ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК

Никитин Александр Викторович

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Иерархические тематические векторные представления слов в коллекциях текстов

Hierarchical topical embeddings in text collections

по направлению подготовки 09.04.04 Программная инженерия
образовательная программа «Системное программирование»

Студент

_____
А.В. Никитин

Научный руководитель:
д.ф.-м.н, проф.

_____
К.В. Воронцов

Москва, 2019

Аннотация

В данной работе рассматривается метод создания иерархических тематических векторных представлений по большим коллекциям текстов, то есть векторных представлений слов, построенных на основе тематической модели. В работе проведены эксперименты по подбору гиперпараметров алгоритма, применению полученных векторных представлений к различным прикладным задачам автоматической обработки текстов, описан метод построения иерархических векторных представлений, до этого не рассматривавшийся исследователями. В результате получены векторные представления, показывающие высокие результаты на различных прикладных задачах автоматической обработки текстов: классификация текста и определение близости слов (word-similarity). Полученные в ходе работы предобученные векторные представления, могут быть использованы другими исследователями и коммерческими организациями для решения задач автоматической обработки текстов.

Ключевые слова: тематическое моделирование, NLP, векторные представления, ARTM.

Annotation

This work discusses the method of creating hierarchical topical word embeddings of large collections of texts, that is, vector representations of words built on the basis of the topic model. In this work experiments have been conducted on the selection of hyperparameters of the algorithm, the application of the obtained vector representations to various problems of natural language processing. The method of constructing hierarchical word embeddings, which have not been considered by researchers, is described. As a result, obtained word embeddings showed high results on various applied tasks of natural language processing: text classification and word-similarity tasks. The pretrained word embeddings obtained in the course of this work, can be used by other researchers and commercial organizations to solve the problems of natural language processing.

Keywords: topic modelling, NLP, word embeddings, ARTM.

Содержание

1 Введение

Задачи автоматической обработки текстов (natural language processing, NLP) являются важными задачами машинного обучения. В последние годы появляется большое количество коммерческих приложений автоматической обработки текстов, среди которых:

- создание вопросно-ответных систем. В приложениях обеспечивающие автоматическую обратную связь пользователям в банковских, ритейл и других системах, возникает задача ответа на вопросы пользователя,
- анализ контента в социальных сетях. Среди приложений в этой области: мониторинг имиджа компании в социальной сети, классификация и кластеризация пользовательских постов, определение желтых заголовков и т.д.,
- создание диалоговых агентов. Агент, который в зависимости от задачи или может поддерживать диалог на общие темы, или обеспечивать движение по некоторому сценарию,
- создание поисковых систем. Поисковые системы (яндекс, google) используют алгоритмы автоматической обработки текстов для улучшения качества поисковой выдачи, обработки текстов и извлечения полезной информации,
- построение рекомендательных систем. В системах содержащих базу текстов (блоги, новостные сайты, поисковые системы и т.п.) зачастую реализованы рекомендательные системы, рекомендующие пользователям похожий или возможно интересный им контент.

Алгоритмы машинного обучения эффективно работают с многомерными векторными пространствами (метрическими или линейными), поэтому для работы с текстами необходимо уметь преобразовывать текст в некоторое векторное пространство. В данной работе мы рассмотрим методы создания векторных представлений из слов (embeddings) и рассмотрим их эффективность на различных прикладных задачах.

Построение эффективных представлений для слов на различных языках является важной задачей автоматической обработки текстов (Natural language processing). Качество решения многих задач так или иначе зависят от построения качественного векторного представления слов или всего текста. Среди таких задач:

- анализ тональности (задача определения эмоциональной окраски текста),
- классификация текста. Определение принадлежности текста к одному из нескольких классов по семантическим свойствам. Наиболее популярными задачами классификации текстов являются: Reuters Newswire Topic Classification, 20 newsgroup classification [18], UCI's spambase и многие другие,
- похожесть слов. Определение меры семантической похожести пары слов,
- текстовая аналогия. Подбор аналогичного слова к данному,

- морфологическая разметка (Part-Of-Speech tagging). Разметка слов текста по их морфологическим и грамматическим свойствам,
- выделение именованных сущностей (Named entity recognition). Выделение n-грамм из текста, которые являются определенными именами или названиями (имена, названия организаций, названия мест и т.д.),
- и многие другие;

В последние годы было разработано большое количество методов построения представлений слов: Word2vec[1], Glove[2], Fasttext[3] и многие другие. В качестве расширения идеи векторных представлений слов были разработаны векторные представления в биоинформатике, thought vectors, векторные представления текста (ELMO[4], Infersent[5] и многие другие), произвольных типов контента (StarSpace[6]).

Несмотря на высокие результаты, полученные методами на основе векторных представлений слов, существенным недостатком этих методов является покоординатная неинтерпретируемость векторов, представляющих слова или фрагменты текстов. В то же время, тематические модели также строят векторные представления слов, и также основаны на матричном разложении. Однако их преимущество в том, что векторные представления получаются интерпретируемыми. Это происходит благодаря вероятностной постановке задачи, в которой векторное представление является дискретным распределением по темам, и появляется возможность приписать каждой теме (т.е. каждой координате векторного представления) частотный словарь слов, выделив лексическое ядро темы. Более того, тематические модели позволяют строить иерархические представления, разделяя темы на подтемы, разреженные – определяя для любого слова или текстового фрагмента небольшое число тем, и мультимодальные – используя в качестве исходных данных не только слова, но и модальности нетекстовой природы (авторов, время, категории, пользователей, и т.д.). В работе [13] было показано, что тематические векторные представления способны решать задачи семантической аналогии слов не хуже, чем векторные представления типа word2vec, если их строить по данным о парной совместной встречаемости слов. В той же работе показано, что использование нескольких модальностей повышает качество модели. Поэтому в данной работе поставлены следующие **цели**:

- исследовать и описать методы построения векторных представлений слов на основе тематического моделирования,
- изучить и выбрать методы оценки качества векторных представлений, подготовить набор экспериментов для оценки качества полученных в ходе экспериментов,
- построить векторные представления на базе тематического моделирования по корпусу текстов Википедии,
- исследовать и провести эксперименты с иерархическими тематическими моделями для построения векторных представлений слов,
- сравнить результаты работы полученных векторных представлений с векторными представлениями, использующимися в индустрии.

Критериями успешности данных экспериментов является получение на тестовых задачах качества работы построенных векторных представлений сравнимого с активно используемыми векторными представлениями, такими как Word2Vec [1] и Glove. [2].

Данная работа организована следующим образом. В главе 2 написаны основные определения и понятия тематического моделирования, далее в разделе 3 этой работы описаны возможности библиотеки для тематического моделирования bigARTM, которая активно используется в проведенных экспериментах. Затем в главе 4 рассматриваются способы получения векторных представлений. В главе 5 описаны основные результаты и эксперименты данной работы. И в 6 подведены итоги и описаны дальнейшие направления исследований по теме работы. Наконец в Библиографии перечислены материалы, ссылки на которые встречаются по ходу статьи.

2 Определения и необходимые понятия

Тематическая модель – модель коллекции текстовых документов, которая определяет, к каким темам относится каждый документ коллекции. Алгоритм построения тематической модели получает на входе коллекцию текстовых документов. На выходе для каждого документа выдаётся числовой вектор, составленный из оценок степени принадлежности данного документа каждой из тем. Размерность этого вектора, равная числу тем, может либо задаваться на входе, либо определяться моделью автоматически [7].

Введем обозначения:

- W — конечное множество слов,
- D_p — конечное множество псевдодокументов,
- T — конечное множество тем (латентная переменная),
- p — функция вероятности.

