метод k-средних ко всему облаку точек (Рис.12 и Рис.13). Данный алгоритм является одним из простейших алгоритмов кластеризации, однако он отлично справился с задачей разделения векторов по семантической составляющей.

Стоит отметить, что для векторов алгоритма skip-gram количество кластеров было увеличено до 5, так как метод подбора оптимального параметра (метод локтя) показал, что именно 5 кластеров – наилучший вариант разделения облака точек (Рис.14). Идея метода локтя (Elbow method) состоит в том, чтобы запустить кластеризацию k-средних в наборе данных для диапазона значений k и для каждого параметра k вычислить сумму квадратов ошибок. Затем строится линейная диаграмма SSE для каждого значения k.

Если линейный график выглядит как рука, то «локоть» на руке является наилучшим значением k.

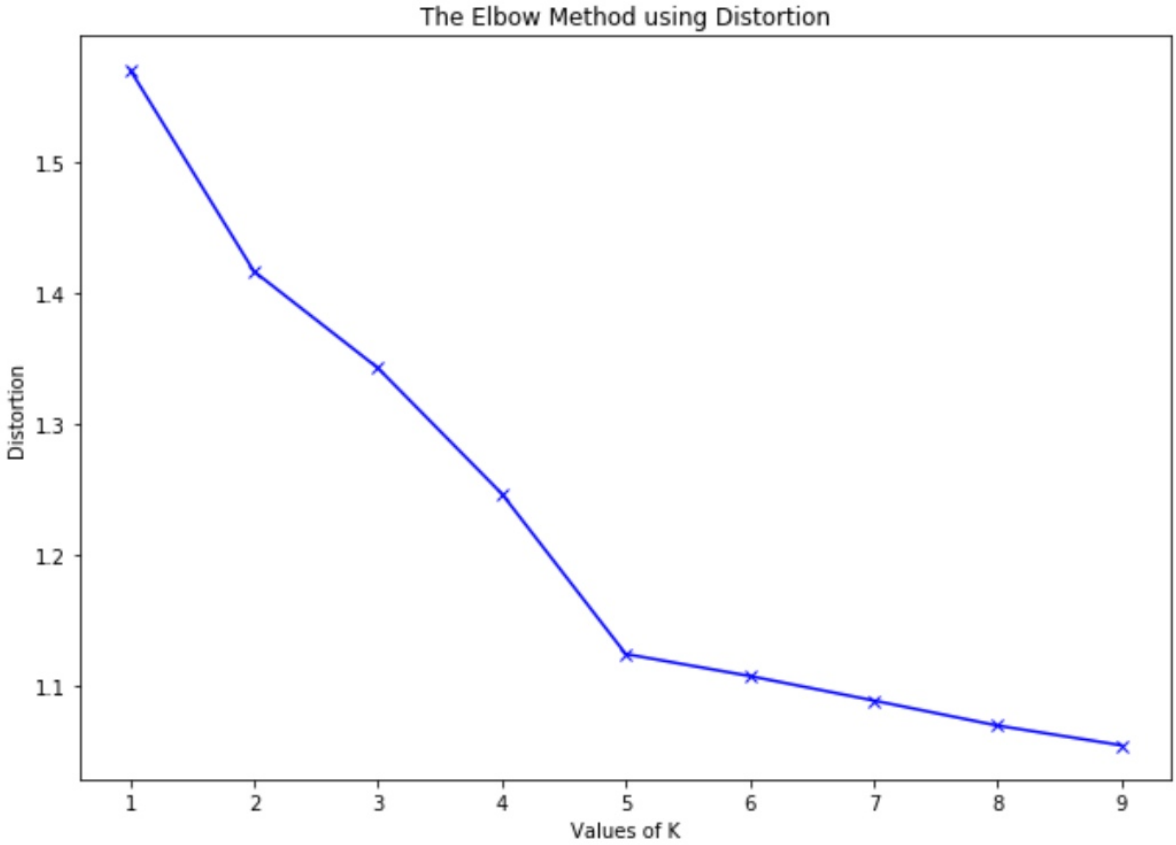

Рис. 14 – Elbow method для подбора оптимального параметра кластеризации [Авторская разработка].

