МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра дискретной математики и информационных технологий

ВЕКТОРНОЕ ПРЕДСТАВЛЕНИЕ СЛОВ (WORD EMBEDDINGS)

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Кузьмина Романа Александровича

Научный руководитель
профессор, д. ф.-м. н. _____ В. А. Молчанов

Заведующий кафедрой
доцент, к. ф.-м. н. _____ Л. Б. Тяпаев

Саратов 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Идея векторного представления слов основана на контекстной близости слов. Каждое слово может быть представлено в виде вектора, близкие координаты векторов могут быть интерпретированы как близкие по смыслу слова. Таким образом, извлечение семантических отношений (отношение синонимии, родо-видовые отношения и другие) может быть автоматизировано. Установление семантических отношений вручную считается трудоемкой и необъективной задачей, требующей большого количества времени и привлечения экспертов. Но среди ассоциативных слов, сформированных с использованием одной из модели векторного представления слов, встречаются слова, не представляющие никаких отношений с главным словом, для которого был представлен ассоциативный ряд.

Векторное представление слов может применяться в таких прикладных задачах, как:

— Использование отзывов клиентов или ответов на опросы сотрудников, чтобы понять их отношение к конкретному продукту или компании.

— Использование текстов песен, которые слушает конкретный пользователь для рекомендации этому пользователю похожих песен

— Использование службы веб-перевода, такой как Google Translate, для перевода статей на веб-страницах на другой язык.

Целью данной работы является изучение основ векторного представления слов и последующее построение векторного пространства слов для корпуса текста с использованием приобретенных знаний. Все рассматриваемые примеры текстов будут на английском языке, т.к., во-первых, наборов данных на русском языке достаточно немного, во-вторых, в виду наличия особенностей языка, для русского языка необходимы специальные библиотеки и другие программные инструменты, которых практически нет на данный момент.

Задачи данной научной работы:

1. Изучить теоретические основы векторного представления слов.
2. Рассмотреть дистрибутивные векторные представления слов.
3. Подробно рассмотреть на практике два подхода модели Word2Vec.
4. Построить векторные представления слов для произведений Уильяма Шекспира.

1 Основная часть

В первом разделе выпускной квалификационной работы рассматриваются основные теоретические понятия, используемые в области обработки естественного языка и векторного представления слов. Основные понятия из этого раздела:

1. Векторное представление слов (word embedding) - представление слов для текстового анализа, которое обычно является некоторым вещественным вектором, который кодирует значение слова таким образом, что слова, которые имеют схожее по смыслу значение, находятся ближе друг к другу в векторном пространстве.
2. Векторизация - процесс конвертации текста в вектора.

Первый раздел состоит из трех подразделов.

Первый подраздел содержит описание такого метода векторного представления слов, как one-hot кодирование. Суть данного кодирования заключается в том, чтобы, в случае текстовых данных, взять вектор длины словаря и поставить только одну единицу в этом векторе в позиции, соответствующей порядковому номеру слова в словаре. Данный метод не обладает свойством семантической близости, однако, не смотря на это, он используется во многих других подходах, например, в подходе Word2Vec, который рассмотрен позднее в этой работе.

Второй подраздел описывает подход Bag of Words (BoW). Основная идея данного подхода заключается в том, что документы являются схожими, если имеют в своем составе схожие слова. В названии данного подхода фигурирует слово "мешок"(bag). Это связано с тем, что любая информация о порядке или структуре слов в документе никаким образом не проявляется в подходе Bag of Words. Далее в этом подразделе будет рассмотрена модель BoW на примере. Подраздел состоит из 4 подпунктов.

В первом подпункте описывается процедура сбора данных, которая заключается в нахождении документов из предметной области, которая будет изучаться с помощью подхода BoW.

Второй подпункт содержит описание процесса формирования словаря, который будет использоваться для формирования векторов. Процесс состоит из токенизации документов и последующем составлении массива из уникальных токенов.

Третий подпункт описывает процесс формирования векторов слов. В результате данного процесса получается двоичный вектор длины массива из уникальных токенов (словаря), где единица на соответствующей позиции ставится в том случае, если соответствующее слово из словаря присутствует в конкретном документе.

