MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 28, NO. 3, 2021
journal homepage: www.mais-journal.ru

THEORY OF DATA

Word-embedding Based Text Vectorization Using Clustering

V. I. Yuferev[1], N. A. Razin[2]

DOI: 10.18255/1818-1015-2021-3-1-20

[1]Department of Information Technologies of the Central Bank of the Russian Federation, Laboratory of innovations "Novosibirsk", 12 Neglinnaya str., Moscow 107016, Russia.
[2]Department of Counteraction to Unfair Practices, the Central Bank of the Russian Federation, 12 Neglinnaya str., Moscow 107016, Russia.

MSC2020: 97R40, 68T50
Research article
Full text in Russian

Received June 23, 2021
After revision August 16, 2021
Accepted August 25, 2021

It is known that in the tasks of natural language processing, the representation of texts by vectors of fixed length using word-embedding models makes sense in cases where the vectorized texts are short.
The longer the texts being compared, the worse the approach works. This situation is due to the fact that when using word-embedding models, information is lost when converting the vector representations of the words that make up the text into a vector representation of the entire text, which usually has the same dimension as the vector of a single word.
This paper proposes an alternative way for using pre-trained word-embedding models for text vectorization. The essence of the proposed method consists in combining semantically similar elements of the dictionary of the existing text corpus by clustering their (dictionary elements) embeddings, as a result of which a new dictionary is formed with a size smaller than the original one, each element of which corresponds to one cluster. The original corpus of texts is reformulated in terms of this new dictionary, after which vectorization is performed on the reformulated texts using one of the dictionary approaches (TF-IDF was used in the work). The resulting vector representation of the text can be additionally enriched using the vectors of words of the original dictionary obtained by decreasing the dimension of their embeddings for each cluster.
A series of experiments to determine the optimal parameters of the method is described in the paper, the proposed approach is compared with other methods of text vectorization for the text ranking problem – averaging word embeddings with TF-IDF weighting and without weighting, as well as vectorization based on TF-IDF coefficients.

Keywords: word embedding; Fasttext; TF-IDF; averaging; clustering; text similarity; distance; text ranking

INFORMATION ABOUT THE AUTHORS

Vitaly I. Yuferev
correspondence author
orcid.org/0000-0003-3245-6240. E-mail: YuferevVI@mail.cbr.ru
Chief expert, Master of science.

Nikolai A. Razin
orcid.org/0000-0002-7669-776X. E-mail: razinna@cbr.ru
Head of division, PhD.

For citation: V. I. Yuferev and N. A. Razin, "Word-embedding Based Text Vectorization Using Clustering", *Modeling and analysis of information systems*, vol. 28, no. 3, pp. 1-20, 2021.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 28, № 3, 2021
сайт журнала: www.mais-journal.ru

THEORY OF DATA

Векторизация текстов на основе word-embedding моделей с использованием кластеризации

В. И. Юферев[1], Н. А. Разин[2]

DOI: 10.18255/1818-1015-2021-3-1-20

[1]Департамент информационных технологий Центрального банка Российской Федерации, Инновационная лаборатория «Новосибирск», ул. Неглинная, д. 12, г. Москва, 107016 Россия.
[2]Департамент противодействия недобросовестным практикам, Центральный банк Российской Федерации, ул. Неглинная, д. 12, г. Москва, 107016 Россия.

УДК 004.8
Научная статья
Полный текст на русском языке

Получена 23 июня 2021 г.
После доработки 16 августа 2021 г.
Принята к публикации 25 августа 2021 г.

Известно, что в задачах обработки естественного языка представление текстов векторами фиксированной длины с использованием word-embedding моделей оправдано в тех случаях, когда векторизуемые тексты являются короткими. Чем сравниваемые тексты длиннее, тем подход работает хуже. Такая ситуация обусловлена тем, что при использовании word-embedding моделей происходит потеря информации при преобразовании векторных представлений слов, составляющих текст, в векторное представление всего текста, имеющее обычно ту же размерность, что и вектор отдельного слова.

В настоящей работе предлагается альтернативный способ использования предобученных word-embedding моделей для векторизации текстов. Суть предлагаемого способа заключается в объединении семантически близких элементов словаря имеющегося корпуса текстов путем кластеризации их (элементов словаря) эмбеддингов, в результате чего формируется новый словарь размером меньше исходного, каждый элемент которого соответствует одному кластеру. Исходный корпус текстов переформулируется в терминах этого нового словаря, после чего на переформулированных текстах выполняется векторизация одним из словарных подходов (в работе применялся TF-IDF). Полученное векторное представление текста дополнительно может обогащаться с использованием векторов слов исходного словаря, полученных путем уменьшения размерности их эмбеддингов по каждому кластеру. В работе описана серия экспериментов по определению оптимальных параметров предлагаемого подхода; для задачи ранжирования текстов приведено сравнение подхода с другими способами векторизации – усреднением эмбеддингов слов со взвешиванием по TF-IDF и без взвешивания, а также с векторизацией на основе TF-IDF коэффициентов.

Ключевые слова: эмбеддинговые модели; Fasttext; TF-IDF; усреднение; кластеризация; семантическое сходство текстов; определение расстояний; ранжирование текстов

ИНФОРМАЦИЯ ОБ АВТОРАХ

Виталий Иванович Юферев автор для корреспонденции	orcid.org/0000-0003-3245-6240. E-mail: YuferevVI@mail.cbr.ru Главный эксперт, магистр техники и технологий.
Николай Алексеевич Разин	orcid.org/0000-0002-7669-776X. E-mail: razinna@cbr.ru Начальник отдела, кандидат физ.-мат. наук.

Для цитирования: V. I. Yuferev and N. A. Razin, “Word-embedding Based Text Vectorization Using Clustering”, *Modeling and analysis of information systems*, vol. 28, no. 3, pp. 1-20, 2021.

Введение

Распространенной задачей в области обработки естественного языка является ранжирование текстов, то есть определение, какой из двух текстов T_1 или T_2 семантически ближе к тексту T.

Базовый подход для определения семантического сходства текстов состоит из двух основных этапов: представление сравниваемых текстов в векторном виде, отражающем семантику текста, и последующее определение расстояний между полученными векторами [1].

Существуют различные способы представления текстов в векторном виде: с использованием словарей, с использованием word-embedding моделей, с использованием языковых моделей, основанных на архитектуре Transformer.

К недостаткам словарных подходов, в частности основанных на TF-IDF [2], можно отнести отсутствие учета семантики слов [3] при определении близости текстов, если составляющие их слова не пересекаются, то они будут определены как далекие друг от друга, независимо от того, содержат ли тексты слова, похожие по смыслу. Применение современных языковых моделей, основанных на архитектуре Transformer, также имеет некоторые ограничения: высокие требования к вычислительным ресурсам и предельный размер входа [4—6].

Применение word-embedding моделей для векторизации текстов выглядит оправданным в случаях, когда имеют место указанные выше ограничения других подходов. Особенно актуально применение предобученных word-embedding моделей, когда имеющийся корпус текстов страдает от недостатка данных [7].

