УДК 004.6/.85

ИСПОЛЬЗОВАНИЕ МОДЕЛИ WORD2VEC ДЛЯ КЛАСТЕРИЗАЦИИ БОЛЬШИХ ТЕКСТОВЫХ ДАННЫХ

Черепков Е.А., Глебов С.А.

Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Калуга, e-mail: e.cherepkov@ya.ru

В данной статье приведен вариант использования модели Word2Vec и алгоритма k-means для кластеризации большого количества текстовой информации. Рассмотрены основные этапы работы модели: подготовка данных, генерация векторов слов и кластеризация слов. На этапе генерации векторов слов для обучения модели используется набор данных 20 Newsgroups data set. Для генерации используется модель Skip-gram. Для кластеризации используется алгоритм K-means, который разбивает N элементов на K-кластеров. Далее оценивается производительность модели и эффективность классификации, при помощи показателя F1-micro и времени работы алгоритма. На основе полученных результатов был сделан вывод о том, что модель Word2Vec успешно справляется с поиском семантически связанных слов и в результате получаются правильные кластеры.

Ключевые слова: большие данные, Word2Vec, кластеризация, k-means

USING WORD2VEC MODEL FOR CLUSTERING BIG TEXT DATA

Cherepkov E.A., Glebov S.A.

Kaluga branch of the Federal State Budget Education Institution of Higher Education «Moscow State Technical University named after N.E. Bauman (National Research University)», Kaluga, e-mail: e.cherepkov@ya.ru

This article describes the use Word2Vec model and k-means algorithm for clustering large amount of textual information. Describes the main stages of the model: data preparation, generation of words vectors and clustering words. At the stage of generation of word vectors for learning the model, the data set 20 Newsgroups data set is used. To generate the model is used Skip-gram. For clustering, the K-means algorithm is used, which splits N elements into K-clusters. Further, the model performance and classification efficiency are estimated using the F1-micro indicator and the algorithm running time. Based on the results obtained, it was concluded that the Word2Vec model successfully copes with the search for semantically related words, and as a result, the right clusters are obtained.

Keywords: big data, Word2Vec, clustering, k-means

Большие данные [1] – это огромный набор данных со сложной структурой [2]. Такие данные встречаются в различных областях: биология, медицина, социальные сети, аналитика продаж и гуманитарные науки [3]. Наличие в системе большого количества данных накладывает ограничения на применяемые технологии хранения [4] и обработки данных. Обработка большого набора данных является сложной задачей для традиционных инструментов. Поэтому были созданы новые методы, как альтернатива традиционным СУБД и методам Business Intelligence. Сюда можно включить такие средства массово-параллельной обработки неопределенно структурированных данных, как NoSQL решения, библиотеки проекта Hadoop, а также алгоритмы MapReduce. Для анализа больших данных применяются технологии искусственного интеллекта и алгоритмы машинного обучения [5]. Чтобы сделать алгоритм обучения более эффективным и устранить риск переобучения, необходим способ уменьшить размер данных.

Модель Word2vec [6], созданная Google, представляет собой нейронную сеть, которая обрабатывает текстовые данные. Word2Vec – это не один алгоритм, он включает в себя две модели обучения: «Continuous Bag of Words» (CBOW) и Skip-gram [7, 8]. CBOW – «непрерывный мешок со словами», архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста. Архитектура типа Skip-gram действует иначе: она использует текущее слово, чтобы предугадывать окружающие его слова. Имеется возможность переключаться и делать выбор между алгоритмами. Порядок слов контекста не оказывает влияния на результат ни в одном из этих алгоритмов. На вход в обучающую модель Word2Vec подается текстовый массив данных, а на выходе генерируются векторы слов. Кроме того, Word2Vec имеет возможность вычислять косинусное расстояние между каждым словом. Поэтому выгодно группировать похожие слова на основе их расстояний. Сгруппировав подобные слова, исходный размер данных проецируется на новое, более низкое измерение. На первом этапе в модель Word2Vec подается крупномасштабный текстовый массив данных для создания векторов слов. На следующем эта-

пе для кластеризации этих векторов слов используется алгоритм машинного обучения (K-means) [9], где слова (признаки) группируются в K различных кластеров. Алгоритм k-means разбивает исходное множество объектов на k кластеров так, что средние в кластере (для всех переменных) максимально возможно отличаются друг от друга.