В четвертом подпункте описываются способы управления словарем. Рассматриваются такие техники по оптимизации словаря как:

1. Игнорирование регистра
2. Игнорирование пунктуации
3. Игнорирование часто встречающихся слов, которые не несут важной информации (предлоги, артикли, союзы)
4. Исправление неточностей написания слов
5. Лемматизация

В третьем подразделе рассматривается подход TF-IDF, который был заимствован из области информационного поиска. Суть этого подхода заключается в том, что вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции. В данном случае под коллекцией документов подразумевается набор рассматриваемых текстов.

Во втором разделе кратко рассматривается история развития подходов в области векторного представления слов от методик, заимствованных из области информационного поиска, до использования дистрибутивного закона.

В третьем разделе рассматриваются суть дистрибутивного закона и представлений слов, основанных на этом законе, а также некоторые типы дистрибутивных представлений. Подавляющая часть современных векторных представлений слов основаны на дистрибутивном законе, поэтому данная тема является чрезвычайно важной для рассмотрения. Суть этого закона заключается в том, чтобы понять смысл слова в контексте путем предположения зависимостей этого слова с остальными словами в тексте. Данный раздел состоит из 2 подразделов.

Первый подраздел описывает статические векторные представления слов - самый ранний тип векторного представления слов. Этот подход заключается в том, чтобы каждому слову в словаре сопоставить статический вектор (например, порядковый номер слова в словаре). У статических векторных пред-

ставлений слов есть несколько существенных недостатков, из-за чего этот подход практически не используется.

Во втором подразделе рассматривается подход Word2Vec - семейство архитектур моделей и оптимизаций. Данный подход оказался крайне эффективным в виду относительно небольшой ресурсоемкости и точности получаемых с помощью него векторных представлений слов. Подход Word2Vec был создан в 2013 году командой исследователей из Google во главе с Томашем Миколовым. Далее, в этом подразделе рассматриваются различные модели подхода Word2Vec. Подраздел состоит из 7 подпунктов

В первом подпункте рассматривается модель Word2Vec с моделью языка нейронной сети с прямой связью.

Во втором подпункте описывается модель Word2Vec с моделью языка нейронной сети с рекуррентной связью. В этих двух первых подпунктах рассматривается внутреннее устройство соответствующих нейронных сетей, их принцип работы и вычислительная сложность обучения.

Третий подпункт описывает в общем лог-линейные (log-linear) модели и их преимущества по сравнению с моделями, в которых используются классические нейронные сети.

Четвертый подпункт описывает модель Word2Vec, которая получила название Continuous Bag of Words. Суть этой модели заключается в том, что она предсказывает слово, находящееся в центре по тем словам, которые его окружают.

В пятом подпункте описывается модель Word2Vec, которая называется Skip-gram модель. Основная идея данной модели в том, чтобы предсказывать окружающие слова (или контекст) для рассматриваемого слова.

В шестом подпункте рассматривается иерархический подход вычисления функции softmax. Эта функция применяется к каждому слову в словаре в качестве функции активации на выходном слое. Суть данной функции заключается в том, что она преобразует числа, которые соответствуют векторному представлению слов в вероятность того, что данное слово является результатом. Применение функции softmax является самой ресурсозатратной операцией в подходах Word2Vec, иерархический же способ вычисления этой функции позволяет снизить время и количество ресурсов, необходимых для применения этой функции.

В седьмом подпункте также описывается подход более эффективного применения функции softmax, который получил название negative sampling. Его суть заключается в том, что он максимизирует вероятность встречи нужного слова в типичном контексте и в то же время минимизирует вероятность встречи этого слова в нетипичном контексте.

В четвертом разделе описывается конвейер обработки естественного языка. Суть этого подхода состоит в том, чтобы разбить комплексную задачу на небольшие части, которые легко решить. Далее рассмотрим на какие составные части рабивается задача "очистки"текста от неполезной информации. Раздел состоит из 7 подразделов.

В первом подразделе описывается процесс выделения предложений. В результате этого процесса исходный текст разбивается на отдельные предложения.

Второй подраздел описывает процесс токенизации - разбиение текста на более мелкие части, токены. К токенам относятся как слова, так и знаки пунктуации.

В третьем подразделе рассматривается процесс, в результате которого для каждого слова в предложении определяется какой частью речи является это слово.