Наиболее простой и часто применяемый подход по формированию векторного представления текста с использованием word-embedding моделей заключается в формировании эмбеддингов входящих в текст слов с последующим формированием вектора текста путем усреднения (сложения) полученных эмбеддингов слов [8, 9]. В качестве улучшения подхода дополнительно может выполняться взвешивание полученных векторов эмбеддингов слов, например, по TF-IDF [10].

Решение прикладных задач обработки естественного языка подтверждает, что применение подхода с векторизацией текстов при помощи усреднения (сложения) эмбеддингов слов, как правило, даёт приемлемое качество (конкретные метрики и их значения зависят от задачи) на коротких текстах (на предложениях и меньше) [11] и по мере увеличения длины текстов качество снижается до неприемлемого.

Низкое качество при усреднении (сложении) на длинных текстах можно объяснить следующим. Эмбеддинг слова представляет его семантику в виде вектора. Усреднение (сумма) двух эмбеддингов (обозначим A и B) также представляет собой вектор (обозначим C) той же размерности. При этом возникает неопределенность, связанная с тем, что вектор C может быть получен указанной комбинацией (усреднение, сумма) как исходных векторов A и B, так и некоторых других векторов D и E, отражающих семантику, отличную от семантики, кодируемой векторами A и B. Соответственно, чем больше исходных эмбеддингов усредняется, тем выше у результирующего вектора неопределенность относительно семантики исходных слов.

Попытка увеличения размерности вектора текста по сравнению с векторами слов приводится в [12—14]. Как и в описываемом в настоящей работе подходе в этих работах предлагается выполнить кластеризацию на словаре корпуса текстов. Однако указанные подходы имеют следующие ограничения: необходимость наличия достаточно большого корпуса текстов для обучения wod2vec модели, отсутствие описания возможности применения для n-грамм.

1. Описание подхода

Чтобы обойти ограничения алгоритмов по векторизации текстов – словарного и основанного на word-embedding – предлагается совместить данные подходы.

Попытки совмещения word-embedding и TF-IDF предпринимались и ранее. Суть таких улучшений состоит в использовании при усреднении эмбеддингов слов весовых коэффициентов, соответствующих TF-IDF этих слов. Однако описанная выше во Введении неоднозначность не устраняется, поскольку размерность итогового вектора остается неизменной.

Далее приводится алгоритм формирования векторного представления текста в рамках предлагаемого подхода. При этом необходимо учитывать, что векторизация текста обычно выполняется в рамках решения какой-либо прикладной задачи. Приведенный алгоритм актуален для решения задачи ранжирования текстов (описана во Введении).

1. Имеется исходное множество текстов T, на которых требуется выполнять ранжирование, то есть для заданного текста t_i из T упорядочить множество T по степени близости к t_i. Также имеется предобученная word-embedding модель M.
2. На множестве T строится словарь V всех слов, входящих в тексты t_i из T.
3. Для каждого слова v_i из словаря V получаем его эмбеддинг при помощи предобученной модели M: $e_i = M(v_i)$. Все e_i в совокупности составляют множество эмбеддингов E.
4. Выполняется кластеризация на множестве E, в результате которой получается кластеризующая модель C, которая по эмбеддингу слова выдает кластер, к которому он относится. Для каждого e_i из E определяется его кластер $c_i=C(e_i)$. Множество всех кластеров c_i модели C также обозначим символом C.
5. Для каждого текста t_i из T получаем новый текст t^c_i следующим способом:
   5.1. Копируем t_i в t^c_i
   5.2. Для каждого слова w_j из текста t^c_i
      a) Определяем его кластер $c_j = C(M(w_j))$.
      b) Заменяем в тексте t^c_i слово w_j на номер соответствующего кластера c_j

   В результате путем замены всех t_i из T на t^c_i получено новое множество T_c. Все тексты этого множества состоят из номеров кластеров с символами-разделителями между ними.
6. Выполняется векторизация текстов T_c при помощи TF-IDF на n-граммах слов. В результате чего получается:
   - множество X_c TF-IDF-векторов x^c_i для каждого t^c_i из T_c,
   - словарь n-грамм V_c, а также
   - N_{min} и N_{max} — заданные в качестве входных параметров алгоритма минимальная и максимальная длины n-грамм, используемых для построения словаря TF-IDF.
7. Для каждого t_i из T x^c_i из X_c далее рассматривается как векторное представление текста t_i.

Поскольку объединенные в один кластер одни слова исходного словаря могут быть ближе друг к другу, чем другие, обогащение векторного представления информацией о взаимной близости слов кластера может повысить качество этого векторного представления. Чтобы учесть в векторном представлении текстов взаимную близость слов друг к другу в рамках одного кластера, предлагается следующее улучшение подхода в виде дополнительных шагов алгоритма.

Формирование обогащающих векторов слов.

8. По каждому кластеру c_i из C.
   8.1. Для каждого относящегося к c_i эмбеддинга e_j слова w_j исходного словаря V выполняется снижение размерности до одного (располагаются на одной числовой оси). Полученные в результате числовые значения обозначим через e^r_j.
   8.2. С использованием min-max–нормализации выполняется масштабирование числовых представлений e^r_j эмбеддингов e_j слов w_j кластера c_i на отрезок [0;1]. Отрезок [0;1] разбивается на D-1 частей (D — размерность обогащающего вектора слова, задается в качестве одного из входных параметров алгоритма), пронумерованных от 1 до D-1.

8.3. Для каждого слова w_j исходного словаря V выполняется формирование обогащающего вектора e^e_j следующим образом. Нулевая позиция вектора всегда заполняется значением «1». Далее по каждому из D-1 отрезков, на которые разбит интервал [0;1], если e^r_j попадает в k-й интервал, то k-я позиция вектора заполняется значением «1», иначе «0».

Формирование обогащенных векторов текстов.

9. Для каждого текста t^c_i из T_c.
   Каждая позиция (обозначим индексом j) TF-IDF вектора x^c_i
   соответствует TF-IDF-коэффициенту x^c_{ij} для n-граммы w^c_j из словаря V_c. Для каждой j-й позиции вектора x^c_i сформируем обогащающий вектор e^{xc}_j следующим образом.
   9.1. На основе текста t_i из T сформируем обогащающие векторы всех входящих в t_i n-грамм длиной от N_{min} до N_{max} путем конкатенации обогащающих векторов входящих в них слов (полученных на шаге 8.3). Максимальная длина каждого такого вектора равна N_{max}*D. Если вектор получен из n-граммы длиной меньше N_{max}, вектор дополняется справа нулями до максимальной длины.
   9.2. Каждой n-грамме из t_i соответствует некоторая n-грамма из V_c. Выполним усреднение обогащающих векторов n-грамм из t_i по соответствующим им n-граммам из V_c. Таким образом, получены обогащающие векторы для тех позиций вектора x^c_i, для которых соответствующие n-граммы w^c_j из V_c входят в t^c_i.
   9.3. Если соответствующие j-й позиции из x^c_i n-граммы отсутствуют в t^c_j, то соответствующие обогащающие векторы состоят из N_{max}*D нулевых элементов.
   9.4. Обогащенный вектор x^{ce}_i текста t_i вычисляется конкатенацией векторов x^c_{ij} *e^{xc}_j (произведение TF-IDF-коэффициента n-граммы w^c_j на ее обогащающий вектор).

