Использование метода определения схожести слов для оценки алгоритмов векторизации текста

А.А. Сайгин, С.А. Федосин

Мордовский государственный университет им. Н.П. Огарёва, Саранск, Россия

Аннотация: В статье представлено краткое описание существующих методов векторизации текстов на естественном языке. Описывается оценка методом определения похожести слов. Проводится сравнительный анализ работы нескольких моделей векторизаторов. Описывается процесс подбора данных для оценки. Сравниваются результаты оценки работы моделей.
Ключевые слова: обработка естественного языка, векторизация, словоформенный эмбеддинг, семантическая близость, корреляция.

За последние несколько лет количество задач, связанных с обработкой естественного языка, сильно возросло. К данной категории задач относятся распознавание голоса, системы вопрос-ответ, классификация и кластеризация текстов, извлечение сущностей, машинный перевод [1].

Основная проблема при обработке текстов связана с тем, что вычислительная техника может оперировать только числовыми данными. Поэтому, для удобства дальнейшей обработки текст отображается в векторное пространство. При этом, итоговый вектор должен учитывать, что слова могут употребляться с разным значением в зависимости от контекста.

На данный момент существует множество алгоритмов, предназначенных для векторизации текстов. Результатом применения векторизующего алгоритма является словоформенный эмбеддинг [2]. Различия в работе каждого метода позволяют применять их для решения разных задач.

Оценка работы векторизаторов обычно происходит в рамках решаемой задачи. Но в таком случае оценивается непосредственно работа конкретной системы (например, классификатора), а не конкретно алгоритма векторизации. Из-за этого получается наложение ошибок векторизации и

модели-решателя. Для оценки работы векторизатора в [3-5] предлагается использовать метод определения семантической близости.

Данный метод применяется следующим образом. Используется набор данных формата «текст_1 текст_2 похожесть». Тексты прогоняются через оцениваемый векторизатор. Между парами векторов, полученных в результате, находится расстояние, которое будет характеризовать похожесть текстов. Данное утверждение основывается на том, что если содержание двух текстов близко, то и их вектора будут находиться на близком расстоянии в пространстве. Далее вычисляется значение коэффициента корреляции между полученными значениями похожести и оценками экспертов, который будет показателем качества работы модели.

Метод оценки схожести валиден, поскольку для близких по смыслу слов (синонимов, гипонимов) он даст высокие результаты в семантическом отношении, а для далёких и противоположных по смыслу – близкие к нулю. Кроме того, семантическое сходство лежит в основе большей части задач обработки естественного языка.

Рассмотрим некоторые популярные методы векторизации текстов. Возьмём предобученные модели следующих алгоритмов:

– word2vec (модель ruwikiruscorpora_upos_cbow_300_10_2021) [6];

– GloVe (модель navec_hudlit_v1_12B_500K_300d_100q) [7];

– FastText (модель cc.en.300) [8];

– USE (модель google/universal-sentence-encoder) [9];

– ELMo (модель ruwikiruscorpora_tokens_elmo_1024_2019) [6];

– BERT (модель google-bert/bert-base-multilingual-cased) [10];

Стоит отметить особенности работы перечисленных алгоритмов. На выходе моделей word2vec, GloVe и ELMo получается матрица размерностью n x m, где n – количество слов в последовательности, m – размерность выхода модели. Для получения одномерного вектора столбцы в матрице необходимо

суммаризовать. То же самое касатеся модели BERT, с тем отличием, что алгоритм использует собственный токенизатор, из-за чего количество строк n равно количеству токенов, на который был разбит текст. Так же генерируется отдельный вектор с меткой [CLS], который также можно использовать при оценке. Алгоритмы FastText и USE всегда на выходе выдают одномерный вектор.

Суммаризировать столбцы можно, либо найдя сумму элементов в столбце (обозначим такую суммаризацию как sum), либо их среднее значение (mean), либо максимальный элемент (max).

Для тестирования алгоритмов векторизации методом определения схожести слов уже существует ряд датасетов. На русском языке самым большим набором данных является Human Judgements of Word Pairs от сообщества Russian Semantic Similarity Evaluation (RUSSE). Он представляет из себя объединение наиболее популярных англоязычных датасетов, переведённых на русский язык. В нём содержится 398 пар слов с оценками их похожести [11]. Пример содержимого датасета представлен в таблице № 1.

