Исследование алгоритмов обработки текстовых данных в социальных сетях

Ю.А. Курбатов[1], И.А. Рыцарев[1], А.В. Куприянов[1,2]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086
[2]Институт систем обработки изображений РАН - филиал ФНИЦ «Кристаллография и фотоника» РАН, Молодогвардейская 151, Самара, Россия, 443001

Аннотация. В работе проводится исследование различных алгоритмов кластеризации большого объёма текстовых данных. Был проведён анализ существующих способов реализации и выбраны алгоритмы Word2Vec и GloVe. Исходные текстовые данные для тестирования алгоритмов были получены путём сбора записей из открытых сообществ ВКонтакте. Полученные результаты показали, что, применение данных алгоритмов позволяет оценить частоту употребления и значимость отдельных слов относительно контекста исследуемого сообщества. Также в работе было произведено сравнение результатов применения алгоритмов и сделан вывод об их эффективности.

1. Введение

В настоящее время постоянно генерируется огромное количество данных, такие как банковские операции, данные мобильных операторов, телеметрия и т.д. Особый интерес вызывают социальные сети, так как они представляют собой разнообразную, непрерывно обновляемую информацию огромного размера. Многим компаниям необходимо анализировать данные, полученные из социальных сетей, для оценки отношения пользователей к своим продуктам. Кроме этого, анализ данной области используется в решении вопросов безопасности. Собрав и кластеризировав текстовые данные из социальной сети, можно определить основные темы и события, обсуждаемые пользователями социальных сетей в различных городах и странах.

2. Word2Vec

Кластеризация позволяет использовать общие атрибуты различных классификаций в целях выявления кластеров. Исследуя один или более атрибутов, или классов, можно сгруппировать отдельные элементы данных вместе, получая структурированное заключение [1].

При реализации задач кластеризации используются различные алгоритмы. Одним из широко используемых является Word2Vec. Идея алгоритма в сравнении не самих слов или составленных из них последовательностей (так называемых н-грамм), а семантических классов, в которые они попадают [2].

Для обучения был взят большой объем текстового материала, полученного из социальной сети «Вконтакте» путем скачивания данных через API(Модель реализована на Python с использованием библиотек NumPy, SciPy.[3]). Алгоритм каждому слову сопоставляет вектор. Причем близкие слова соответствуют близким векторам. Мерой близости слов выступает их контекстная близость: близкие слова встречаются в тексте рядом с одинаковыми словами. А расстоянием между векторами измеряется при помощи косинусного сходства (cosine similarity) [4-5]. Косинусное сходство между векторами A и B вычисляется по формуле:

$$similarity = \frac{(A,B)}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1} B_i^2}}$$

Обучая нейронные сети, Word2Vec максимизирует косинусную меру близости между векторами слов, которые встречаются в похожих контекстах и минимизирует косинусную меру между словами, которые не встречаются рядом. На выход Word2Vec передает координаты векторов, соответствующих данным словам.

В Word2Vec можно использовать две различных архитектуры нейронной сети с помощью которой осуществляется перевод слова в вектор: Continuous Bag of Words (CBOW) и Skip-gram. Выбор одной из этих моделей выполняется с помощью задания гиперпараметра ‘sg’. По умолчанию sg=0, используется модель CBOW. Если sg=1, используется Skip-gram[6-7].

Другим гиперпараметром является размер окна, в котором рассматривается контекст данного слова. В данной реализации используется параметр ‘window’, который определяет максимальное количество слов между данным словом и соседним внутри предложения, слова стоящие от данного дальше не будут рассматриваться как его контекст. Притом какое слово стоит в тексте ближе, а какое дальше от данного слова, не учитывается, если оба слова попали в окно.

Еще один важный гиперпараметр ‘size’ - размерность векторов, соответствующих словам. Если его величина мала, то модель получается грубой, но при большом значении роль машинного обучения теряется, и сопоставление словам векторов может превратиться в унитарное кодирование слов (one-hot encoding).

Для более детального анализа лучше всего сочетать различные подходы и методы в зависимости от количества обрабатываемых данных.

3. GloVe

GloVe – это еще один популярный алгоритм машинного обучения без учителя для получения векторных представлений слов. Принцип работы GloVe очень схож с Word2Vec, но в отличии от Word2Vec, у которого в основе лежит модель «предсказания», GloVe построен на основе модели «вычисления» [8].

GloVe стремится решить проблему захватом значения одного слова со структурой всего обозримого корпуса. Сначала он обходит весь корпус и собирают статистику появления слов, после чего составляет матрицу совместного появления слов. Матрица совместного появления слов – это матрица, каждая строка и столбец которой соответствуют какому-то слову из корпуса, а на пересечениях строк и столбцов стоят числа, соответствующие тому, сколько раз слово в строке стояло рядом со словом в столбце. Расстояние, на котором должны отстоять друг от друга слова определяется параметром алгоритма. [9].