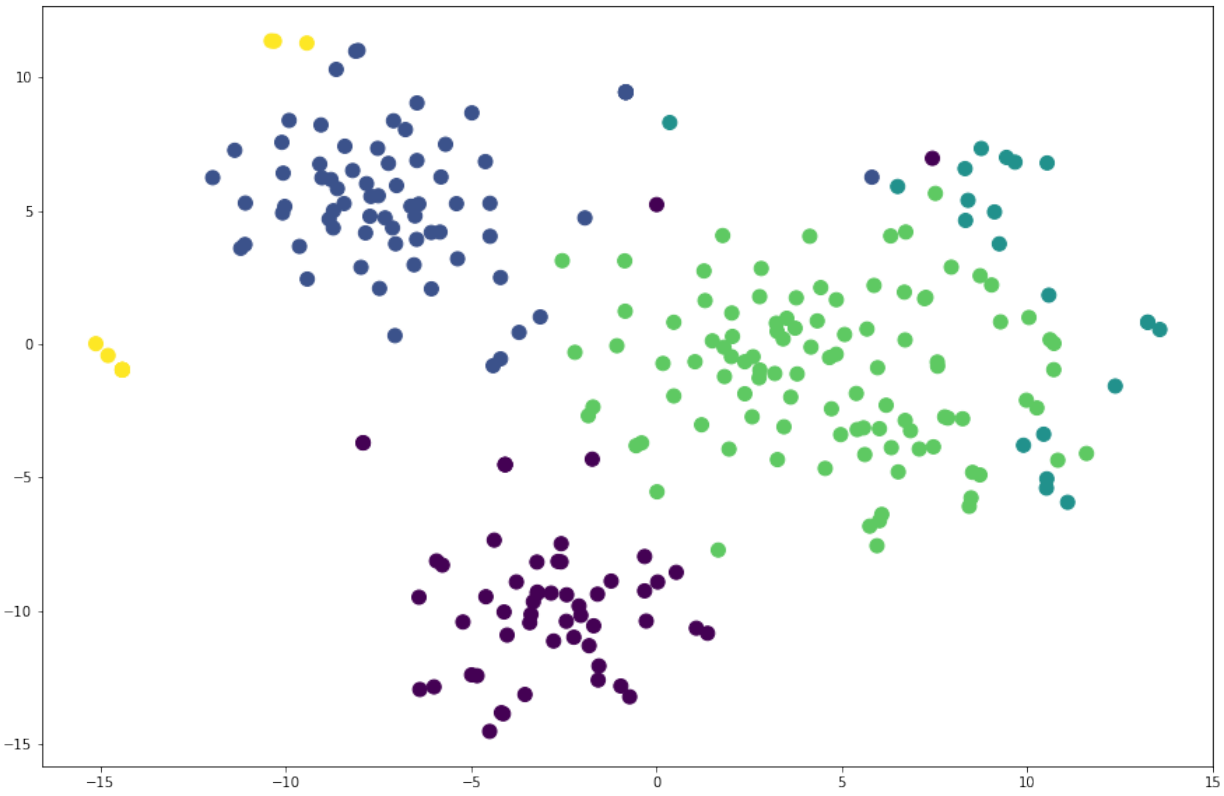

Рис. 15 – Визуализация кластеризации методом k-средних векторов skip-gram [Авторская разработка].