Таким образом, получено множество обогащенных векторных представлений X_{ce} текстов T.

2. Апробация подхода

2.1. Условия проведения апробации

Проверка качества подхода по векторизации текстов осуществлялась в контексте решения задачи ранжирования текстов по семантической близости.

Для проверки качества подхода имелось в распоряжении 13772 примера вида: (T_1, T_2, T), где T_1, T_2 и T — тексты, такие что T_1 ближе к T, чем T_2.

Все тексты из 13772 примеров сформированы на основе 940 текстов, представляющих собой внутреннюю переписку в Банке России.

Качество подхода по ранжированию текстов определяется по точности (Accuracy) как отношение количества корректно определенных примеров к общему их количеству.

Распределение длин текстов корпуса представлено на рисунке 1. По горизонтальной оси отложены длины текстов в символах. По вертикальной — количество текстов заданной длины.

Тексты корпуса предварительно были приведены к нижнему регистру, из них были отфильтрованы символы, не являющиеся пробелом или буквами русского или латинского алфавитов.

Размер словаря корпуса текстов составляет 73731 слово.

В качестве меры, при помощи которой определяется близость между векторными представлениями текстов, использовалась косинусная близость.

2.2. Определение оптимальных параметров

2.2.1. Параметры алгоритма

В рамках предлагаемого подхода:

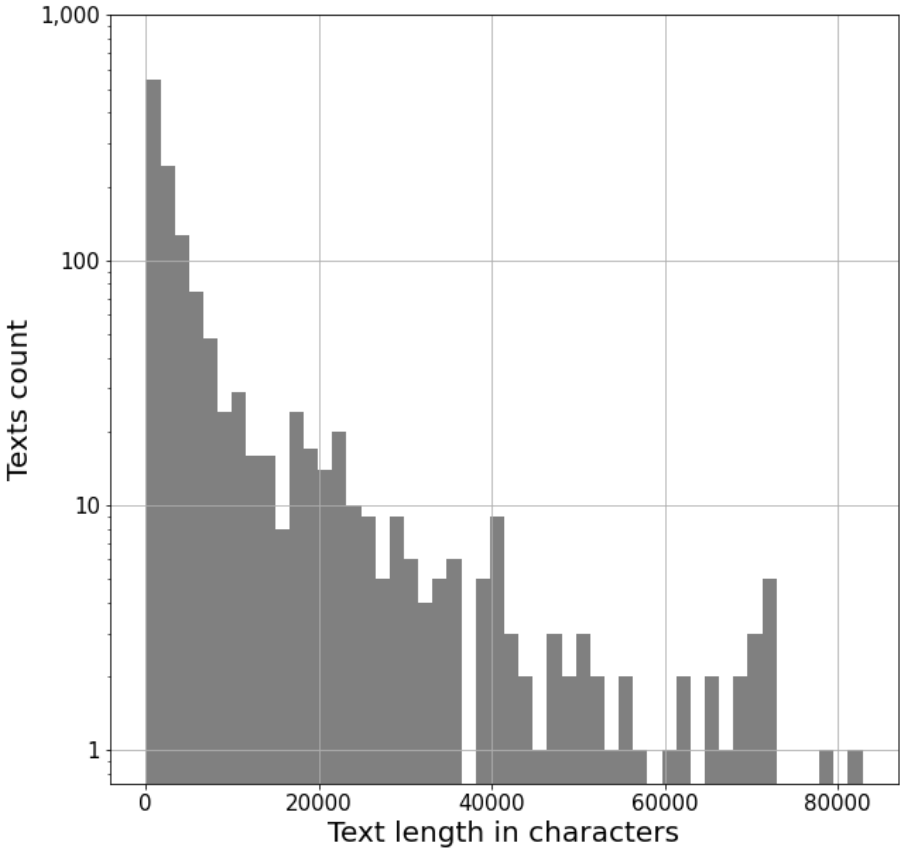

Fig. 1. Distribution of text lengths in the corpus

Рис. 1. Распределение длин текстов в корпусе

- в качестве word-embedding модели использовалась FastText модель от DeepPavlov[1] , обученная на русскоязычной части Wikipedia совместно с набором данных Lenta ru news. Размерность выходного вектора 300. Использованная в работе модель Fasttext формирует эмбеддинг слова с использованием входящих в это слово символьных последовательностей, что позволяет применять ее к словам out-of-vocabulary, то есть таким, которые в обучении FastText модели не участвовали.

- в качестве метода кластеризации использован Kmeans,

- в качестве метода снижения размерности на шаге 8.1 использован t-SNE,

- при векторизации TF-IDF устанавливается фильтр на минимальное количество документов, в которых встречается n-грамма, равная двум.

Изменяемыми параметрами для алгоритма являются: количество кластеров, диапазон n-грамм, размерность обогащающего вектора слова.

Применим предлагаемый подход для всех возможных комбинаций параметров из представленных в Таблице 1, выполнив серию экспериментов по три для каждой комбинации параметров. Здесь размерность обогащающего вектора слова, равная нулю, означает, что обогащение не производится, а используются непосредственно TF-IDF-векторы.

[1]http://files.deeppavlov.ai/embeddings/ft_native_300_ru_wiki_lenta_lower_case/ft_native_300_ru_wiki_lenta_lower_case.bin

Table 1. The values of the parameters to be checked

Таблица 1. Проверяемые значения параметров алгоритма

Параметр	**Значения**
Количество кластеров	100, 1000, 3000, 5000, 10000, 15000, 20000, 25000, 45000, 65000
Размерность обогащающего вектора слова	0, 2, 5, 10, 15, 20
Диапазон n-грамм	(1,1), (1,2), (1,3), (1,4)

2.2.2. Результаты

Таблица с полными результатами экспериментов приведена в Приложении А.

Значения параметров, на которых получены лучшие показатели точности для различных диапазонов n-грамм, представлены в Таблице 2.

Из таблицы видно, что лучшие результаты получены для следующей комбинации параметров: количество кластеров 25000, размер обогащающего вектора слова 2, диапазон n-грамм (1, 4).

Table 2. Parameters that give the best results

Таблица 2. Параметры с лучшими результатами

диапазон n-gram	**Количество кластеров**	**Размерность обогащающего вектора слова**	**Точность**	**Стандартное отклонение**
(1, 1)	20000	0	0.934	1.321
(1, 2)	25000	0	0,940	1.151
(1, 3)	25000	2	0.944	1.156
(1, 4)	25000	2	**0.947**	1.16

2.2.3. Интерпретация результатов

Отметим, что значение в 25000 кластеров, на котором получено лучшее значение точности, составляет приблизительно одну третью часть от размера словаря корпуса текстов (73731 слово).

Ниже приведены графики зависимости точности от параметров алгоритма.

На Рисунке 2 приведен график зависимости точности от количества кластеров для разной размерности обогащающего вектора слова. Значения точности для графика получены усреднением точности для всех диапазонов n-грамм.

Из графика видно, что увеличение количества кластеров до определенного значения приводит к значительному повышению точности. При количестве кластеров 25000 точность достигает максимального значения, после чего начинает убывать.