Выбор числа k может основываться на предыдущих результатах, теоретических основах или интуиции. Если нет предположений относительно этого числа, рекомендуют начинать с меньшего количества, постепенно увеличивая его и сравнивая полученные результаты.

Работа алгоритма состоит из нескольких этапов:

- случайным образом выбрать k точек, являющихся начальными «центрами масс» кластеров;
- отнести каждый объект к кластеру с ближайшим «центром масс»;
- пересчитать «центры масс» кластеров согласно их текущему составу;
- если критерий остановки алгоритма не удовлетворен, вернуться к п. 2.

В качестве критерия остановки работы алгоритма обычно выбирают одно из двух условий:

- «центры масс» кластеров стабилизировались, т.е. все наблюдения принадлежат кластеру, которому принадлежали до текущей итерации;
- число итераций равно максимальному числу итераций.

Генерация вектора слов

Для обучения используется популярный набор данных 20 Newsgroups data set [10]. Данные «The 20 Newsgroups» – это коллекция примерно из 20000 новостных документов, разделенная (приблизительно) равномерно между 20 различными категориями. Изначально она собиралась Кеном Ленгом (Ken Lang) для его работы «Newsweeder: Learning to filter netnews» («Новостной обозреватель: учимся фильтровать новости из сети»), хотя он явно не заявлял об этом. Коллекция «The 20 newsgroups» стала популярным набором данных для экспериментов с техниками машинного обучения для текстовых приложений, таких как классификация текста или его кластеризация. Данные разделены на 20 групп, и каждая текстовая группа соответствует разным доменам, таким как sci.med и rec.autos. Для оценки модели используется 11314 обучающих данных. Прежде чем подавать обучающие данные в Word2Vec, необходимо их подготовить. Для работы необходимо обучить нейронную сеть. Набор учебных данных для нейросети Word2vec конструируем следующим образом.

1. Очищаем входной текст T от лишних символов (знаки препинания и т.п.).

2. Из очищенного текста T собираем словарь W.

3. Для каждого слова $w_i \in T$ собираем контекст, т.е. набор слов $C_i \subset T$, удалённых от w_i не более чем на s позиций в последовательности слов T. $C_i = \{w_j \in T:(i - s) \leq j \leq (i + s), j \neq i\}$ иначе говоря – C_i это слова $w_j \in T$ из s-окрестности слова $w_i \in T$.

4. Выполняем унитарное кодирование (one-hot encoding) словаря W, т.е. каждому слову $w_i \in W$ ставится в соответствие вектор $u_i \in U$ из нулей и одной единицы, длина вектора u_i равна размеру словаря W, позиция единицы в векторе u_i соответствует номеру слова в словаре W.

5. Заменяем слова в тексте T и контекстах C соответствующими кодами P и Q из U.

Таким образом, получаем два множества – кодированный текст P и наборы кодированных слов контекста Q.

Чтобы хорошо сгенерировать векторы слов, мы используем модель Skip-gram, потому что она обладает лучшей способностью к обучению, чем CBOW, если не учитывать скорость вычислений [6].

На вход модели skip-граммы подается одно слово w_i, а на выходе мы получаем слова w_i в контексте $\{w_{O,1}, \ldots, w_{O,c}\}$, определяемом размером окна слов. После обучения каждому слову соответствует вектор. Наконец, строится матрица большого размера. Каждая строка в матрице представляет каждый пример обучения, а столбцы – сгенерированные векторы слов. Как следствие, слово имеет несколько степеней подобия. Чтобы проверить сходство слов, рассмотрим несколько примеров в табл. 1, в ней приведены два наиболее похожих слова и их косинусные расстояния. Например, с учетом слова «computer» мы получаем два похожих слова «evaluation» и «algorithm» с их расстояниями до «computer». Результаты показывают, что Word2Vec успешно находит семантически связанные слова.