Тематическая модель строится в следующих предположениях:

- $D_p \times W \times T$ — дискретная генеральная совокупность,
- гипотеза о существовании тем (каждое вхождение терма w в документ d, связано с некоторой темой из заданного конечного множества тем T)
- гипотеза мешка слов (в задаче обработки текстов зачастую полагают, что порядок термов в документе не важен. Замечание: мешок слов сам по себе зачастую используется в качестве векторного представления текста, но в данной работе он не будет рассматриваться),
- гипотеза о вероятностном представлении данных (текст – это набор слов полученных из заданного распределения),
- гипотеза условной независимости: $p(w|t,d) = p(w|t)$, таким образом вероятность появления термов в документе зависит от темы, но не зависит от документа.

Вероятностная тематическая модель записывается в виде: Вероятностная тематическая модель порождения текста:

$$p(w|d) = \sum_{t \in T} p(w|t,d)p(t|d) = \sum_{t \in T} p(w|t)p(t|d) = \sum_{t \in T} \phi_{wt}\theta_{td}$$

Таким образом в тематической модели даны: n_{dw} — частоты слов в документах $\Rightarrow \hat{p}(w|d) = \frac{n_{dw}}{n_d}$

И необходимо найти:

- $\phi_{wt} = p(w|t)$ — вероятности слов в темах

- $\theta_{td} = p(t|d)$ — вероятности тем в документах

Запишем критерий максимизации правдоподобия для задачи тематического моделирования:

$$\mathcal{L}(\Phi;\Theta) = \ln p\big((d,w)_{d\in D_p, w\in W};\Phi,\Theta\big) = \ln \prod_{w\in W}\prod_{d\in D_p}\big(p(w|d)p(d)\big)^{n_{dw}} =$$

$$= \sum_{w\in W}\sum_{d\in D_p} n_{dw}\ln\sum_{t\in T}\phi_{wt}\theta_{td} \to \max_{\Phi;\Theta}$$

$$\sum_{w\in W}\phi_{wt} = 1,\ \phi_{wt}\geq 0,\ \sum_{t\in T}\theta_{td} = 1,\ \theta_{td}\geq 0.$$

Метод максимального правдоподобия в библиотеке ARTM [9], записывается как

$$\sum_{w\in W}\sum_{d\in D_p} n_{dw}\ln\sum_{t\in T}\phi_{wt}\theta_{td} + R(\Phi,\Theta) \to \max_{\Phi,\Theta};$$

$$R(\Phi,\Theta) = \sum_{i=1}^{k}\tau_i R_i(\Phi,\Theta)$$

$$\sum_{w\in W}\phi_{wt} = \{0,1\},\ \phi_{wt}\geq 0,\ \sum_{t\in T}\theta_{td} = \{0,1\},\ \theta_{td}\geq 0.$$

Далее решается записанная оптимизационная задача относительно матриц Φ и Θ.

2.1 Методы оценки качества тематической модели

Для оценки качества тематической модели используются внутренние и внешние критерии качества.

В качестве **внутренних критериев** оценки качества применяются: перплексия, средняя когерентность, разреженность матриц Φ и Θ, различность тем, статистический тест условной независимости.

В качестве **внешних критериев** оценки качества тематической модели применяется большое количество методов: экспертная оценка, качество категоризации слов, качество ранжирования и т.д.., то есть оценка использования информации, полученной из тематической модели на смежных задачах автоматической обработки текстов (natural language processing).

Дадим важные для дальнейшего повествования определения.

Когерентностью (coherence) темы t по k лучшим словам называется величина:

$$PMI_t = \frac{2}{k(k-1)}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} PMI(w_i, w_j)$$

где w_i - i-е слово в порядке убывания ϕ_{wt}. $PMI(u,v) = ln(\frac{|D|N_{uv}}{N_u * N_v})$, N_{uv} – число документов, в которых слова u и v встретились в окне из 10 токенов, N_u – число документов, в которых

слово u встретилось как минимум один раз. Было показано, что когерентность близка к экспертным оценкам согласованности слов.

Перплексией языковой модели текста $p(w|d)$ называется

$$\Pi(D) = exp(-\frac{1}{n}\sum_{d\in D}\sum_{w\in d} n_{dw} ln(p(w|d)))$$

$n = \sum_{d\in D}\sum_{w\in d} n_{dw}$

2.2 Иерархическая тематическая модель

Расширением идеи тематического моделирования служит создание иерархии тем. Таким образом иерархия каждого более низкого уровня служит расширением тем верхнего уровня.

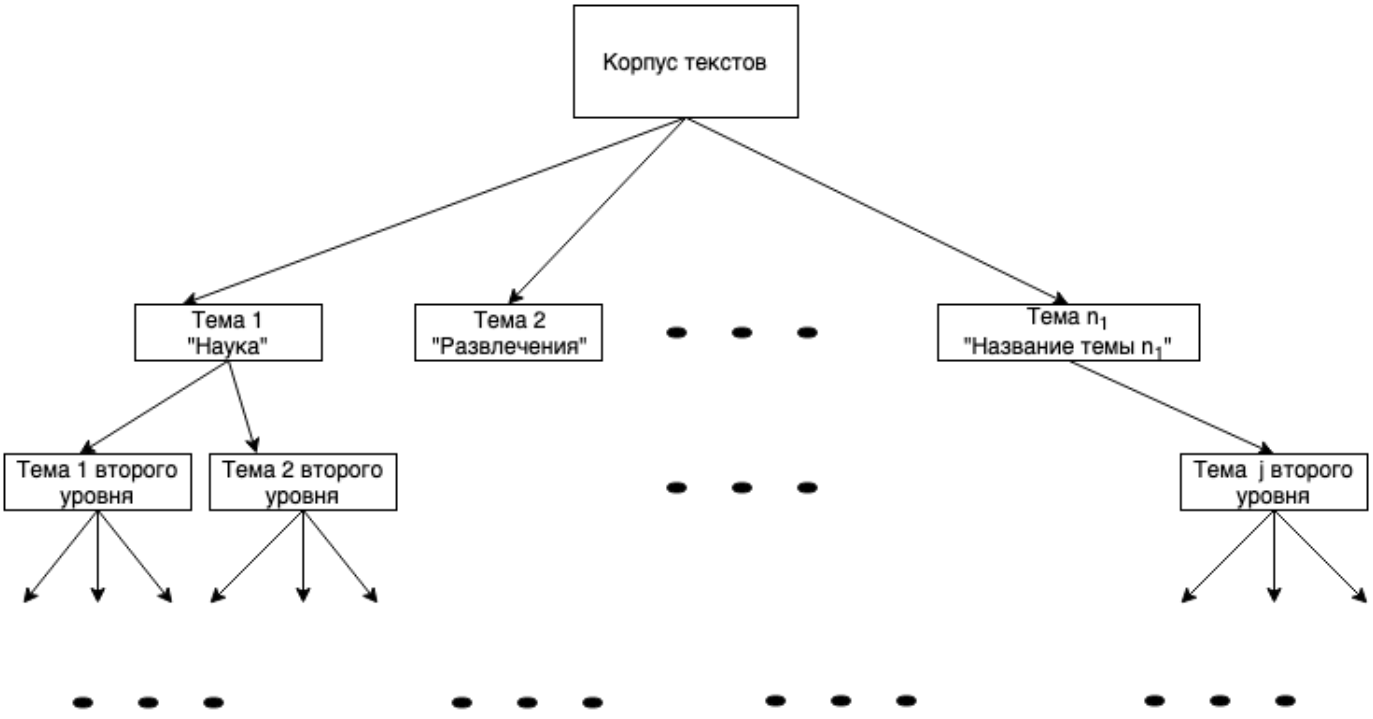

Рис. 1: Схематичное изображение иерархической тематической модели

Существует несколько подходов к построению иерархии. Во-первых по направлению построения иерархии различают **восходящие алгоритмы** (сначала документы разделяются на большое количество тем и похожие темы объединяются в одну) или нисходящие (когда сначала строятся наиболее общие темы, а потом более детальные, спускаясь вниз по дереву). Во-вторых, по количеству тематических моделей: можно строить модель для каждого

узла иерархии или строить одну модель для целого уровня. В реализации, используемой в bigARTM [9] иерархия строится сверху вниз с одной моделью на уровень.