Далее происходит факторизация матрицы. Схожая операция происходит также внутри алгоритма Word2Vec и называется негативным семплированием. В результате, от GloVe требуется минимизировать следующий функционал:

$$J = \Sigma_{i=1}^{V}\Sigma_{j=1}^{V} f\left(X_{ij}\right)_{l}\omega_i^T\omega_j + b_i^+ b_J - \log(X_{ij}))^2,$$

где V – величина словаря; ωi – вектор главного слова; ωj – вектор контекстного слова; bi, bj – скалярные смещения; $f(Xij)$ – взвешенная функция, которая предотвращает переобучения на часто повторяющихся парах.

Отличительной особенностью GloVe является еще и то, что этот алгоритм может учитывать различия между словами, которые достаточно близко лежат на векторной плоскости, но все-таки имеют определенные различия, например, пара брат-сестра. Для этого в GloVe

реализована математическая модель, которая устанавливает примерно одинаковые векторные расстояния между такими словами.

4. Результаты

Для сбора данных в целях анализа эффективности разработанных алгоритмов были отобраны сообщества схожих сфер интересов. Мы выбрали ВКонтакте для анализа сообществ в социальной сети, так как она является одной из самых популярных русскоязычных в интернете. Особенностью социальной сети является то, что она доступна всем и свободно предоставляет API для написания внешних приложений. Для получения данных из социальной сети было создано приложение на сервере ВКонтакте и получены ключи доступа. Следующим этапом нашей работы была разработка программного модуля для сбора данных. Реализация была осуществлена в Python с использованием скриптовой библиотеки для ВКонтакте. Все взаимодействие с социальной сетью затем осуществляется через модуль.

После сбора записей выбранных сообществ на совокупности всех текстов обучили Word2Vec, реализованный в библиотеке gensim. Для анализа данных взяли матрицу с шириной окна $5*2 = 10$, а размерностью векторов 100. Для каждого слова из текста был получен соответствующий ему вектор и рассчитано конусное сходство между векторами. Пример векторных расстояний, полученных при применении World2Vec представлен в таблице 1.

Таблица. 1. Векторные расстояния между словом «Музыка» и другими словами по мере word2vec.

Слово	Векторное расстояние
Для	0,985627
Онлайн	0,751461
Популярная	0,721355
Танцевальная	0,691231
Новинки	0,678548

Параллельно к тем же данным был применен алгоритм GloVe. Реализация алгоритма включает в себя создание матрицы векторных представлений формой (5,100), где 5 – это максимальное число слов, а 100 – размерность представления, каждый элемент *i* которой содержит вектор с размером размерности представления, соответствующий слову с номером *i* в индексе, созданном в ходе токенизации. На рисунке 1 приведен пример векторных расстояний, полученных при применении GloVe.

Рисунок 1. Векторные расстояния между словами в модели GloVe.

В процессе работы для сравнения результатов обработки данных ранее описанными алгоритмами было взято 3 основные метрики: точность результата анализа, скорость обработки данных и объем используемой оперативной памяти (рис. 2).

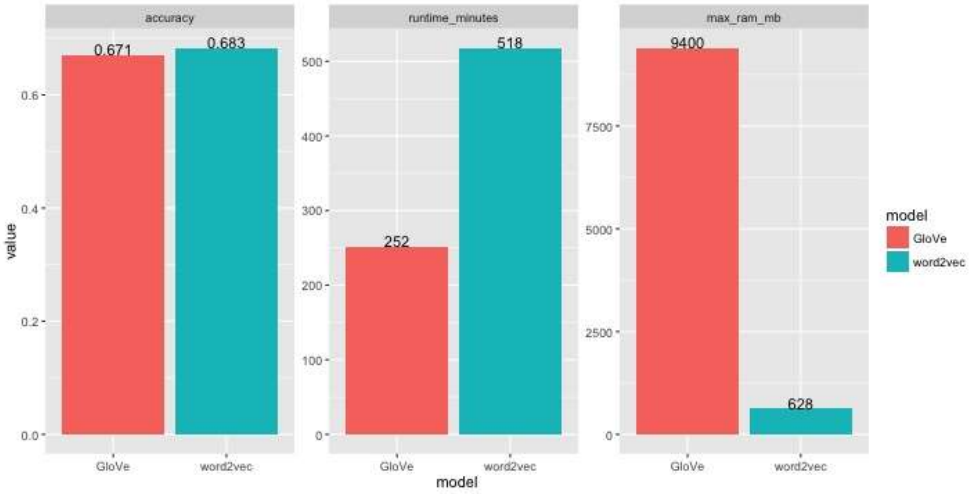

Рисунок 2. Сравнительный анализ результатов применения алгоритмов Word2Vec и Glove.