Завершается график горизонтальным участком, что можно объяснить следующим: чем ближе параметр «количество кластеров» при кластеризации к количеству кластеризуемых объектов, тем больше получается «пустых» кластеров, то есть, несмотря на увеличение значения параметра «количество кластеров», словарь V_c описанного алгоритма растет незначительно.

Зависимость точности от размерности обогащающего вектора слова имеет разный характер, в зависимости от количества кластеров, в связи с чем для зависимости точности от размерности обогащающего вектора слова приводятся два графика: график для количества кластеров до 20000 приведен на Рисунке 3, а график для количества кластеров 20000 и более приведен на Рисунке 4. Значения точности для графиков получены усреднением точности для всех диапазонов n-грамм.

Из графиков видно, что чем меньше количество кластеров, тем больший прирост точности дает процедура обогащения, а начиная с определенного количества кластеров, в целом, обогащение понижает точность. При этом наиболее существенный прирост точности наблюдается на размерности, равной двум. Также видно, что при использовании обогащения максимальный прирост точности на малом количестве кластеров больше, чем максимальное снижение на большом.

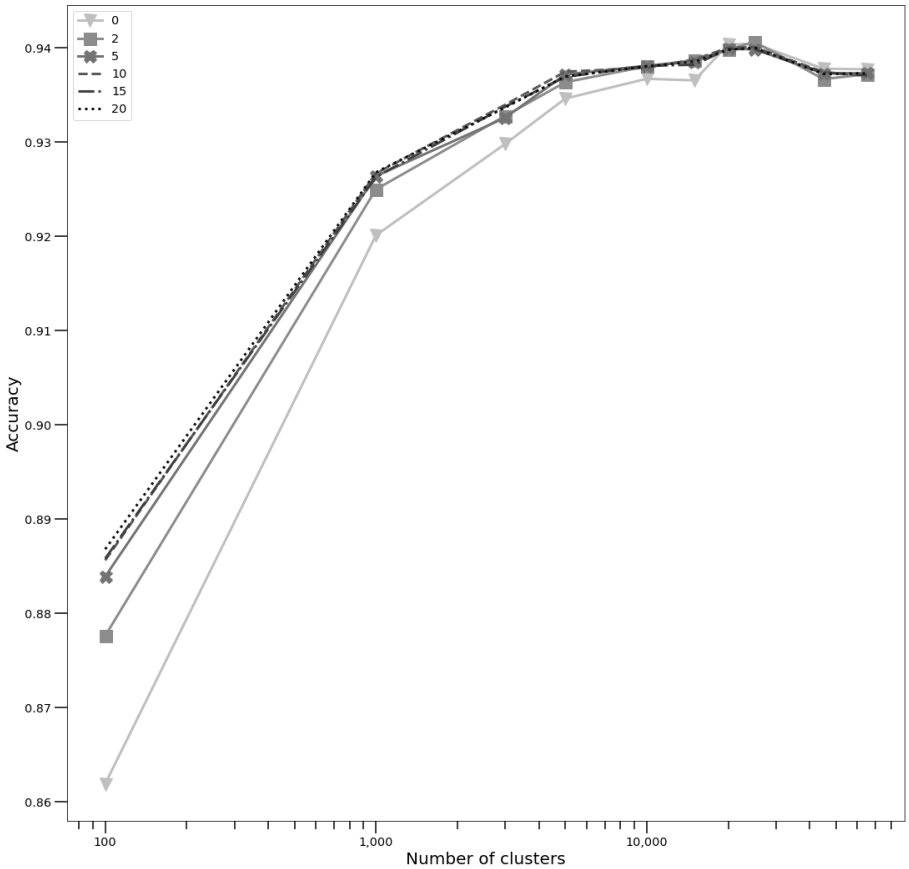

Fig. 2. Dependence of accuracy on the number of clusters

Рис. 2. Зависимость точности от количества кластеров

График зависимости точности от количества кластеров для различных диапазонов используемых n-грамм приведен на Рисунке 5. Значения точности для графика получены усреднением точности для всех размерностей обогащающего вектора. Из графика видно, что чем больше используемый диапазон n-грамм, тем точность выше.

2.3. Сравнение с baseline-подходами

Сравниваются следующие подходы:
- предлагаемый в настоящей работе (обогащенные TF-IDF-векторы на номерах кластеров),
- TF-IDF на текстах корпуса,
- TF-IDF на лемматизированных текстах корпуса,
- усреднение эмбеддингов,
- усреднение эмбеддингов, взвешенных по TF-IDF.

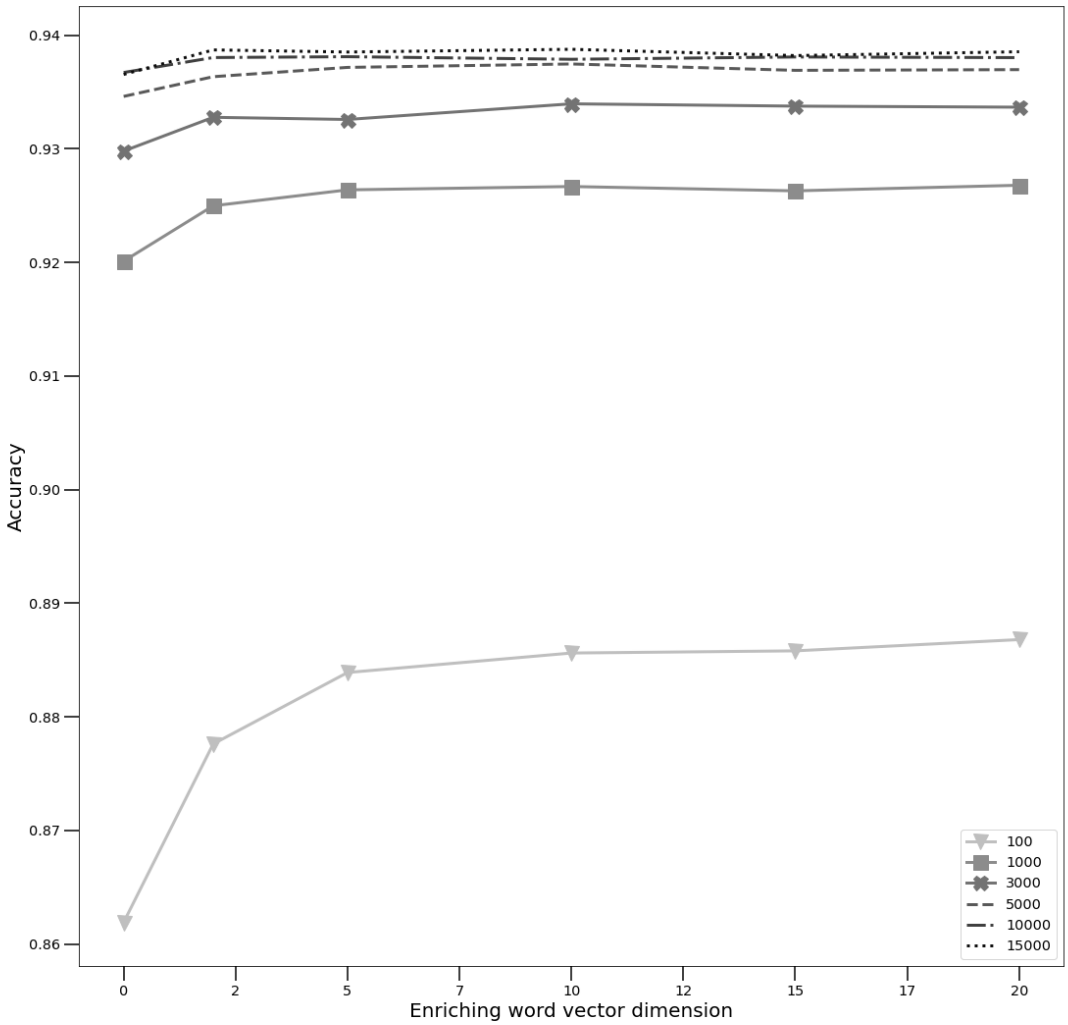

Fig. 3. Dependence of accuracy on enriching word vector for the number of clusters less than 20,000