В четвертом подразделе описывается процесс лемматизации. В результате этого процесса получаем леммы - нормальные (словарные) формы для каждого слова.

Пятый подраздел содержит описание процесса определения "стоп-слов". В английском языке достаточно много вспомогательных слов - различные союзы и артикли. При статистическом анализе текста такие вспомогательные слова создают много шума, так как появляются чаще, чем остальные. Большинство NLP-конвейеров отмечают такие слова как “стоп-слова” и отсеивают их перед тем, как продолжить обработку текста.

В шестом подразделе содержится описание процесса извлечения зависимостей из текста. Итогом этого шага является дерево, в котором каждый токен имеет единственного родителя. Также, помимо определения родителя, устанавливается тип связи между двумя словами.

В седьмом подразделе описывается процесс распознавания именованных сущностей. Именованные сущности - это персоны, локации, организации.

В пятом разделе содержится описание процесса применения подходов Skip-gram и negative sampling на практическом примере. В качестве примера предложения, на котором рассматриваются эти подходы, берется следующее предложение: "The wide road shimmered in the hot sun."В качестве инструментов, которые помогут применить на практике подходы Skip-gram и negative sampling используются язык программирования Python версии 3.6.9, библиотеки keras и tensorflow и среда разработки Google Colab. Этот раздел состоит из 7 подразделов.

В первом подразделе приведен фрагмент программы на языке Python, благодаря которому происходит настройка окружения - устанавливается значение необходимых глобальных переменных, подключаются необходимые библиотеки.

Второй подраздел содержит фрагмент программы, который векторизирует предложение и создает словарь, в котором ключом является слово из предложения, а соответствующим этому ключу значением является индекс слова в предложении.

В третьем подразделе происходит формирование skip-грам для каждого слова в предложении. В данном примере размер окна будет равен двум.

В четвертом подразделе выбираются отрицательные примеры для каждой skip-граммы, составленной на прошлом шаге. Т.е., другими словами, применяется подход negative sampling.

Пятый подраздел содержит описание процесса создания тренировочного примера для модели Word2Vec. В результате данного процесса происходит конкатенация skip-граммы и отрицательных примеров для нее, а также добавление т.н. меток.

В шестом подразделе описывается процесс создания таблицы выборки (sampling table). Функцию make_sampling_table из библиотеки keras можно использовать для создания таблицы вероятностной выборки на основе частотного ранга.

В седьмом подразделе происходит объединение всех предыдущих процедур из этого раздела в одну функцию, которую можно будет применять для любого предложения.

В шестом разделе данной работы описывается процесс подготовки тренировочных данных для модели Word2Vec. Зная, как работать с одним пред-

ложением для модели Word2Vec, основанной на отрицательной выборке, можно приступить к созданию обучающих примеров из большего списка предложений. Данный раздел состоит из 4 подразделов.

Первый подраздел содержит фрагмент программы, с помощью которого произойдет скачивание корпуса текста, на котором будет тренироваться модель Word2Vec. В качестве корпуса в данной работе будут использоваться произведения У. Шекспира.

Второй подраздел снова описывает процесс векторизации предложений, однако в данном случае используется функция из библиотеки keras для этой задачи в виду удобства ее использования. Для использования этой функции описывается специальная функция кастомизированной стандартизации текста. Стандартизация заключается в приведении текста в один регистр (нижний) и удалении знаков пунктуации.

В третьем подразделе рассматриваетсся процесс получения последовательностей из набора данных. Этот процесс необходим для того, чтобы подготовить набор данных для обучения модели Word2Vec. Данный процесс необходим, поскольку далее будет происходить перебор каждого предложения в наборе данных для получения положительных и отрицательных примеров.

В четвертом подразделе описывается процедура генерации тренировочных примеров из последовательностей, которые были получены в предыдущем подразделе. В данном подразделе используется функция, которая была определена ранее, чтобы сгенерировать обучающие примеры для модели Word2Vec.