Таким образом, оба алгоритма обеспечивают один и тот же основной вывод: вектор на слово, с векторами в полезном расположении - с относительными расстояниями/направлениями, которые примерно соответствуют нашим представлениям об общей взаимосвязанности слов и даже связанности по определенным семантическим измерениям.

Работая из одного корпуса, создавая векторы слов одинаковой размерности и уделяя то же внимание мета-оптимизации, качество их результирующих векторов слов будет примерно одинаковым. Существенные различия лишь в скорости обработки данным и в объеме используемой оперативной памяти.

По сути, где GloVe предварительно вычисляет в памяти большую матрицу совместного использования слов, а затем быстро ее факторизирует, word2vec просматривает предложения в режиме онлайн, обрабатывая каждое совместное вхождение отдельно. Таким образом, существует компромисс между использованием большего количества памяти (GloVe) и более продолжительным обучением (word2vec). Кроме того, после вычисления GloVe может повторно использовать матрицу совместного вхождения для быстрой факторизации с любой размерностью, в то время как word2vec нужно обучать с нуля после изменения его размерности вложения.

5. Заключение

Вопросы, связанные с кластеризацией и дальнейшей классификацией текстовых данных, являются актуальными в связи с колоссальным распространением социальных сетей и интернет сервисов во всем мире. Подходы и методы, представленные в статье, планируются к апробации над текстовыми данными, собираемыми из социальной сети Вконтакте в российском сегменте. Сбор необходимых данных ведется при помощи разработанного программного комплекса. Планируется развить данную тему в направлении вывода и оптимизации параллельных алгоритмов кластеризации.

6. Благодарности

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (№ 18-37-00418, № 19-29-01135, № 19-31-90160) и Министерства науки и высшего образования Российской Федерации в рамках выполнения государственного задания Самарского университета и ФНИЦ «Кристаллография и фотоника» РАН.

7. Литература

[1] Коваленко, Т.В. Разработка библиотеки построения векторной модели текста на основе морфемного разбора слов / Т.В. Коваленко, Р.Б. Галинский, Ю.В. Яковлева, И.В.

Никифоров // Неделя науки СПбПУ: материалы научной конференции с международным участием – СПб.: Изд-во Политехн. ун-та, 2017.
[2] Vector Representations of Words [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/tutorials/word2vec (21.11.2019).
[3] Краткий обзор языка Python [Электронный ресурс]. – Режим доступа: http://www.helloworld.ru/texts/comp/lang/python/python2/index.htm.
[4] Mikolov T. Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean // Proceedings of NIPS, 2013.
[5] Efficient Estimation of Word Representations in Vector Space / T. Mikolov, K. Chen, G. Corrado, J. Dean // Proceedings of Workshop at ICLR, 2013.
[6] Rytsarev, I.A. Application of the principal component analysis to detect semantic differences during the content analysis of social network / I.A. Rytsarev, D.D. Kozlov, N.S. Kravtsova, A.V. Kupriyanov, K.S. Liseckiy, S.K. Liseckiy, R.A. Paringer, N.Yu. Samykina // CEUR Workshop Proceedings. – 2018. – Vol. 2212. – P. 262-269.
[7] Рыцарев, И.А. Классификация текстовых данных социальной сети Twitter / И.А. Рыцарев, А.В. Благов // Сборник трудов «Информационные технологии и нанотехнологии» (ИТНТ) – Самара: Новая техника, 2016. – С. 1073-1076.
[8] Global Vectors for Word Representation [Electronic resource]. – Access mode: https://nlp.stanford.edu/projects/glove/(15.11.2019)..
[9] Suzuki, J. A Unified Learning Framework of Skip-Grams and Global Vectors / J. Suzuki, M. Nagata // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing – Beijing, China, 2015. – P. 186-191.

Research of text data processing algorithms in social networks

Y.A. Kurbatov[1], I.A. Rytsarev[1], A.V. Kupriyanov[1,2]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086
[2]Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

Abstract. In this paper we investigate various clustering algorithms for a large amount of text data. An analysis of the existing implementation methods was carried out and the algorithms Word2Vec and GloVe were selected. The initial textual data for testing the algorithms were obtained by collecting records from open VKontakte communities. The results showed that the use of these algorithms allows us to assess the frequency of use and the significance of individual words relative to the context of the studied community. The results of the algorithms`applications were compared and the conclusion about their efficiency was made in the work as well.