Рис. 3. Зависимость точности от размерности обогащающего вектора слова для количества кластеров менее 20000

2.3.1. TF-IDF

Векторизация текстов на основе TF-IDF для следующих диапазонов n-грамм: (1, 1), (1, 2), (1, 3), (1, 4).

Векторизация TF-IDF осуществлялась с использованием библиотеки sklearn.

При векторизации TF-IDF установлен фильтр на минимальное количество документов, в которых встречается n-грамма, равная двум.

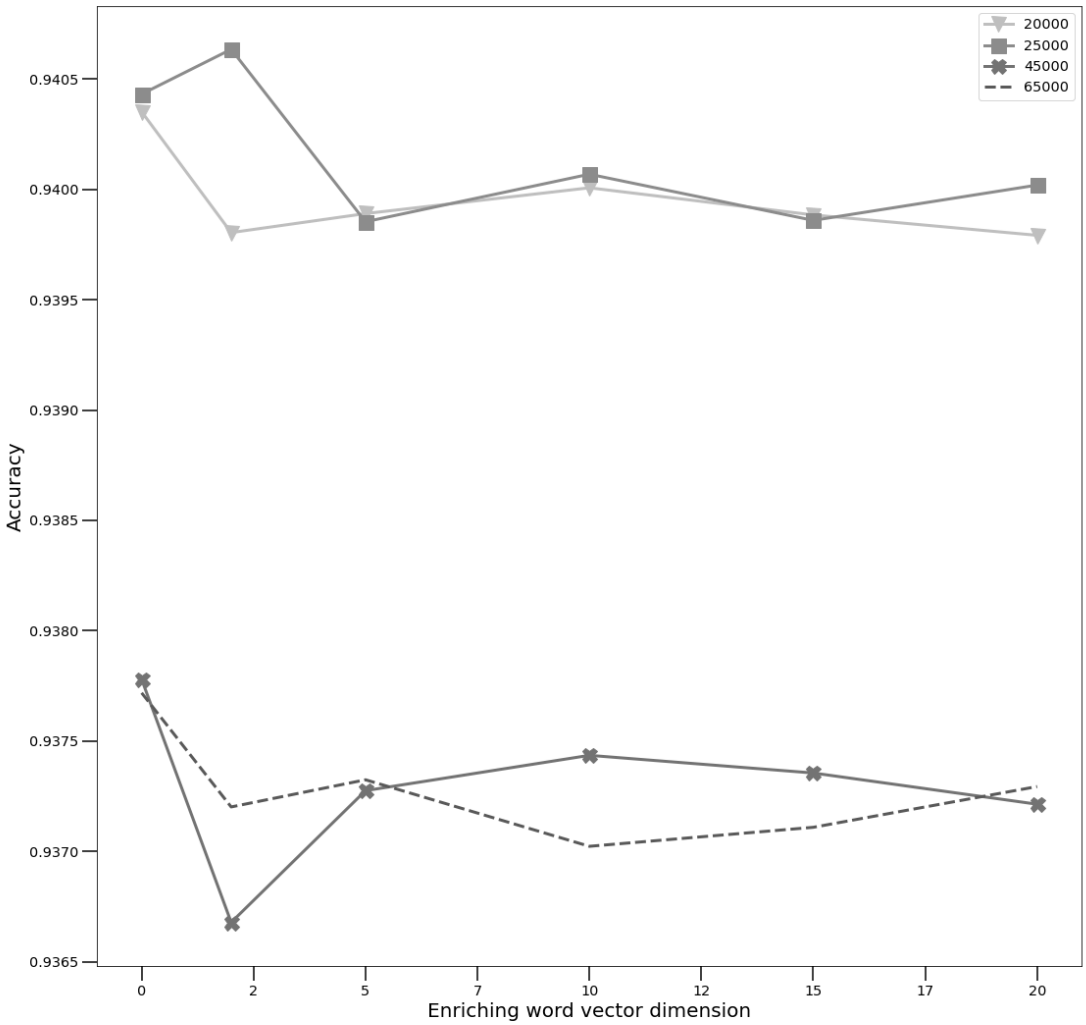

Fig. 4. Dependence of accuracy on the dimension of the enriching word vector for the number of clusters of 20,000 or more

Рис. 4. Зависимость точности от размерности обогащающего вектора слова для количества кластеров 20000 и более

2.3.2. TF-IDF с лемматизацией

Практика решения задач обработки естественного языка показывает, что поскольку в русском языке разные формы слова часто имеют разное написание, непосредственное применение подхода TF-IDF для векторизации текстов в большинстве случаев работает хуже, чем в случае, если выполнено предварительное приведение слов текста в нормальную форму.

В эксперименте приведение слов в нормальную форму выполнялось путем лемматизации при помощи библиотеки Mystem от компании Яндекс.