Таблица № 1

Содержимое набора данных RUSSE

Слово 1	**Слово 2**	**Похожесть**
автомобиль	машина	0.958333
маг	волшебник	0.958333
доллар	бакс	0.952381
мальчик	парень	0.952381
кладбище	погост	0.916667

Для измерения расстояния используем косинусное расстояние, Евклидово расстояние и Манхэттенское расстояние. Для получения оценки

качества работы модели используем коэффициенты корреляции Пирсона и Спирмана.

Результаты оценки работы моделей представлены в таблице № 2.

Таблица № 2

Оценки работы моделей (набор данных RUSSE)

Модель	**Расстояние**	**Корреляция**	
		Спирмена	**Пирсона**
1	2	3	4
Word2vec	cosine	0,572713	0,486117
	manhattan	0,441845	0,4057
	Euclidean	0,44212	0,407079
GloVe	cosine	0,704874	0,655224
	manhattan	0,639388	0,609942
	Euclidean	0,64121	0,410588
FastText	cosine	0,055477	0,0725197
	manhattan	0,06448035	0,0924884
	Euclidean	0,055477	0,0890618
USE	cosine	-0,052675	-0,0221543
	manhattan	-0,051252	0,0118752
	Euclidean	-0,052675	0,0104548
ELMo	cosine	0,637945	0,642051
	manhattan	0,525595	0,541645
	Euclidean	0,527238	0,542009
BERT mean	cosine	0,253176	0,283375
	manhattan	0,149414	0,202141
	Euclidean	0,14498	0,200673
BERT max	cosine	0,253176	0,283375
	manhattan	0,149414	0,202141
	Euclidean	0,14498	0,200673

1	2	3	4
BERT sum	cosine	0,247441	0,274571
	manhattan	0,034752	0,0472385
	Euclidean	0,0379105	0,0459377
BERT cls	cosine	0,300889	0,31965
	manhattan	0,298956	0,350305
	Euclidean	0,292281	0,345443

Самую высокую оценку получила модель алгоритма GloVe, самую низкую – FastText и USE. Результат интересен тем, что решения, которые являются более новыми и продвинутыми за счёт способности распознавать контекст использования слов (USE, ELMo, BERT) показывают результат хуже, чем более старые решения (word2vec, GloVe). Связано это скорее всего с тем, что в наборе данных содержатся одиночные слова. Старые алгоритмы принимают на вход слово и выдают соответствующий им вектор. За счёт этого текст обрабатывается корректнее, особе учитывая, что новые модели не понимают контекст слова.

Для дополнительной проверки можно использовать набор данных, содержащий пары предложений. Подобный датасет не был найден, поэтому он был собран самостоятельно. Принцип составления был примерно такой же, как и у данных RUSSE. Был составлен набор предложений, часть которых относились к области программирования, а другая часть – к области художественной литературы. Их них были составлены пары, после чего был произведён опрос того, насколько предложения в парах похожи. Возможными ответами были варианты «Высокое сходство», «Умеренное сходство», «Слабое сходство» и «Совсем не похожи». На основании ответов в опросе были сформированы оценки. Всего получилось 67 пар предложений. Пример содержимого датасета представлен в таблице № 3.

Таблица № 3

Содержимое собственного набора данных

Предложение 1	**Предложение 2**	**Похожесть**
Голова выразительно посмотрела на Лену, потом на кран, от которого шла трубка к горлу головы, и два раза подняла брови.	Профессор пристально посмотрел на Лену, затем на аппарат, от которого к горлу головы шла трубка, и дважды поднял брови.	0.8667
Python представляет популярный высокоуровневый язык программирования, который предназначен для создания приложений различных типов.	Однажды вечером, изучая медицинские журналы перед сном, Лена обнаружила статью профессора Иванова о последних научных открытиях.	0.0

Произведём оценку на новом наборе данных по тому же алгоритму, что и ранее. Результаты представлены в таблице № 4.