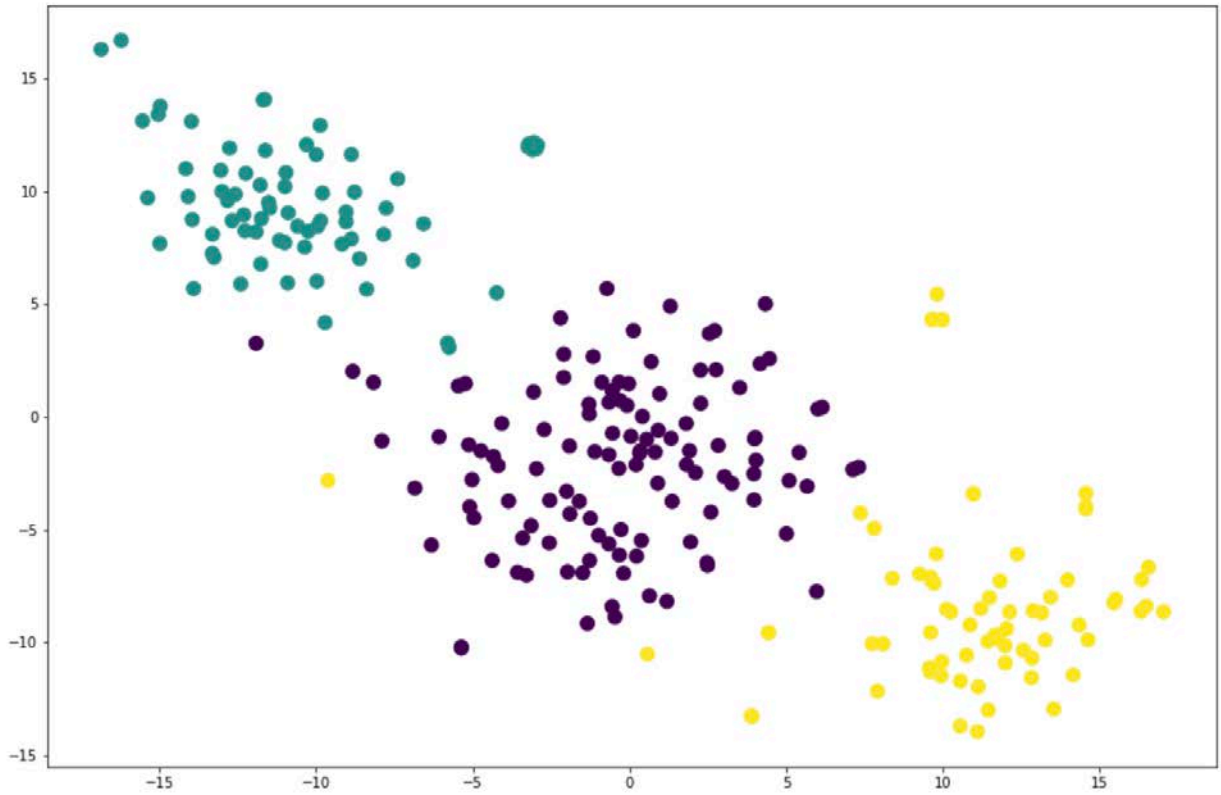

Рис. 16 – Визуализация кластеризации методом k-средних векторов cbow [Авторская разработка].

В завершении анализа была рассчитана точность – метрика, которая демонстрирует долю объектов, распознанных как объекты положительного класса - верно. Для cbow точность составила 96,9%, для skip-gram – 88,7%. Визуальный анализ кластеризации, продемонстрированной на Рис.15 и Рис.16, доказывает лучшей качество разделения по тематикам векторов cbow. Таким образом, наилучшей из двух рассмотренных моделей является – cbow, а значит именно ее приоритетно стоит использовать для векторизации.

В настоящей работе исследована возможность применения нейронных сетей с использованием алгоритма Word2Vec для целей проведения кластеризации статей, относящихся к разным тематикам. Практическая ценность данного исследования неоспорима, так как описанный подход в обработке может быть применен в различных сферах деятельности. Например, для автоматизированного формирования электронных библиотек, а также в системах предварительной проверки и оценки научных работ.

Библиографический список

1. Бенгфорт Б., Билбро Р., Охеда Т. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка – СПб.: Питер, 2019. – 368 с.: ил. – (Серия «Бестселлеры O'Relly»)

2. Грас Дж. Data Science. Наука о данных с нуля: Пер. с англ. – СПб.: БХВ-Петербург, 2017. – 336 с.: ил.

3. Интернет-ресурс N+1, Елизавета Ивтушок, «Векторное представление слов» [Электронный ресурс]. — Ресурс доступа — URL: https://nplus1.ru/news/2018/04/04/word-embedding (дата обращения 10.03.2020)

4. Моделировани и анализ информационных систем, Векторное представление слов с семантическими отношениями: экспериментальные наблюдения [Электронный ресурс]. — Ресурс доступа — URL: http://www.mathnet.ru/php/ archive.phtml?wshow=paper&jrnid=mais&paperid=659&option_lang=rus (дата обращения 14.03.2020)

5. Шолле Ф. Глубокое обучение на Python. – СПб.: Питер, 2018. – 400 с.: ил. – (Серия «Библиотека программиста»)

6. AI-news.ru [Электронный ресурс]. — Ресурс доступа — URL: https://ai-news.ru/2018/07/word2vec_kak_rabotat_s_vektornymi_predstavleniyami_slov.html (дата обращения 17.03.2020)

7. Kutuzov A., Andreev I., "Texts in, meaning out: neural language models in semantic similarity task for Russian", 2015 [Электронный ресурс]. — Ресурс доступа — URL: https://arxiv.org/abs/1504.08183 (дата обращения 13.01.2020)

8. Nature, Tshitoyan, V., Dagdelen, J., Weston, L. et al. Unsupervised word embeddings capture latent knowledge from materials science literature. Nature 571, 95–98 (2019) [Электронный ресурс]. — Ресурс доступа — URL: https://www.nature.com/articles/s41586-019-1335-8 (дата обращения 16.03.2020)

9. RusVectōrēs: семантические модели для русского языка [Электронный ресурс]. — Ресурс доступа — URL: https://rusvectores.org/ru/ (дата обращения 15.03.2020)

10. Wang C., Cao L., Zhou B., “Medical Synonym Extraction with Concept Space Models”, 2015 [Электронный ресурс]. — Ресурс доступа — URL: https://arxiv.org/pdf/1506.00528.pdf (дата обращения 15.03.2020)

Оригинальность 92%