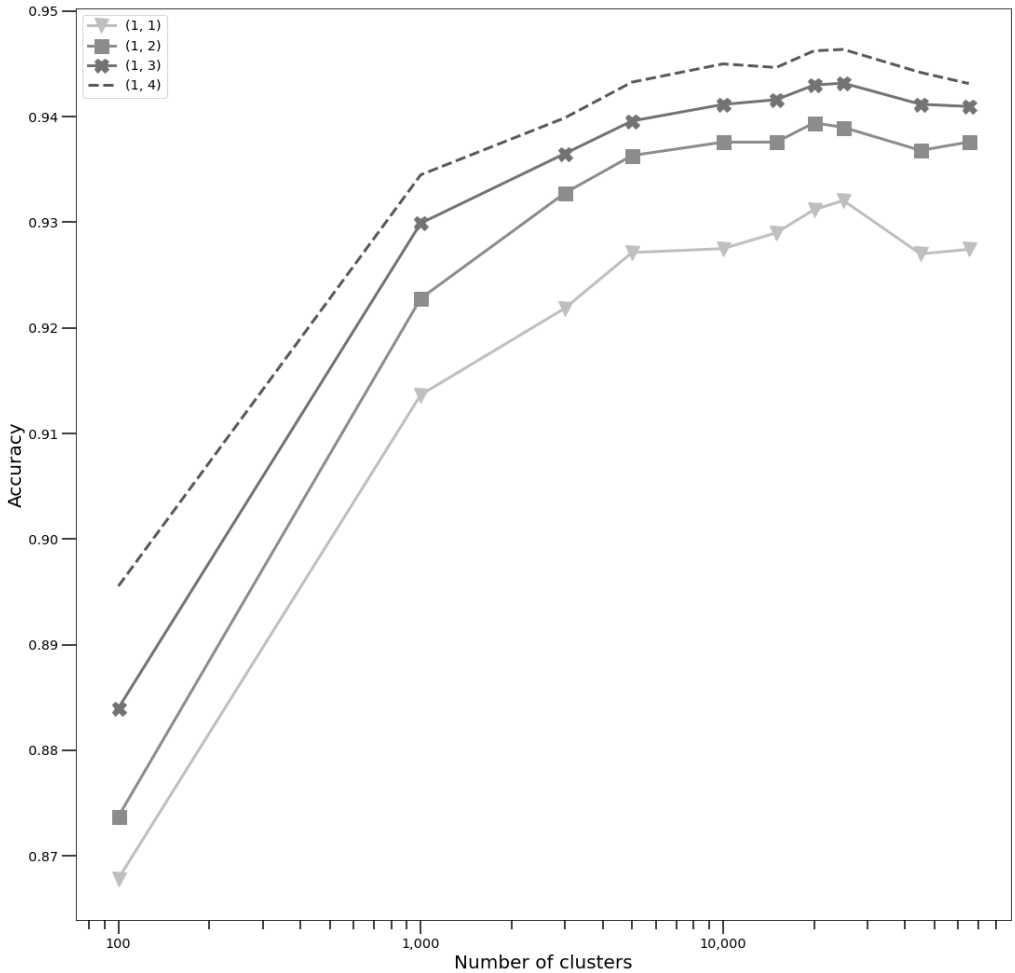

Fig. 5. Dependence of accuracy on the number of clusters for different n-gram ranges

Рис. 5. Зависимость точности от количества кластеров для разных диапазонов n-грамм

Векторизация текстов на основе TF-IDF выполнялась для следующих видов n-грамм: (1, 1), (1, 2), (1, 3), (1, 4).

Векторизация TF-IDF осуществлялась с использованием библиотеки sklearn.

При векторизации TF-IDF установлен фильтр на минимальное количество документов, в которых встречается n-грамма, равная двум.

2.3.3. Усреднение эмбеддингов

Усреднение эмбеддингов для текста заключается в преобразовании каждого слова текста в векторное представление при помощи word-embedding модели и последующем усреднении эмбеддингов слов с получением итогового вектора текста той же размерности, что и векторы эмбеддингов.

2.3.4. Усреднение эмбеддингов, взвешенных по TF-IDF

Подход отличается от простого усреднения эмбеддингов тем, что при усреднении эмбеддинги слов берутся с весовыми коэффициентами, соответствующими их TF-IDF-коэффициентам, рассчитанным на имеющемся корпусе текстов [10]. В настоящей работе коэффициенты TF-IDF рассчитывались на всем корпусе документов.

2.3.5. Результаты сравнения подходов

Результаты сравнения подходов приведены в таблице 3.

Table 3. Methods comparison **Таблица 3.** Сравнение подходов

Диапазон n-грамм	**TF-IDF**	**TF-IDF со стеммингом**	**Усреднение эмбеддингов Fasttext**	**Усреднение эмбеддингов Fasttext со взвешиванием по TD-IDF**	**Обогащенные TF-IDF векторы на кластерах**
(1,1)	0,9287	0,9279	0,779	0,8679	0.9302
(1,2)	0,9333	0,9364	не применимо	не применимо	0.9405
(1,3)	0,9366	0,9396	не применимо	не применимо	0.9459
(1,4)	0,9381	0,94	не применимо	не применимо	0.947

Заключение

В настоящей работе предложен подход по векторизации текстов при помощи word-embedding моделей, экспериментально определены оптимальные параметры для решения задачи ранжирования текстов, представляющих собой 940 писем внутренней переписки Банка России, выполнено сравнение предложенного подхода с распространенными подходами.

Предлагаемый подход в сравнении с представленными baseline-подходами показывает лучшие результаты.

Использовавшийся в предлагаемом подходе алгоритм Kmeans для кластеризации эмбеддингов элементов словаря в качестве входного параметра принимает количество кластеров. При этом остается открытым вопрос, насколько компактными получаются кластеры, что может влиять на качество представления текста при помощи кластеров.

Для решения данной проблемы видятся перспективными следующие направления дальнейших исследований:

- Анализ компактности получаемых кластеров для улучшения представления текстов в виде номеров кластеров, например, фильтрация элементов кластера по порогу расстояния до центроида.
- Использование такого подхода для кластеризации, при котором параметры определяют расстояния между объектами, а не количество кластеров. Однако, по сравнению с Kmeans, такие подходы более требовательны к вычислительным ресурсам. Соответственно, в контексте данного направления актуально решение задачи поиска эффективного способа кластеризации элементов словаря.

Предложенный подход для векторизации апробирован в рамках решения задачи ранжирования текстов. Необходимо исследовать подход на применимость для решения других задач обработки естественного языка.

Appendix A. Experimental results
Приложение А. Результаты экспериментов

№ п/п	Количество кластеров	Диапазон n-грамм	Размерность вектора обогащения слова	Точность	Стандартное отклонение
1	100	(1, 1)	0	0,858	1,051
2	100	(1, 1)	2	0,863	1,058
3	100	(1, 1)	5	0,87	1,066
4	100	(1, 1)	10	0,871	1,067
5	100	(1, 1)	15	0,871	1,067
6	100	(1, 1)	20	0,873	1,069
7	100	(1, 2)	0	0,86	1,054
8	100	(1, 2)	2	0,871	1,066
9	100	(1, 2)	5	0,877	1,074
10	100	(1, 2)	10	0,878	1,075
11	100	(1, 2)	15	0,878	1,075
12	100	(1, 2)	20	0,879	1,076
13	100	(1, 3)	0	0,862	1,056
14	100	(1, 3)	2	0,882	1,08
15	100	(1, 3)	5	0,888	1,087
16	100	(1, 3)	10	0,89	1,09
17	100	(1, 3)	15	0,891	1,091
18	100	(1, 3)	20	0,892	1,092
19	100	(1, 4)	0	0,867	1,062
20	100	(1, 4)	2	0,895	1,096
21	100	(1, 4)	5	0,901	1,104
22	100	(1, 4)	10	0,903	1,106
23	100	(1, 4)	15	0,903	1,106
24	100	(1, 4)	20	0,904	1,107
25	1000	(1, 1)	0	0,91	1,115
26	1000	(1, 1)	2	0,912	1,118
27	1000	(1, 1)	5	0,915	1,121
28	1000	(1, 1)	10	0,915	1,12
29	1000	(1, 1)	15	0,915	1,12
30	1000	(1, 1)	20	0,915	1,12
31	1000	(1, 2)	0	0,918	1,125
32	1000	(1, 2)	2	0,923	1,13
33	1000	(1, 2)	5	0,924	1,131
34	1000	(1, 2)	10	0,924	1,132
35	1000	(1, 2)	15	0,924	1,131
36	1000	(1, 2)	20	0,924	1,132
37	1000	(1, 3)	0	0,924	1,132
38	1000	(1, 3)	2	0,93	1,139
39	1000	(1, 3)	5	0,931	1,14