Таблица № 4

Оценки работы моделей (собственный набор данных)

Модель	**Расстояние**	**Корреляция**	
		Спирмена	**Пирсона**
1	2	3	4
Word2vec mean	cosine	0,78598	0,869359
	manhattan	0,824069	0,889657
	Euclidean	0,816202	0,882152
Word2vec max	cosine	0,770879	0,810427
	manhattan	0,764177	0,899406
	Euclidean	0,754634	0,876964
Word2vec sum	cosine	0,78598	0,869359
	manhattan	0,820738	0,805977
	Euclidean	0,810828	0,799789

1	2	3	4
GloVe mean	cosine	0,914327	0,888
	manhattan	0,875379	0,924136
	Euclidean	0,870046	0,92284
GloVe max	cosine	0,804657	0,815452
	manhattan	0,859359	0,863322
	Euclidean	0,83449	0,84826
GloVe sum	cosine	0,914327	0,888
	manhattan	0,830812	0,809045
	Euclidean	0,829341	0,811062
FastText	cosine	0,362155	0,427243
	manhattan	0,36289	0,535271
	Euclidean	0,365792	0,528708
USE	cosine	0,203136	0,315013
	manhattan	0,208735	0,339731
	Euclidean	0,203136	0,335449
ELMo mean	cosine	0,931818	0,88769
	manhattan	0,927302	0,880374
	Euclidean	0,916309	0,875325
ELMo max	cosine	0,900513	0,885446
	manhattan	0,904396	0,849871
	Euclidean	0,90601	0,856566
ELMo sum	cosine	0,931818	0,887959
	manhattan	0,76765	0,730307
	Euclidean	0,76058	0,723892
BERT mean	cosine	0,927609	0,928438
	manhattan	0,916023	0,919026
	Euclidean	0,907236	0,913528
BERT max	cosine	0,844033	0,883614
	manhattan	0,888661	0,906904
	Euclidean	0,879977	0,906105

1	2	3	4
BERT sum	cosine	0,927609	0,928438
	manhattan	0,804044	0,753122
	Euclidean	0,808273	0,749718
BERT cls	cosine	0,733811	0,224865
	manhattan	0,737551	0,483231
	Euclidean	0,742618	0,406004

Оценки работы возросли для всех моделей, по сравнению с предыдущим набором данных. Кроме того, самые высокие результаты теперь показывают модели ELMo и BERT. Самые низкие показатели сохранились у USE. Значит, предположение о разном результате при работе с одиночными словами и предложениями верно.

Результаты оценки похожести слов может стать основанием для выбора векторизатора для решения дальнейших задач обработки естественного языка, особенно в тех, где семантическая близость слов особенно важна. Метод оценки семантической близости является хорошей метрикой при обучении собственных языковых моделей. При оценке предобученных моделей было получено, что современные модели, умеющие учитывать контекст слов, предпочтительнее при работе с большими текстами. Если же задача связана с обработкой одиночных слов, то можно использовать более старые решения.

В будущем можно проверить больше различных моделей и улучшить набор данных, разнообразив тематики и увеличив количество предложений.

Литература

1. Гольдберг Й. Нейросетевые методы в обработке естественного языка. М.: ДМК Пресс, 2019. 282 с.

2. Jurafsky D., Martin J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Englewood Cliffs: Prentice-Hall, 2009. 988 с.

3. Welty C., Paritosh P., Aroyo L. Metrology for AI: From benchmarks to instruments // arXiv preprint arXiv:1911.01875. – 2019. URL: arxiv.org/abs/1911.01875 (дата обращения: 21 июня 2024).

4. Kalyan K. S., Sangeetha S. SECNLP: A survey of embeddings in clinical natural language processing //Journal of biomedical informatics. – 2020. – Т. 101. – С. 103323.

5. Reimers N., Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks //arXiv preprint arXiv:1908.10084. – 2019. URL: arxiv.org/abs/1908.10084 (дата обращения: 21 июня 2024).

6. Kutuzov A., Kuzmenko E. WebVectors: a toolkit for building web interfaces for vector semantic models //Analysis of Images, Social Networks and Texts: 5th International Conference, AIST 2016, Yekaterinburg, Russia, April 7-9, 2016, Revised Selected Papers 5. – Springer International Publishing, 2017. – С. 155-161.

7. Кукушкин А. Navec — компактные эмбеддинги для русского языка // Проект Natasha — набор Python-библиотек для обработки текстов на естественном русском языке. URL: natasha.github.io/navec/ (дата обращения: 21 июня 2024).