Построение такой тематической модели может быть произведено с помощью добавления регуляризации матрицы Θ или матрицы Φ.

$$p(t|d) \approx \sum_{s \in S} p(t|s)p(s|d) = \sum_{s \in S} \psi_{ts}\theta_{sd} \Leftrightarrow \Theta^{parent} \approx \psi\Theta$$

Тогда новая оптимизационная задача будет иметь вид:

$$\sum_{d,w} n_{dw} \ln \sum_{s \in S} \varphi_{ws}\theta_{sd} + \lambda \sum_{d,t} \theta_{td}^{parent} \ln \sum_{s \in S} \psi_{ts}\theta_{sd} + R(\Phi, \Theta, \Psi) \to \max_{\Phi,\Theta,\psi}$$

Регулязатор Θ равносилен добавлению в модель еще одной модальности – тем t родительского уровня.

Иерархическая тематическая модель реализована в библиотеке bigARTM в классе hARTM, эта реализация и была использована для экспериментов в данной работе.

Аналогично можно добавить регуляризатор матрицы Φ.

$$p(w|t) \approx \sum_{s \in S} p(w|s)p(s|t) = \sum_{s \in S} \varphi_{ws}\tilde{\psi}_{st} \Leftrightarrow \Phi^{\text{ parent }} \approx \Phi\widetilde{\Psi}$$

В этом случае оптимизационная задача будет иметь вид:

$$\sum_{d,w} n_{dw} \ln \sum_{s \in S} \varphi_{ws}\theta_{sd} + \lambda \sum_{t,w} \varphi_{wt}^{parent} \ln \sum_{s \in S} \varphi_{ws}\tilde{\psi}_{st} + R(\Phi, \Theta, \widetilde{\psi}) \to \max_{\Phi,\Theta,\widetilde{\Psi}}$$

Подробное описание иерархического тематического моделирования дано в [8].

2.3 Выводы и результаты по главе

В данной главе были даны базовые определения тематического моделирования и иерархического тематического моделирования, описана математическая модель и даны предположения, в которых модель построена. Используя аппарат тематического моделирования в данной работе будут построены векторные представления слов.

3 BigARTM

В данной работе использовалась библиотека для построения тематического моделирования BigARTM [9]. BigARTM основывается на аддитивной регуляризации для матричных разложений, подробно описанном в [10]. Данная библиотека превосходит своих конкурентов (gensim, MALLET) по скорости работы и конфигурируемости параметров. Она активно используется в различных исследованиях и коммерческих разработках по тематическому моделированию. Отличительными чертами библиотеки относительно других существующих решений являются:

- параллельность,
- возможность для онлайн обучения (промежуточные результаты не хранятся в оперативной памяти, что позволяет использовать библиотеку для больших коллекций),
- расширяемое API. Для передачи данных используется формат protobuf, что позволяет использовать API из любых языков программирования,
- кросс-платформенность. Библиотека протестирована под разными компиляторами и операционными системами,
- открытый исходных код. Библиотека опубликована на сайте github, что позволяет использовать исправлять исходных код и легко обращаться к разработчикам с возможными проблемами.

В bigARTM реализованы также иерархические тематические модели, описанные в разделе 2.2 данной работы.

3.1 Онлайн и оффлайн обучение

Важной особенностью библиотеки bigARTM является наличие двух режимов для обучения тематических моделей online и offline. Онлайн алгоритм обучения тематических моделей был предложен в работе [11] и реализован в библиотеке bigARTM.

Оффлайн алгоритм производит несколько проходов по коллекции и один проход по документу (регулируется параметром $num_document_passes$, матрица Φ обновляется только один раз за проход (в конце прохода)). Как правило, оффлайн алгоритм используется при обучении тематических моделей на небольших коллекциях (тысячи документов).

Онлайн алгоритм производит один проход по всей коллекции (можно использовать обучение нескольких эпох для лучшей сходимости матрицы Φ), делая много проходов по одному документу. Делается несолько обновлений матрицы Φ за один проход по коллекции. Обычно данный алгоритм используется при обработке больших коллекций данных из-за лучшей производительности.

3.2 Гиперпараметры bigARTM

Для улучшения качества работы тематической модели в библиотеке bigARTM важно аккуратно подобрать гиперпараметры модели. В данной работе, для получения лучших векторных представлений слов:

- размерность векторного пространства слов,
- способ рассчета вектоного представления, по формуле Байеса из матрицы Φ или из матрицы Θ,
- количество проходов по коллекции (иногда, в данной работе этот параметр будет называться количеством эпох),
- количество проходов по одному документу (num_document_passes),
- алгоритм обучения тематической модели (онлайн или оффлайн),
- используемая регуляризация.

Ниже в данной работе показано, что подбор гиперпараметров сильно влияет на качество векторных представлений слов на внешнем тестировании и даны общие рекомендации для подбора гиперпараметров для обучения векторных представлений слов на коллекции текстов англоязычной Википедии.

3.3 Выводы и результаты по главе

В данной главе описана библитека тематического моделирования bigARTM, описан ее основной функционал, перечислены настраиваемые параметры. В данной работе bigARTM используется для построения тематических моделей в экспериментальной части.

4 Векторные представления слов

Распределенные векторные представления слов — это представления слов в векторном пространстве, где алгебраические операции над векторами имеют семантическую и/или синтаксическую интерпретацию. Представления (embeddings) являются объектами многомерного евклидова пространства: $emb(w) \in R^d$, где d обычно несколько сотен. Эти вектора подаются на вход другим моделям, которые и решают высокоуровневые задачи машинного обучения: анализ тональности, классификация текста, выделение именнованных сущностей и т.д. Разработка таких представлений является важной задачей в автоматической обработке текстов, так как композиции векторных представлений слов позволяют получить векторное представление текста и использовать его в решении многих задач автоматической обработки текстов, подробнее описано в работе [12]

Построение распределенных векторных представлений опирается на дистрибутивную гипотезу, звучащую в оригинале, как: "You shall know a word by the company it keeps. сформулированную J.R.Firth в 1957 году в работе Applications of general linguistics.