40	1000	(1, 3)	10	0,932	1,141
41	1000	(1, 3)	15	0,931	1,14
42	1000	(1, 3)	20	0,931	1,141
43	1000	(1, 4)	0	0,928	1,136
44	1000	(1, 4)	2	0,935	1,145
45	1000	(1, 4)	5	0,936	1,146
46	1000	(1, 4)	10	0,936	1,147
47	1000	(1, 4)	15	0,936	1,146
48	1000	(1, 4)	20	0,937	1,147
49	3000	(1, 1)	0	0,92	1,126
50	3000	(1, 1)	2	0,921	1,128
51	3000	(1, 1)	5	0,922	1,129
52	3000	(1, 1)	10	0,923	1,131
53	3000	(1, 1)	15	0,923	1,13
54	3000	(1, 1)	20	0,923	1,13
55	3000	(1, 2)	0	0,929	1,138
56	3000	(1, 2)	2	0,933	1,142
57	3000	(1, 2)	5	0,932	1,142
58	3000	(1, 2)	10	0,934	1,144
59	3000	(1, 2)	15	0,934	1,144
60	3000	(1, 2)	20	0,934	1,144
61	3000	(1, 3)	0	0,934	1,143
62	3000	(1, 3)	2	0,937	1,148
63	3000	(1, 3)	5	0,936	1,147
64	3000	(1, 3)	10	0,937	1,148
65	3000	(1, 3)	15	0,938	1,148
66	3000	(1, 3)	20	0,937	1,148
67	3000	(1, 4)	0	0,937	1,147
68	3000	(1, 4)	2	0,94	1,151
69	3000	(1, 4)	5	0,94	1,151
70	3000	(1, 4)	10	0,941	1,152
71	3000	(1, 4)	15	0,941	1,152
72	3000	(1, 4)	20	0,941	1,152
73	5000	(1, 1)	0	0,926	1,134
74	5000	(1, 1)	2	0,927	1,135
75	5000	(1, 1)	5	0,927	1,136
76	5000	(1, 1)	10	0,928	1,137
77	5000	(1, 1)	15	0,927	1,135
78	5000	(1, 1)	20	0,927	1,136
79	5000	(1, 2)	0	0,934	1,144
80	5000	(1, 2)	2	0,936	1,147
81	5000	(1, 2)	5	0,937	1,148
82	5000	(1, 2)	10	0,937	1,148
83	5000	(1, 2)	15	0,937	1,147
84	5000	(1, 2)	20	0,936	1,147

85	5000	(1, 3)	0	0,937	1,148
86	5000	(1, 3)	2	0,94	1,151
87	5000	(1, 3)	5	0,94	1,151
88	5000	(1, 3)	10	0,941	1,152
89	5000	(1, 3)	15	0,94	1,151
90	5000	(1, 3)	20	0,94	1,151
91	5000	(1, 4)	0	0,941	1,152
92	5000	(1, 4)	2	0,943	1,155
93	5000	(1, 4)	5	0,944	1,156
94	5000	(1, 4)	10	0,944	1,156
95	5000	(1, 4)	15	0,944	1,156
96	5000	(1, 4)	20	0,944	1,156
97	10000	(1, 1)	0	0,928	1,137
98	10000	(1, 1)	2	0,927	1,135
99	10000	(1, 1)	5	0,928	1,137
100	10000	(1, 1)	10	0,927	1,136
101	10000	(1, 1)	15	0,927	1,136
102	10000	(1, 1)	20	0,927	1,136
103	10000	(1, 2)	0	0,936	1,147
104	10000	(1, 2)	2	0,938	1,149
105	10000	(1, 2)	5	0,938	1,148
106	10000	(1, 2)	10	0,938	1,149
107	10000	(1, 2)	15	0,938	1,149
108	10000	(1, 2)	20	0,938	1,148
109	10000	(1, 3)	0	0,94	1,151
110	10000	(1, 3)	2	0,942	1,153
111	10000	(1, 3)	5	0,941	1,153
112	10000	(1, 3)	10	0,941	1,153
113	10000	(1, 3)	15	0,941	1,153
114	10000	(1, 3)	20	0,942	1,153
115	10000	(1, 4)	0	0,943	1,155
116	10000	(1, 4)	2	0,945	1,158
117	10000	(1, 4)	5	0,945	1,158
118	10000	(1, 4)	10	0,945	1,158
119	10000	(1, 4)	15	0,946	1,158
120	10000	(1, 4)	20	0,945	1,158
121	15000	(1, 1)	0	0,928	1,136
122	15000	(1, 1)	2	0,93	1,139
123	15000	(1, 1)	5	0,929	1,138
124	15000	(1, 1)	10	0,929	1,138
125	15000	(1, 1)	15	0,929	1,138
126	15000	(1, 1)	20	0,929	1,138
127	15000	(1, 2)	0	0,936	1,146
128	15000	(1, 2)	2	0,938	1,149
129	15000	(1, 2)	5	0,938	1,149

130	15000	(1, 2)	10	0,938	1,149
131	15000	(1, 2)	15	0,938	1,148
132	15000	(1, 2)	20	0,938	1,149
133	15000	(1, 3)	0	0,94	1,151
134	15000	(1, 3)	2	0,942	1,154
135	15000	(1, 3)	5	0,942	1,154
136	15000	(1, 3)	10	0,942	1,154
137	15000	(1, 3)	15	0,942	1,153
138	15000	(1, 3)	20	0,942	1,154
139	15000	(1, 4)	0	0,943	1,155
140	15000	(1, 4)	2	0,945	1,157
141	15000	(1, 4)	5	0,945	1,157
142	15000	(1, 4)	10	0,945	1,158
143	15000	(1, 4)	15	0,945	1,157
144	15000	(1, 4)	20	0,945	1,158
145	20000	(1, 1)	0	0,934	1,143
146	20000	(1, 1)	2	0,931	1,14
147	20000	(1, 1)	5	0,931	1,14
148	20000	(1, 1)	10	0,931	1,14
149	20000	(1, 1)	15	0,931	1,14
150	20000	(1, 1)	20	0,93	1,14
151	20000	(1, 2)	0	0,94	1,151
152	20000	(1, 2)	2	0,939	1,15
153	20000	(1, 2)	5	0,939	1,15
154	20000	(1, 2)	10	0,94	1,151
155	20000	(1, 2)	15	0,939	1,15
156	20000	(1, 2)	20	0,939	1,15
157	20000	(1, 3)	0	0,943	1,154
158	20000	(1, 3)	2	0,943	1,155
159	20000	(1, 3)	5	0,943	1,155
160	20000	(1, 3)	10	0,943	1,155
161	20000	(1, 3)	15	0,943	1,155
162	20000	(1, 3)	20	0,943	1,155
163	20000	(1, 4)	0	0,945	1,158
164	20000	(1, 4)	2	0,946	1,159
165	20000	(1, 4)	5	0,946	1,159
166	20000	(1, 4)	10	0,946	1,159
167	20000	(1, 4)	15	0,946	1,159
168	20000	(1, 4)	20	0,946	1,159
169	25000	(1, 1)	0	0,933	1,143
170	25000	(1, 1)	2	0,932	1,142
171	25000	(1, 1)	5	0,932	1,141
172	25000	(1, 1)	10	0,932	1,141
173	25000	(1, 1)	15	0,931	1,141
174	25000	(1, 1)	20	0,932	1,141