8. Grave E., Bojanowski P., Gupta P., Joulin A., Mikolov T. Learning word vectors for 157 languages //arXiv preprint arXiv:1802.06893. – 2018. URL: arxiv.org/abs/1802.06893 (дата обращения: 21 июня 2024).

9. Cer D., Yang Y., Kong S. Y., Hua N., Limtiaco N., John R. S., Constant N., Guajardo-Cespedes M., Yuan S., Tar C., Sung Y.-H., Strope B., Kurzweil R. Universal sentence encoder //arXiv preprint arXiv: 1803.11175. – 2018. URL: arxiv.org/abs/1803.11175 (дата обращения: 21 июня 2024).

10. Devlin J., Chang M. W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding // arXiv preprint arXiv:1810.04805. – 2018. URL: arxiv.org/abs/1810.04805 (дата обращения: 21 июня 2024).

11. Panchenko A., Ustalov D., Arefyev N., Paperno D., Konstantinova N., Loukachevitch N., Biemann C. Human and machine judgements for Russian semantic relatedness //Analysis of Images, Social Networks and Texts: 5th International Conference, AIST 2016, Yekaterinburg, Russia, April 7-9, 2016, Revised Selected Papers 5. – Springer International Publishing, 2017. – С. 221-235.

References

1. Goldberg Y. Neyrosetevyye metody v obrabotke yestestvennogo yazyka. [Neural Network Methods for Natural Language Processing] M.: DMK Press, 2019. 282 p.

2. Jurafsky D., Martin J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Englewood Cliffs: Prentice-Hall, 2009. 988 p.

3. Welty C., Paritosh P., Aroyo L. Metrology for AI: From benchmarks to instruments arXiv preprint arXiv: 1911.01875. 2019. URL: arxiv.org/abs/1911.01875 (data obrashcheniya: 21 iyunya 2024).

4. Kalyan K. S., Sangeetha S. SECNLP: A survey of embeddings in clinical natural language processing Journal of biomedical informatics. 2020. V. 101. P. 103323.

5. Reimers N., Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks arXiv preprint arXiv: 1908.10084. 2019. URL: arxiv.org/abs/1908.10084 (data obrashcheniya: 21 iyunya 2024).

6. Kutuzov A., Kuzmenko E. WebVectors: a toolkit for building web interfaces for vector semantic models Analysis of Images, Social Networks and Texts: 5th

International Conference, AIST 2016, Yekaterinburg, Russia, April 7-9, 2016, Revised Selected Papers 5. Springer International Publishing, 2017. P. 155-161.

7. Kukushkin A. Navec — kompaktnyye embeddingi dlya russkogo yazyka [Navec — compact embeddings for the Russian language]. Proyekt Natasha — nabor Python-bibliotek dlya obrabotki tekstov na yestestvennom russkom yazyke URL: natasha.github.io/navec/ (data obrashcheniya: 21 iyunya 2024).

8. Grave E., Bojanowski P., Gupta P., Joulin A., Mikolov T. Learning word vectors for 157 languages arXiv preprint arXiv: 1802.06893. 2018. URL: arxiv.org/abs/1802.06893 (data obrashcheniya: 21 iyunya 2024).

9. Cer D., Yang Y., Kong S. Y., Hua N., Limtiaco N., John R. S., Constant N., Guajardo-Cespedes M., Yuan S., Tar C., Sung Y.-H., Strope B., Kurzweil R. Universal sentence encoder arXiv preprint arXiv: 1803.11175. 2018. URL: arxiv.org/abs/1803.11175 (data obrashcheniya: 21 iyunya 2024).

10. Devlin J., Chang M. W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding arXiv preprint arXiv: 1810.04805. 2018. URL: arxiv.org/abs/1810.04805 (data obrashcheniya: 21 iyunya 2024).

11. Panchenko A., Ustalov D., Arefyev N., Paperno D., Konstantinova N., Loukachevitch N., Biemann C. Human and machine judgements for Russian semantic relatedness Analysis of Images, Social Networks and Texts: 5th International Conference, AIST 2016, Yekaterinburg, Russia, April 7-9, 2016, Revised Selected Papers 5. Springer International Publishing, 2017. P. 221-235.

Дата поступления: 26.05.2024

Дата публикации: 5.07.2024