Кластеризация слов

Необходимо найти центры каждого кластера слов. В настоящее время некоторые алгоритмы кластеризации могут выполнять эту работу [11]. Учитывая простоту, применяем алгоритм кластеризации K-means. Кластеризация K-means – это метод векторного квантования [12], который разбивает N элементов на K кластеров, минимизируя общую ошибку. Мерой расстояния в данном алгоритме является евклидово расстояние [13]. Единственным параметром, который нужно

установить, является количество кластеров K. В табл. 2 показано несколько примеров того, что содержат 4 кластера при K равном 500. Мы можем найти похожие слова, которые почти правильно кластеризованы.

Многоклассовая классификация

Чтобы оценить производительность модели [14], данные вписываются в многоклассовую классификационную задачу. Во-первых, преобразуем многоклассовую классификацию в несколько двоичных классификаций. Для этого применяется метод One-vs-Rest [13], на выходе которого можно получить единичный классификатор для каждого класса. Экземпляр одного класса рассматривается как положительный класс; Другие считаются отрицательными. Затем применяется LinearSVC (Linear Support Vector Classifier), на вход которой подаются результаты всех единичных классификаторов, а на выходе определенный моделью класс.

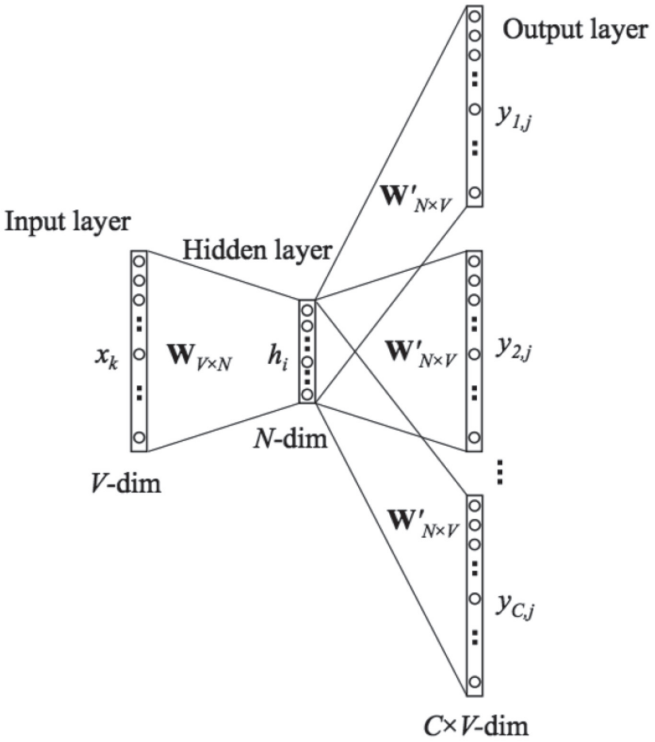

Диаграмма модели Skip-gram

Таблица 1

Сходство слов

Слова	Сходство слов			
	1-е слово	Косинусное расстояние	2-е слово	Косинусное расстояние
computer	evaluation	0,9325	algorithm	0,9292
president	property	0,9306	population	0,9123
american	Churches	0,9438	Greece	0,9365

Таблица 2

Содержание кластеров

Номер кластера	Содержимое кластера
1	u’my’, u’has’, u’is’, u’in’, u’the’, u’had’, u’for’…
2	u’components’, u’speed’, u’trigger’, u’applications’, u’stage’, u’developers’, u’manufacturers …
3	u’Medical, u’cigarettes’, u’usma’, u’smoked’, u’food’, u’Laboratory’, u’chewing…
4	u’population’, u’federal’, u’laws’, u’treasure’, u’crime, u’organizations’, u’Pope’…