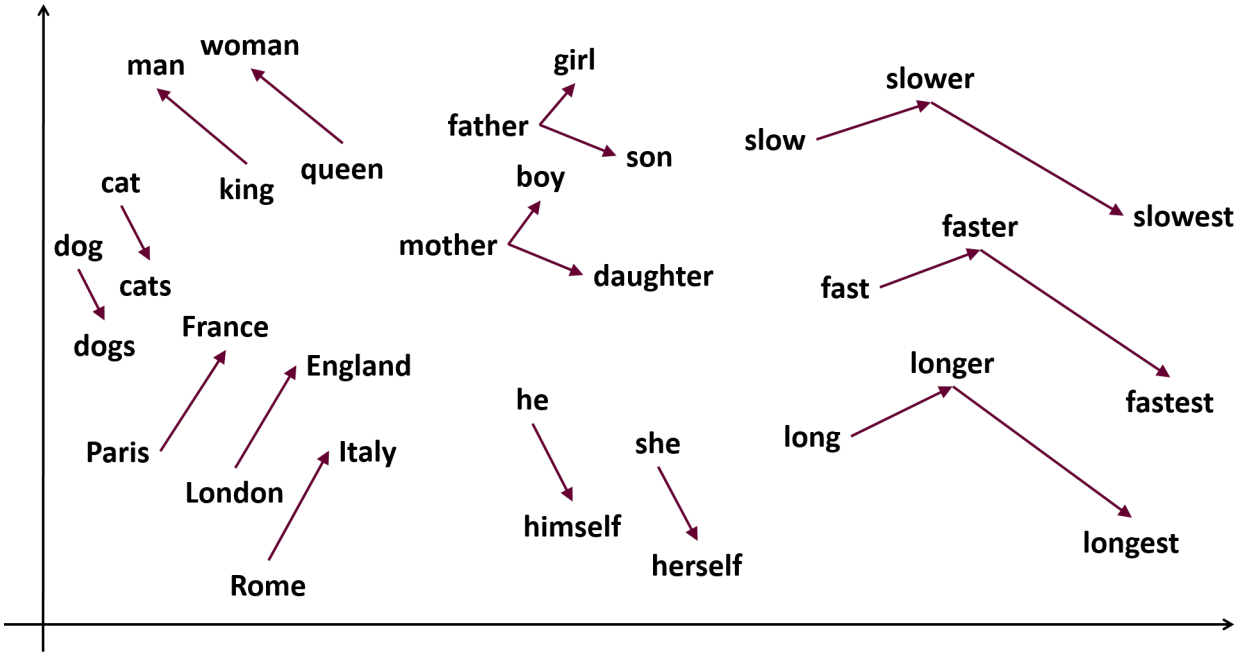

Рис. 2: Условное изображение векторного пространства слов `https://www.samyzaf.com/ML/nlp/nlp.html]`

Схематично векторные представления изображены на рисунке 2. Векторные представления слов обладают тем свойством, что алгебраические операции над векторами совпадают с семантическими операциями, например $v(\text{"}king\text{"})-v(\text{"}qween\text{"}) = v(\text{"}man\text{"})-v(\text{"}women\text{"})$. Эти свойства позволяют производить алгебраические операции над векторами, что может быть использовано, например, для получения векторного представления текста (как взвешенного среднего слов текста).

4.1 Glove

Одним из самых популярных методов получения векторных представлений слов является glove [2] Многие векторные представления построены на основе матрицы совстречаемости слов (coocurances). Обозначим матрицу совстречаемостей как X, тогда X_{ij} количество раз, сколько слово с индексом j встречалось в контексте слова i: $X_i = \sum_k X_{ik}$. Тогда обозначим $P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$, оценим отношение вероятностей совместной встречаемости слова с некоторым третьим словом как функцию от векторных представлений слов: $F(w_i, w_j, \widetilde{w_k}) = \frac{P_{ik}}{P_{jk}}$. Из общих соображений понятно, что F должна зависеть не от векторов w_i и w_j, а от их разницы. Таким образом, функция F примет вид: $F(w_i - w_j, \widetilde{w_k}) = \frac{P_{ik}}{P_{jk}}$, поскольку слева стоит функция от векторов, а справа скаляр, можем упростить F до

$$F((w_i - w_j)^T \widetilde{w_k}) = \frac{P_{ik}}{P_j k}$$

формула должна быть симметричной относительно замены слова на контекст и обратно, таким образом F должна быть гомоморфизмом из $(R, +)$ в $(R_{>0}, \times)$, таким образом

$$F((w_i - w_j)^T \widetilde{w_k}) = F(w_i^T \widetilde{w_k}) F(w_i^T \widetilde{w_k})$$

Решением данного уравнения является $F = exp$ или

$$w_i^T \widetilde{w_k} = log(P_{ik}) = log(X_{ik}) - log(X_i)$$

. Отсюда получим уравнение:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$$

Таким образом задача сводится к факторизации логарифма матрицы со-встречаемостей.

Идея Glove состоит в сведении задачи к взвешенной регрессии с весами $f(X_{ij})$ и функией стоимости

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$

где V – размер словаря. Подробнее про эксперименты с данной моделью и отношением с другими векторными представлениями слов написано в [2]

Векторные представления, полученные при помощи алгоритма glove активно используются для решения различных задач, например:

- кластеризация слов,
- морфологическая разметка слов,
- задачи классификации текстов (в качестве векторного представления текста используется взвешенное усредненное векторов слов текста).

В данной работе было решено использовать glove для сравнения качества с построенными векторными представлениями на ряде задач в качестве базового решения.

4.2 Векторные представления на основе тематического моделирования

Работы по получению векторных представлений на основе тематических моделей ведутся относительно недавно: в 2017 году была опубликована статья Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks [13], в которой авторами был предложен метод получения векторных интерпретируемых представлений слов, рассматриваемый в данной работе. Отдельным преимуществом данных векторных представлений является то, что их компоненты возможно интерпретировать исходя из тематической модели.

Идея построения векторных представлений при помощи тематической модели во многом похожа на метод Skip-Gram. Требуется научиться предсказывать вероятность слова u в локальном контексте v:

$$p(u|v) = \sum_{t \in T} p(u|t)p(t|v) = \sum_{t \in T} \phi_{ut}\theta_{tv}$$

В данной формулировке задачи вероятностная модель предсказывает не матрицу слова-документы, а матрицу совместных встречаемостей слов (совстречаемостей). Слова можно рассматривать как псевдодокументы: конкатенацию всех локальных контекстов слова v в корпусе. Обучение этой вероятностной модели (probabilistic word embeddings, PWE), эквивалентно стохастическому матричному векторному разложению.

Эта задача сводится к построению тематической модели, во-первых контексты слов интерпретируются как документы, во-вторых аддитивная регуляризация тематической модели может быть использована для увеличения разреженности матрицы Ф. Аддитивная регуляризация в данном случае имеет вид[13]:

$$R = -\tau \sum_{t \in T} \sum_{u \in W} \beta_u \ln \phi_{ut}$$

.

Используя подход multiARTM можно добавить новые модальности в полученную тематическую модель (например, в экспериментах данной работы в качестве еще одной модальности тематической модели используются категории текстов Википедии).

$$\begin{array}{l} \sum_{m \in M} \lambda_m \sum_{v \in W^0} \sum_{u \in W^m} n_{uv} \ln p(u|v) \to \max_{\Phi,\Theta} \\ \phi_{ut} \geq 0, \sum_{u \in W^m} \phi_{ut} = 1, \forall m \in M \\ \theta_{tv} \geq 0, \quad \sum_{t \in T} \theta_{tv} = 1 \end{array}$$

где λ_m – веса модальностей, W^m – словарь модальностей (при m=0 – модальность текста). Максимизация регуляризованного мультимодального правдоподобия с помощь online EM-алгоритма реализована в библиотеке bigARTM [9]. Важной особенностью реализации online EM-алгоритма является то что, в оперативной памяти компьютера достаточно хранить только матрицу Ф, что позволяет работать с большими коллекциями текстов.

Для улучшения векторного представления, возможно использовать иерархическую тематическую модель, описанную в 2.2. Используя в качестве модели иерархическую тематическую модель можно

- улучшить качество векторных представлений для использования на внешнем тестировании,
- улучшить интерпретируемость тем. Этот метод позволяет строить иерархию тем (в терминах тематического моделирования: иерархию компонент векторов)

4.3 Оценка качества векторных представлений слов

Существует несколько возможных подходов к оценке качества распределенных векторных представлений текста (внешних критериев качества):

- похожесть слов (word similarity). В порядке уменьшения метрики similarity нужно отранжировать слова [2]. Наиболее известными датасетами по задаче оценки похожести слов являются MEN [19], SIMLEX999 [20], WS353 [21],
- сравнение концепций (concepts comparison) [14],
- word analogies [15],
- paraphrases [16],
- вшешнее оценивание (multitask-оценивание). Оценка качества работы моделей, построенных поверх векторного представления для некоторых внешних задач. Этими задачами могут быть: определение тональности текста [17], классификация текстов [18], поиск похожих текстов, разметка слов по частям речи (Part Of Speech Tagging), распознавание именованных сущностей (Named Entity Recognition) и т.д..