Заключительный седьмой раздел описывает процесс обучения модели Word2Vec. Модель Word2Vec может быть реализована как классификатор, позволяющий отличать истинные контекстные слова с помощью skip-грамм и ложные контекстные слова, полученные с помощью отрицательной выборки(negative sampling). Можно выполнить скалярное произведение между векторными представлениями целевых(target words) и контекстных слов(context words), чтобы получить прогнозы для меток и вычислить потери относительно истинных меток в наборе данных. Данный раздел состоит из 3 подразделов.

В первом подразделе описывается модель Word2Vec с подклассами. С помощью подклассовой модели можно определить функцию call(), которая

принимает пары (target, context), которые затем могут быть переданы на соответствующий embedding уровень.

Второй подраздел содержит фрагмент программы, с помощью которого происходит построение модели Word2Vec, а также происходит обучение этой модели с помощью набора данных, который был подготовлен в предыдущих разделах.

В третьем подразделе происходит анализ полученных векторных представлений слов. Для этого используется инструмент Embedding Projector, который предоставляет проект Tensorflow. С помощью данного инструмента можно построить векторное пространство слов на основе векторных представлений этих слов. В результате можно наблюдать, что слова близкие по смыслу находятся в этом векторном пространстве ближе друг к другу, чем слова, смысл которых не похож.

ЗАКЛЮЧЕНИЕ

В данной работе были изучены теоретические основы векторного распределения слов (word embeddings), основные оставляющие данной области, такие как one-hot encoding, bag-of-words, TF-IDF. Также была рассмотрена история развития области векторного распределения слов, начиная с методик, которые пришли в данную сферу из науки, известной как информационный поиск, заканчивая современными методиками, в частности Word2Vec, которая была рассмотрена подробно с изучением математического аппарата этой методики.

В качестве практической составляющей работы было построено векторное пространство слов. В качестве корпуса для этой работы были взяты работы Уильяма Шекспира. Для этой задачи были использованы такие инструменты, как библиотека Tensorflow, библиотека Keras и среда исполнения кода Google Collab.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 A. Gulli, A. Kappor, S. Pal. Deep learning with TensorFlow 2 and Keras: Second edition. - Birmingham: Packt Publishing Ltd., 2019. - 657 p.

2 Yoav Goldberg. Neural Network Methods for Natural Language Processing. - Toronto: Morgan & Claypool, 2017. - 309 p.

3 Mohamed Abdelrahman Zahran Mohamed. New trends for building arabic language resources. - Giza: Cairo University, 2015. - 94 p.

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

5 K. Simov, P. Osenova, J. Hajic, A. Branco. The Workshop on Deep Language Processing for Quality Machine Translation (DeepLP4QMT) - Sofia: QTLeap, 2016. - 54 p.

6 G.E. Hinton, J.L. McClelland, D.E. Rumelhart. Distributed representations. In: Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations, MIT Press, 1986.

7 T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, S. Khudanpur. Extensions of recurrent neural network language model, In: Proceedings of ICASSP 2011.

8 T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning, 2007.

9 Andriy Mnih, Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. arXiv preprint arXiv:1206.6426, 2012

10 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.

11 Макмахан Б., Рао Д. Знакомство с PyTorch: глубокое обучение при обработке естественного языка. — СПб.: Питер, 2020. — 256 с.

12 Language Models are Unsupervised Multitask Learners [Электронный ресурс]. URL: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf (дата обращения: 11.05.2021).

13 Т. Рашид. Создаем нейронную сеть.:Пер. с англ. - Спб.: ООО "Альфа-книга 2017. - 272 с.: ил. - Парал. тит. англ.

14 Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606, 2016.

15 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013

16 Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. Lake Tahoe, Nevada, United States, 2013.

17 Вьюгин В. В. Математические основы машинного обучения и прогнозирования. - М.: МЦНМО, 2014. - 304 с.

18 A. Mnih, G. Hinton. A Scalable Hierarchical Distributed Language Model. Advances in Neural Information Processing Systems 21. - MIT Press, 2009.

19 R. Collobert and J. Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In International Conference on Machine Learning, ICML, 2008.

20 Gujarati, Damodar N.; Porter, Dawn C. (2009). "How to Measure Elasticity: The Log-Linear Model". Basic Econometrics. New York: McGraw-Hill/Irwin.

21 Rawlings, John O.; Pantula, Sastry G.; Dickey, David A., eds. (1998). "Applied Regression Analysis". Springer Texts in Statistics.