Таблица 3

Производительность классификации

Размер данных	Производительность классификации	
	F1-Micro	Время (s)
11,314 * 61,189	0,7524	5 * 10^3
11,314 * 2000	0,774	8 *10^2
11,314 * 1500	0,7619	6 * 10^2
11,314 * 1000	0,753	4 * 10^2
11,314 * 500	0,7506	3 * 10^2

Производительность

Чтобы оценить эффективность классификации, вычисляется показатель F1-micro [15] и фиксируется время работы. В табл. 3 показана эффективность классификации для наборов данных с уменьшением размеров. В первой строке показано значение F1-micro и время, затрачиваемое без применения Word2Vec и алгоритма кластеризации. Остальные строки представляют производительность классификации данных с уменьшением размеров. Из полученных данных очевидно, что чем исходный набор данных больше, тем результаты лучше.

Выводы

В работе рассмотрен пример применения Word2Vec к обработке больших данных. Вначале обучающие данные подаются на вход в Word2Vec для генерации векторов слов. Для каждого слова рассчитывается вектор, с помощью которого находятся семантически связанные слова. Далее подобные слова группируются, используя алгоритм кластеризации K-means. Задавая значение *K*, определяется количество кластеров. Был проведен ряд экспериментов с целью определения эффективности предложенной модели. Исходя из результатов проведенных экспериментов, можно судить о достаточно высокой производительности данной модели. Также из экспериментов видно, что чем больше объем исходных данных, тем точность алгоритма выше.

Список литературы

1. Mayer-Schönberger V. Big data: A revolution that will transform how we live, work, and think / V. Mayer-Schönberger, K. Cukier. – Houghton Mifflin Harcourt, 2013. – 242 p.

2. Аксютина Е.М., Белов Ю.С., Обзор архитектур и методов машинного обучения для анализа больших данных // Электронный журнал: наука, техника и образование. – 2016. – № 1 (5). – С. 134–141.

3. Аксютина Е.М., Белов Ю.С. Анализ оптимизированного алгоритма выравнивания биологических последовательностей // Электронный журнал: наука, техника и образование. – 2017. – № 2 (12). – С. 100–106.

4. Васильчикова А.В., Ткаченко А.В., Гришунов С.С. Обзор классов нереляционных баз данных // Электронный журнал: наука, техника и образование. – 2016. – № 4 (9). – С. 81–85.

5. Нифонтов С.В., Белов Ю.С. Применение вейвлет-анализа и нейронных сетей в системах распознавания дикторов // Информационные технологии в науке нового времени: сборник статей Международной научно-практической конференции. – 2016. – С. 41–45.

6. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // Proc. Workshop at ICLR. – 2013.

7. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J., Distributed Representations of Words and Phrases and their Compositionality // Proc. NIPS. – 2013.

8. Mikolov T., Yih W., Zweig G., Linguistic Regularities in Continuous Space Word Representations // Proc. NAACL HLT. – 2013.

9. Wu J. Advances in K-means Clustering: A Data Mining Thinking (Springer Theses: Recognizing Outstanding Ph.D. Research) / J. Wu. – Springer. – 2012. – 180 p.

10. Lang K., 20 newsgroup data set [Электронный ресурс]. – URL: http://qwone.com/~jason/20Newsgroups/ (дата обращения: 17.10.17).

11. Гришанов К.М., Белов Ю.С. Метод классификации k – nn и его применение в распознавании символов // Фундаментальные проблемы науки: сборник статей Международной научно-практической конференции. – 2016. – С. 30–33.

12. Allen G. Vector quantization and signal compression / G. Allen, R.M. Gray. – Springer Science & Business Media. – 2012. – vol. 159. – 732 p.

13. Pedregosa F. Scikit-learn: Machine Learning in Python // JMLR 12. – 2011. – P. 2825–2830.

14. Long Ma, Yanqing Zhang. Using Word2Vec to process big text data // Big Data (Big Data), IEEE International Conference. – 2015.

15. Sebastiani F. Machine learning in automated text categorization // ACM Computing Surveys. – 2002. – № 34 (1). – P. 1–47.