4.4 Выводы и результаты по главе

В данной главе описаны различные методы построения векторных представлений слов, начиная от активно использующегося в данное время Glove, и заканчивая векторными представлениями на основе тематического моделирования. В данной главе предложено использовать в качестве тематической модели векторных представлений слов иерархическую тематическую модель, что позволяет получить иерархию тем построенных векторных представлений. В данной главе также описаны основные способы оценки качества векторных представлений слов, использующиеся в качестве критерия качества векторных представлений слов, построенных в данной работе.

5 Экспериментальные результаты

5.1 Технические детали

Для настройки и проведения экспериментов были осуществлены следующие этапы:

1. Настройка окружения в google.cloud
2. Парсинг и предобработка википедии
3. Обучение тематических моделей на корпусе википедии
4. Обучение тематических моделей по со-встречаемостям, выделение и сохранение распределенных векторных представлений слов
5. Анализ полученных результатов, сравнение различных методов построения распределенных векторных представлений слов

Ниже опишем более подробно все перечисленные этапы.

5.1.1 Настройка окружения в google.cloud

Для проведения экспериментов был использован облачный сервер на google.cloud n1-standard-8 (8 виртуальных CPU и 30 гигабайт оперативной памяти, для некоторых экспериментов память расширялась до 64 гигабайт) с 1Tb внешнего подключаемого диска SSD (для хранения развернутых дампов википедии и батчей в формате ARTM [9]).

5.1.2 Парсинг и предобработка википедии

Для парсинга корпуса википедии использовался актуальный дамп википедии на момент октября 2018 года [22]. При парсинге корпуса была произведена очистка от неинформативных и нечитаемых символов и извлечены категории статей (учитывая в том числе, иерархию категорий размеченных в Википедии).

5.1.3 Обучение тематических моделей на корпусе википедии

В качестве первого этапа построения распределенных векторных представлений по корпусу википедии были обучены классические тематические модели, были подобраны гиперпараметры модели и вручную проверена интерпретируемость тем. Тематическая модель была построена на двух модальностях: токены, извлеченные из текста и категории текста.

Были опробованы различные гиперпараметры модели, оценена перплексия и интерпретируемость для различных наборов гиперпараметров, проведено внешнее тестирование.

5.1.4 Обучение тематических моделей по со-встречаемостям, выделение и сохранение распределенных векторных представлений слов

На следующем этапе были построены распределенные векторные представления на основе тематических моделей. Векторные представления были получены двумя способами: по формуле Байеса из матрицы Φ и как столбцы матрицы Θ.

5.2 Обучение иерархических векторных представлений слов

В дальнейшем были проведены эксперименты с иерархическими векторными представлениями слов, построенными на базе иерархической тематической модели.

Подробно экспериментальные результаты описаны в разделе 5.3.

5.3 Анализ полученных результатов, сравнение различных методов построения распределенных векторных представлений слов

Полученные распределенные векторные представления слов были протестированы на нескольких задачах:

- определение похожести слов (word similarity) Была произведена проверка для датасетов MEN [19], SIMLEX999 [20], WS353 [21])
- анализ тональности текста [17]
- классификация текстов [18]

В качестве базовых решений, использовавшихся для сравнения брались предобученные вектора glove [2] размерности 300 и tf-idf.

5.3.1 IMDB Movie reviews dataset

IMDB Movie reviews [17] – это набор данных для задачи анализа тональности, основанный на отзывах к фильмам, оставленным пользователями на сайте IMDB (imdb.com). Этот набор данных состоит из 50000 отзывов к кинофильмам: 25000 отзывов для тренировочного множества и 25000 отзывов для тестового (также есть 50000 неразмеченных отзывов, которые зачастую рассматриваются для задач обучения без учителя).

Для оценки качества работы алгоритмов на данной выборке была использована точность (accuracy), так как рассматриваемая задача является задачей классификации на два класса и рассматриваемые классы сбалансированы.

5.3.2 20 newsgroups dataset

Данные для задачи классификации новостей на 20 групп (20 newsgroups, [18]) состоят из 18000 новостных постов по 20 различным темам. Посты разделены на 2 множества (тренировочное – содержащее 11314 постов и тестовое – содержащее 7532 поста). Разделение на тренировочные и тестовые множества было произведено по дате.

Распределение количества постов по классам новостей показана на Рисунке 3 и Рисунке 4. Для оценки качества решения данной задачи используется точность (accuracy).

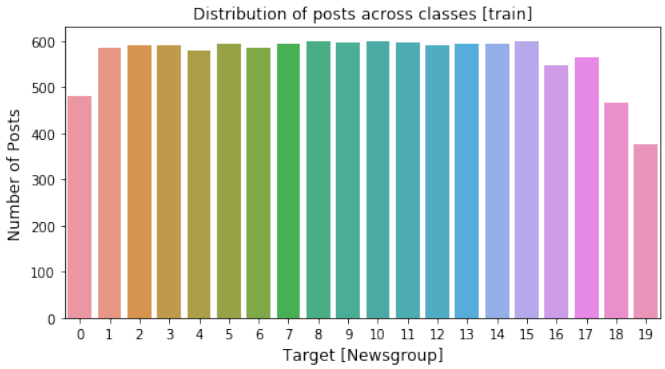

Рис. 3: Распределение постов по темам в датасете 20 newsgroups на тренировочном множестве

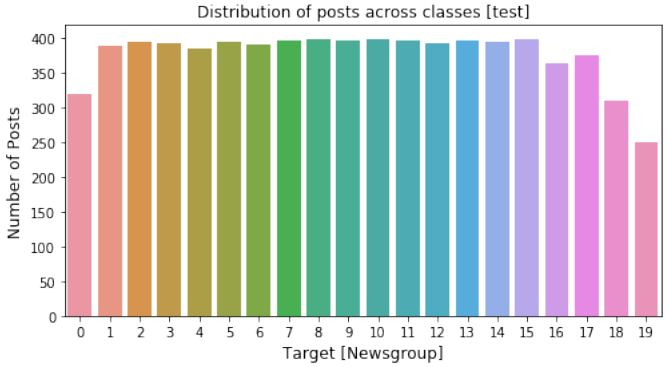

Рис. 4: Распределение постов по темам в датасете 20 newsgroups на тестовом множестве

5.3.3 Задача оценки похожести слов (word similarity)

Распространенным методом для оценивания качества векторных представлений слов является проверка на задаче оценки похожести слов (word similarity task). В таких задачах даны пары слов с их оценкой их похожести ассессорами (часто ассессоров просят отранжировать слова по похожести и отсюда получают оценочный коэффициент для их похожести).

В данной работе рассматриваются наборы данных: MEN [19], SIMLEX-999 [20], WS-353 [21].

Набор данных MEN был опубликован группой исследователей **CLIC**, в датасете даны 2 набора пар английских слов и оценка их похожести по шкале от нуля до десяти. Оценки получены с помощью краудсорсинговой платформы Mechanical Turk с использованием интерфейса CrowdFlower.

Набор данных **SIMLEX-999** состоит из пар слов, размеченных ассессорами, где разметка производилась в терминах синонимичности двух представленных слов (см. `https://fh295.github.io/simlex.html` с подробными инструкциями по разметке корпуса).