175	25000	(1, 2)	0	0,94	1,151
176	25000	(1, 2)	2	0,939	1,15
177	25000	(1, 2)	5	0,938	1,149
178	25000	(1, 2)	10	0,939	1,15
179	25000	(1, 2)	15	0,939	1,15
180	25000	(1, 2)	20	0,939	1,15
181	25000	(1, 3)	0	0,943	1,154
182	25000	(1, 3)	2	0,944	1,156
183	25000	(1, 3)	5	0,943	1,155
184	25000	(1, 3)	10	0,943	1,155
185	25000	(1, 3)	15	0,943	1,155
186	25000	(1, 3)	20	0,943	1,155
187	25000	(1, 4)	0	0,946	1,158
188	25000	(1, 4)	2	0,947	1,16
189	25000	(1, 4)	5	0,946	1,159
190	25000	(1, 4)	10	0,947	1,16
191	25000	(1, 4)	15	0,946	1,159
192	25000	(1, 4)	20	0,946	1,159
193	45000	(1, 1)	0	0,93	1,139
194	45000	(1, 1)	2	0,926	1,134
195	45000	(1, 1)	5	0,927	1,135
196	45000	(1, 1)	10	0,927	1,135
197	45000	(1, 1)	15	0,927	1,135
198	45000	(1, 1)	20	0,927	1,135
199	45000	(1, 2)	0	0,938	1,148
200	45000	(1, 2)	2	0,936	1,146
201	45000	(1, 2)	5	0,937	1,148
202	45000	(1, 2)	10	0,937	1,147
203	45000	(1, 2)	15	0,937	1,147
204	45000	(1, 2)	20	0,937	1,147
205	45000	(1, 3)	0	0,941	1,153
206	45000	(1, 3)	2	0,941	1,153
207	45000	(1, 3)	5	0,941	1,153
208	45000	(1, 3)	10	0,941	1,153
209	45000	(1, 3)	15	0,941	1,153
210	45000	(1, 3)	20	0,941	1,153
211	45000	(1, 4)	0	0,943	1,155
212	45000	(1, 4)	2	0,944	1,156
213	45000	(1, 4)	5	0,944	1,157
214	45000	(1, 4)	10	0,945	1,157
215	45000	(1, 4)	15	0,945	1,157
216	45000	(1, 4)	20	0,944	1,157
217	65000	(1, 1)	0	0,93	1,139
218	65000	(1, 1)	2	0,927	1,135
219	65000	(1, 1)	5	0,927	1,135

220	65000	(1, 1)	10	0,927	1,135
221	65000	(1, 1)	15	0,927	1,135
222	65000	(1, 1)	20	0,927	1,136
223	65000	(1, 2)	0	0,938	1,148
224	65000	(1, 2)	2	0,938	1,148
225	65000	(1, 2)	5	0,938	1,149
226	65000	(1, 2)	10	0,937	1,148
227	65000	(1, 2)	15	0,937	1,148
228	65000	(1, 2)	20	0,938	1,148
229	65000	(1, 3)	0	0,941	1,152
230	65000	(1, 3)	2	0,941	1,152
231	65000	(1, 3)	5	0,941	1,153
232	65000	(1, 3)	10	0,941	1,152
233	65000	(1, 3)	15	0,941	1,152
234	65000	(1, 3)	20	0,941	1,153
235	65000	(1, 4)	0	0,943	1,154
236	65000	(1, 4)	2	0,943	1,155
237	65000	(1, 4)	5	0,943	1,155
238	65000	(1, 4)	10	0,943	1,155
239	65000	(1, 4)	15	0,943	1,155
240	65000	(1, 4)	20	0,943	1,155

References

[1] P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, "A Comparison of Semantic Similarity Methods for Maximum Human Interpretability", vol. 1, 2019, pp. 1–4. DOI: 10.1109/AITB48515.2019.8947433.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. USA: Cambridge University Press, 2008, ISBN: 0521865719.

[3] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt, "Representation Learning for Very Short Texts Using Weighted Word Embedding Aggregation", *Pattern Recogn. Lett.*, vol. 80, no. C, pp. 150–156, Sep. 2016, ISSN: 0167-8655. DOI: 10.1016/j.patrec.2016.06.012.

[4] G. Kim and K. Cho, "Length-Adaptive Transformer: Train Once with Length Drop, Use Anytime with Search", *ArXiv*, vol. abs/2010.07003, 2020.

[5] O. Zafrir, G. Boudoukh, P. Izsak, and M. Wasserblat, "Q8BERT: Quantized 8Bit BERT", *ArXiv*, vol. abs/1910.06188, 2019.

[6] H. Gong, Y. Shen, D. Yu, J. Chen, and D. Yu, "Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Jul. 2020, pp. 6751–6761. DOI: 10.18653/v1/2020.acl-main.603. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.603.

[7] Y. Qi, D. Sachan, M. Felix, S. Padmanabhan, and G. Neubig, "When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation?", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 529–535. DOI: 10.18653/v1/N18-2084. [Online]. Available: https://www.aclweb.org/anthology/N18-2084.

[8] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms", in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 440–450. DOI: 10.18653/v1/P18-1041. [Online]. Available: https://www.aclweb.org/anthology/P18-1041.

[9] A. Rücklé, S. Eger, M. Peyrard, and I. Gurevych, "Concatenated p-mean Word Embeddings as Universal Cross-Lingual Sentence Representations", *ArXiv*, vol. abs/1803.01400, 2018.

[10] P. Turney and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics", *Journal of Artificial Intelligence Research*, vol. 37, pp. 141–188, Mar. 2010. DOI: 10.1613/jair.2934.

[11] A. L. O. Shahmirzadi and K. Younge, "Text Similarity in Vector Space Models: A Comparative Study", in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 659–666. DOI: 10.1109/ICMLA.2019.00120.

[12] V. Gupta, A. Kumar, P. Nokhiz, H. Gupta, and P. Talukdar, "Improving Document Classification with Multi-Sense Embeddings", in *24th European Conference on Artificial Intelligence - ECAI 2020*, Nov. 2020, pp. 2030–2037. DOI: 10.3233/FAIA200324.

[13] V. Mekala Dheeraj and Gupta, B. Paranjape, and H. Karnick, "SCDV : Sparse Composite Document Vectors using soft clustering over distributional representations", in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 659–669. DOI: 10.18653/v1/D17-1069. [Online]. Available: https://www.aclweb.org/anthology/D17-1069.

[14] V. Gupta, H. Karnick, A. Bansal, and P. Jhala, “Product Classification in E-Commerce using Distributional Semantics”, in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 536–546. [Online]. Available: https://www.aclweb.org/anthology/C16-1052.