Набор данных **WordSim-353** состоит из пар слов, которые размечены по связанности (relatedness) по шкале от 0 (абсолютно несвязанные слова) до 10 (сильносвязанные или иден-

тичные слова). Более подробные инструкции по разметке датасета опубликованы в instructions.txt по ссылке `http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`.

5.4 Определение оптимальной размерности векторного пространства

При построении векторных представлений слов важно определиться с оптимальной размерностью векторного пространства. Для поиска оптимальной размерности были проведены эксперименты на задачах описанных в разделе 5.3.

Векторное представление	***IMDB***	***MEN***	***SIMLEX999***	***WS353***	***20news***
Glove 300d	**0.835**	**0.737**	0.371	0.543	0.730
Φ_{100}^{bayes}	0.704	0.575	0.198	0.533	0.622
Φ_{500}^{bayes}	0.762	0.657	0.274	**0.591**	0.708
Φ_{750}^{bayes}	0.766	0.630	0.268	0.542	0.723
Φ_{2000}^{bayes}	0.825	0.631	0.283	0.534	**0.733**

Таблица 1: Экспериментальные результаты полученных векторных представлений в зависимости от размера веткорного пространства

По полученным результатам видно, что лишь тематические векторные представления большой размерности могут получать качество соизмеримое с Glove, это объясняется тем, что тематические векторные представления гораздо более разрежены (информативные слова корпуса встречаются в небольшом количестве тем), нежели стандартные векторные представления. Таким образом, общей рекомендацией при подборе размерности векторного пространства является использование максимально доступной размерности и оценка разреженности полученных векторов.

5.5 Определение оптимального количества проходов по документам

Для достижения лучшей сходимости онлайн метода можно увеличивать количество проходов по одному документу (в bigARTM параметр *num_document_passes*). Можно проверить результаты на задачах, описанных в 5.3 в зависимости от количества проходов по документу при фиксированной размерности векторного пространства.

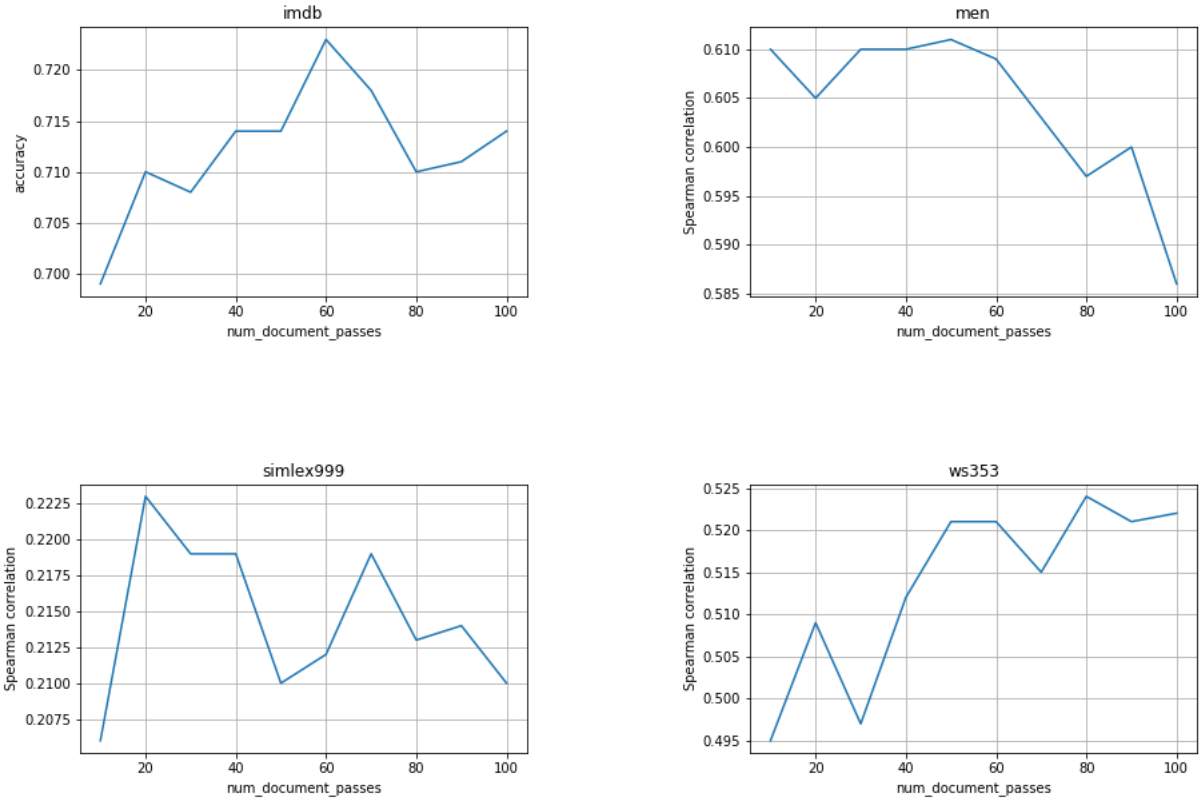

Рис. 5: Графики зависимости качества на внешних задачах в зависимости от количества проходов по документам

Экспериментальные результаты зависимости качества работы алгоритмов на внешнем тестировании показаны на графике 5. Видно, что оптимальный для сходимости алгоритма параметр *num_document_passes* лежит в районе 60.

5.6 Выбор матрицы для извлечения векторных представлений: ϕ или θ

Важным вопросом при построении тематической векторной модели слов является выбор матрицы для извлечения векторов: Phi (полученные по формуле Байеса) или $Theta$ (полученные из столбцов). Для выбора правильной матрицы были проведены аналогичные эксперименты с векторными представлениями полученными из одной и другой матриц, а также в случае с векторными представлениями, полученными линейной комбинацией матриц.

Интересный результат, полученный в ходе данных экспериментов, что матриц Θ быстрее сходится к оптимальному значению и результаты на новых эпохах не меняются. Однако результаты на внешнем тестировании при использовании матрицы Φ оказываются выше.

dataset	Φ_{bayes}	$Theta$
imdb	0.704	0.701
MEN	0.575	0.566
SIMLEX999	0.198	0.183
WS353	0.533	0.499
20newsgroups	0.622	0.628

Таблица 2: Сравнение векторных представлений полученных из Φ и Θ при фиксированной размерности векторного пространства 100

dataset	Φ_{bayes}	$Theta$
imdb	0.762	0.714
MEN	0.657	0.637
SIMLEX999	0.274	0.220
WS353	0.591	0.552
20newsgroups	0.708	0.690

Таблица 3: Сравнение векторных представлений полученных из Φ и Θ при фиксированной размерности векторного пространства 500

Из полученных результатов видно, что в общем случае векторные представления, полученные из матрицы Φ по формуле Байеса показывают лучшие результаты на все наборе рассматриваемых задач для размерности 500, аналогичные эксперименты были проведены для размерностей 100 и 750, результаты получились аналогичные. Стоит отдельно отметить, что аналогичные результаты получены и для больших размерностей векторного пространства.

5.7 Интерпретируемость компонент

Для того, чтобы показать интерпретируемость полученных векторных представлений слов, посчитаем когерентность и проведем экспертную оценку тем по наиболее частым словам (отсортируем по величине компоненты, соответствующей теме).

Для экспертной оценки качества интерпретируемости модели слова были отсортированы по величине одной из компонент и оценено их отнесение к одной из тем (в примере рассматривается тематическая модель на 500 тем после 9 итерации). Результаты отражены в таблице 4

Тема 1	Тема 2	Тема 3
Film_producers_from_New_York_(state)	Side-scrolling_role-playing_video_games	London_Films_films
American_women_cinematographers	Martial_arts_video_games	Films_shot_in_Scotland
People_from_the_Bronx	Spike_Chunsoft_video_games	Films_directed_by_Michael_Powell
Special_effects_people	Sonic_the_Hedgehog_video_games	narrabri
African-American_screenwriters	Video_games_scored_by_Manami_Matsumae	Films_directed$_{by}$_$Sidney_{M}organ$
American_people_of_Austrian-Jewish_descent	Tose_(company)_games	Films_shot_in_Wales
Horror_film_directors	Success_(company)_games	Films_shot_in_Hertfordshire
American_documentary_film_directors	Tiger_handheld_games	British_horror_films
Best_Directing_Academy_Award_winners	Resident_Evil_games	Amicus_Productions_films
Best_Cinematographer_Academy_Award_winners	Bangladeshi_romantic_comedy_films	British_horror_films

Таблица 4: Слова с наибольшими векторными компонентами для нескольких случайно выбранных тем (размерность векторного пространства 500. Тематическая модель обучена на 9 эпохах.)

Таким образом, видно, что при большом количестве эпох обучения алгоритма, наиболее погруженными в тематику получаются категории статей из Википедии.
Аналогично были проверены интерпретируемости векторных представлений полученных векторных представлений других размерностей. Результаты для размерности 2000 отражены в таблице 5

Тема 1	Тема 2	
Belarusian_female_tennis_players	quidam	Belarusian_footballers
'American_women_classical_pianists	Belarus_international_footballers	Curtis_Institute_of_Music_alumni
Sportspeople_from_Minsk	Juilliard_School_alumni	Belarusian_expatriate_footballers
assata	FC_Neman_Grodno_players	American_classical_pianists
FC_Dinamo_Brest_players	kyau	molko
Success_(company)_games	nyd	Belarusian_expatriates_in_Russia
American_classical_cellists	Belarusian_Premier_League_players	21st-century_conductors_(music)

Таблица 5: Слова с наибольшими векторными компонентами для нескольких случайно выбранных тем (размерность векторного пространства 2000. Тематическая модель обучена на 9 эпохах.)

Замечание: через подчеркивания обозначены категории статей из Википедии, использованные как дополнительная модальность. Для них в построенной модели также можно получить векторные представления.

Интересно, что если отсортировать слова по значениям некоторой компоненты, то можно наблюдать несколько подряд идущих тем (идущих тем не менее непрерывно). Например, в полученных векторных представлениях для размерности 2000, если отсортировать по одной из компонент слова и категории Википедии, получится список 6.

Порядок	Слово или категория
1	1980s_American_crime_television_series
2	1980s_American_drama_television_series
3	1990s_American_crime_television_series
4	American_crime_drama_television_series
5	Television_series_about_families
6	sunbathing
7	picnicking
8	1970s$_A merican_d rama_t elevision_s eries$
9	bushwalking
10	wandernadel
11	boardwalks
12	roundtrip
13	Television_series_by_MTM_Enterprises
14	geocaching
15	crabbing
16	bridleways
17	snowmobiling
18	backpacking
...	...

Таблица 6: Список слов с наибольшим значением 1999-й компоненты среди всех слов)

Видно, что в таблице 6 тема сериалов встречается среди первых 5 и плавно сменяется на тему "Активные виды отдыха". Если рассмотреть больше слов, то можно пронаблюдать аналогичные смены еще несколько раз.

Чтобы количественно оценить меру интерпретируемости тем, можно пронаблюдать зависимость когерентности от количества эпох. Для 25K документов и пятисот тем такая зависимость изображена на графике 5.7.

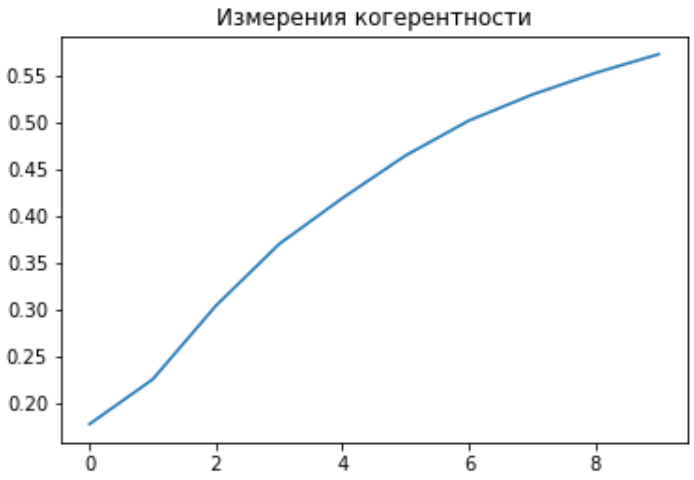

Рис. 6: Изменение когерентности в зависимости от количества эпох

Аналогично, для других размерностей векторного пространства, рассматриваемых в данной работе, когеренстность увеличивалась с количеством эпох.

5.8 Выбор оптимального количества эпох для онлайн-алгоритма

Известно, что обучение онлайн алгоритмом несколько эпох подряд ведет к лучшей сходимости матрицы Ф и как следствие лучшей тематической модели. Для определения оптимального количества эпох при обучении тематической модели были проведены эксперименты на разных размерностях векторного пространства (100, 500, 750, 2000) для задач, описанных в разделе 5.3. Результаты полученные в ходе экспериментов представлены в таблицах 7, 8, 9.

type	size	ep	imdb	news	MEN	SIMLEX999	WS353
tf-idf	-	-	**0.871**	**0.850**	-	-	-
glove	300	-	0.835	0.730	**0.737**	**0.371**	0.543
Φ_b	500	4	0.740	0.700	0.654	0.250	0.598
Φ_b	500	5	0.746	0.700	0.659	0.262	**0.601**
Φ_b	500	6	0.750	0.704	0.659	0.267	0.598
Φ_b	500	7	0.755	0.707	0.658	0.269	0.593
Φ_b	500	8	0.759	0.708	0.655	0.273	0.592
Φ_b	500	9	0.762	0.708	0.657	0.274	0.591

Таблица 7: Экспериментальные результаты для оценки качества построенных векторных представлений в зависимости от количества эпох (размерность 500)

type	size	ep	imdb	news	MEN	SIMLEX999	WS353
tf-idf	-	-	**0.871**	**0.850**	-	-	-
glove	300	-	0.835	0.730	**0.737**	**0.371**	0.543
Φ_b	750	4	0.755	0.710	0.647	0.252	**0.594**
Φ_b	750	5	0.763	0.713	0.649	0.259	0.585
Φ_b	750	6	0.768	0.718	0.645	0.263	0.576
Φ_b	750	7	0.769	0.719	0.642	0.266	0.573
Φ_b	750	8	0.769	0.721	0.634	0.270	0.558
Φ_b	750	9	0.766	0.723	0.630	0.268	0.542

Таблица 8: Экспериментальные результаты для оценки качества построенных векторных представлений в зависимости от количества эпох (размерность 750)

type	size	ep	imdb	news	MEN	SIMLEX999	WS353
tf-idf	-	-	**0.871**	**0.850**	-	-	-
glove	300	-	0.835	0.730	**0.737**	**0.371**	0.543
Φ_b	2000	4	0.811	0.724	0.643	0.272	**0.553**
Φ_b	2000	5	0.821	0.730	0.641	0.285	0.537
Φ_b	2000	6	0.825	0.733	0.631	0.283	0.534
Φ_b	2000	7	0.828	0.737	0.617	0.276	0.517

Таблица 9: Экспериментальные результаты для оценки качества построенных векторных представлений в зависимости от количества эпох (размерность 2000)

Можно заметить, что на размерности 2000 качество векторных представлений достигает качества соизмеримого с glove на задачах классификации текстов и на одной из задач определения похожести слов обгоняет векторные представления glove. Большая размерность векторных пространств обусловлена высокой разреженностью векторов построенных методом, описанным в данной статье.

Для размерности 2000 полученные результаты удобно рассмотреть графически (см Рис. 5.8)

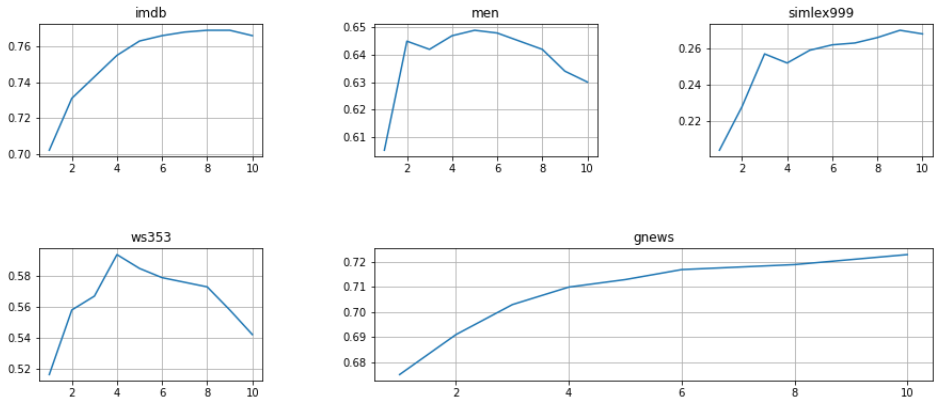

Рис. 7: Графики зависимости качества на внешних задачах в зависимости от количества эпох онлайн алгоритма

5.9 Эксперименты с иерархическими векторными представлениями

В ходе данных экспериментов были построены иерархические тематические модели и проведена оценка векторных представлений полученных из них на внешнем тестировании (задачи классификации текстов и оценки близости слов).

В экспериментах использовалась архитектура иерархической тематической модели изображенная на Рисунке 5.9.

Рис. 8: Архитектура иерархической тематической модели

Лучший результат иерархической модели был показан на самом нижнем уровне модели после 20 итераций оффлайн алгоритма обучения (**IMDB**: 0.815, **MEN**: 0.652, **SIMLEX999**: 0.261, **WS353**: 0.620, **google news**: 0.726), результаты для всех уровней показаны в таблице

10.

level	imdb	news	MEN	SIMLEX999	WS353
3	0.817	0.736	0.632	0.283	0.587
2	0.795	0.701	0.642	0.252	0.622
1	0.736	0.413	-	-	-

Таблица 10: Экспериментальные результаты для оценки качества построенных векторных представлений в зависимости от количества эпох (размерность 2000)

Таким образом для практических задач можно пользоваться векторными представлениями, полученными со второго и третьего уровня.

5.10 Выводы и результаты по главе

В данной главе описан процесс построения иерархических векторных представлений на основе текстовой коллекции статей Википедии, произведено сравнение полученных векторных представлений с Glove и показана интерпретируемость компонент иерархических векторных представлений слов.

6 Заключение

В ходе исследования удалось достичь поставленных целей: были построены векторные представления слов английского языка по корпусу Википедии [22], для полученных векторных представлений были проведены эксперименты на задачах анализа тональности (sentiment analysis), мультикслассовой классификации документов (20 news articles classification) и задаче похожести слов (word similarity) на задачах MEN, SIMLEX999, W353. Подбирая гиперпараметры алгоритма, удалось добиться качества близкого к векторным представлениям Glove, которые активно используется в задачах автоматической обработки текстов. Показана интерпретируемость компонент векторного представления, как тем в тематической модели. Описан и реализован алгоритм построения иерархического векторного представления на основе тематической модели, проведены эксперименты с иерархическими векторными представлениями. Экспериментальные результаты показывают практическую применимость построенных векторных представлений.

Дальнейшие исследования могут вестись по следующим направлениям:

1. эксперименты с онлайн-алгоритмом обучения тематических моделей и векторными представлениями, извлеченными из иерархий,
2. применение векторных представлений к другим прикладным задачам,
3. создание мультиязычных векторных представлений,
4. автоматическое именование тем (и как следствие автоматическое именование компонент векторного представления),
5. получение предобученных векторных представлений для основных языков и корпусов (проведение аналогичных этой работе экспериментов для других языков и корпусов),
6. исследование различных методов усреднения векторных представлений слов для получения векторного представления текстов,
7. эксперименты с другими архитектурами и гиперпараметрами иерархических тематических моделей.

Список литературы

[1] T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean, *Distributed representations of words and phrases and their compositionality* Proceedings of the 2013 conference on neural information processing systems (NIPS).

[2] Jeffrey Pennington, Richard Socher, Christopher Manning, *Glove: Global vectors for word representation* Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

[3] Joulin, Armand and Grave, Edouard and Bojanowski, Piotr and Douze, Matthijs and Jégou, Hérve and Mikolov, Tomas, *FastText.zip: Compressing text classification models* arXiv preprint arXiv:1612.03651 2016.

[4] Peters, Matthew E. and Neumann, Mark and Iyyer, Mohit and Gardner, Matt and Clark, Christopher and Lee, Kenton and Zettlemoyer, Luke *Deep contextualized word representations* Proc. of NAACL 2018.

[5] Conneau, Alexis and Kiela, Douwe and Schwenk, Holger and Barrault, Loïc and Bordes, Antoine, *Supervised Learning of Universal Sentence Representations from Natural Language Inference Data* Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.

[6] Wu, L. and Fisch, A. and Chopra, S. and Adams, K. and Bordes, A. and Weston, J., *StarSpace: Embed All The Things!* arXiv preprint arXiv:1709.03856 2017.

[7] K.V. Vorontsov et.al.
http://www.machinelearning.ru/wiki/index.php?title=Тематическое_моделирование

[8] N. A. Chirkova1, and K. V. Vorontsov, *Additive regularization for hierarchical multimodal topic modeling*, Machine Learning and Data Analysis, 2016 pp.187-200

[9] Документация библиотеки для тематического моделирования bigARTM. `https://bigartm.readthedocs.io/en/stable/api_references/python_interface/artm_model.html`

[10] Vorontsov K., Potapenko A. (2014) *Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization.* In: Ignatov D., Khachay M., Panchenko A., Konstantinova N., Yavorsky R. (eds) Analysis of Images, Social Networks and Texts. AIST 2014. Communications in Computer and Information Science, vol 436. Springer, Cham

[11] Matthew D. Hoffman, David M. Blei, Francis R. Bach, Online Learning for Latent Dirichlet Allocation, NIPS 2019.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, *Distributed Representations of Words and Phrases and their Compositionality*, NIPS 2013.

[13] Potapenko, Anna and Popov, Artem and Vorontsov, Konstantin, *Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks*, 2017

[14] Baroni, Marco and Dinu, Georgiana and Kruszewski, Germán, *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors*, 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference.

[15] Mikolov, T., Yih, W., Zweig, G. *Linguistic Regularities in Continuous Space Word Representations.* In HLT-NAACL (pp. 746–751), 2013.

[16] Task 1: Paraphrase and Semantic Similarity in Twitter at SemEval `http://alt.qcri.org/semeval2015/task1/`

[17] Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher, *Learning Word Vectors for Sentiment Analysis*, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (pp. 142-150), 2011.

[18] The 20 Newsgroups data set `http://qwone.com/~jason/20Newsgroups/`

[19] MEN dataset for word-similarity task `https://staff.fnwi.uva.nl/e.bruni/MEN`

[20] SIMLEX999 dataset for word similarity task `https://fh295.github.io/simlex.html`

[21] WS353 dataset for word-similarity task `http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`

[22] Wikipedia dumps `https://dumps.wikimedia.org`