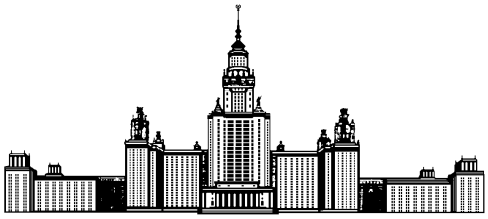

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Попов Артём Сергеевич

Регуляризация тематических моделей для векторных представлений слов

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф.-м.н., доцент, профессор РАН
К.В. Воронцов

Москва, 2017

Содержание

1 Введение

Тематическое моделирование (topic modeling) и векторные представления слов (word embedding) — два различных подхода к автоматическому выявлению смыслов в текстовых коллекциях. Оба подхода применяются для информационного поиска, классификации, категоризации, аннотирования, сегментации текстов.

В основе большинства моделей для построения векторных представлений слов лежит гипотеза дистрибутивности [9], которая заключается в том, что слова, встречающиеся в схожих контекстах, имеют близкие значения. Самыми популярными моделями для построения векторных представлений слов, основанными на гипотезе дистрибутивности, являются нейросетевые модели word2vec [6], позволяющие эффективно решать задачи поиска семантически близких слов и выявления аналогий в парах слов. В то же время, полученные моделями вектора являются плотными, а компоненты векторов неинтерпретируемыми.

В основе вероятностных тематических моделей лежит другая гипотеза — независимости, заключающаяся в том, что порядок слов в документах коллекции не важен для выявления его тематики. В вероятностных тематических моделях слова w и документы d описываются дискретными распределениями $p(w|t)$, $p(t|d)$ на множестве тем $t \in T$, то есть неотрицательными нормированными векторами размерности $|T|$. Компоненты в тематических векторных представлениях слов являются интерпретируемыми и разреженными, так как каждое слово относится, как правило, к небольшому числу тем. В то же время, большее значение в тематических моделях традиционно уделяется представлениям документов, а полученные представления слов сильно проигрывают по качеству решения задач близости и аналогий моделям, основанным на гипотезе дистрибутивности.

Целью данной работы является построение тематической модели, векторные представления слов которой позволяли бы хорошо решать задачи близости и аналогий, но оставались интерпретируемыми и разреженными. Для построения такой модели предлагается использовать тематическую модель ARTM [23], которая позволяет описывать многие тематические модели в рамках единого формализма и комбинировать их. Преимуществом тематической модели ARTM является наличие лёгкого способа учёта дополнительных метаданных, содержащихся в документе — модальностей, и механизм регуляризации, позволяющий накладывать дополнительные ограничения на решение. Для того, чтобы учесть в модели предположение гипотезы дистрибутивности, использу-

ется техника построения псевдо-коллекции [27], в которой каждый документ относится к конкретному слову и содержит все слова, встречающиеся с этим словом в локальных контекстах. В работе проведено сравнение предложенного метода с традиционными методами построения векторных представлений слов и тематических моделей.

В большинстве моделей, строящих векторные представления, для каждого слова строится два векторных представления, одно из которых соответствует ситуации, в которой слово является контекстом другого слова. Построение сразу двух представлний связано с упрощением обучения модели. При этом второе представление либо вообще не применяется в дальнейших приложениях, либо складывается с первым. В работе показано, как с помощью использования специальной инициализации EM-алгоритма, в тематических моделях можно строить единое представление для слова и контекста.

Также в работе показано, как с помощью аппарата модальностей модели ARTM, можно обощить технику построения псевдо-коллекции на случай мультимодальной коллекции. Проверена применимость предложенного метода, показано как метод может быть применён для построения векторных представлений элементов не основной модальности.

2 Модели векторных представлений слов

2.1 Определения и обозначения

Пусть D — коллекция текстовых документов, W — множество всех употребляемых в коллекции слов. Коллекция D представляется последовательностью слов $(w_1, \dots, w_N)$, N — длина коллекции. Каждый документ $d \in D$ представляется последовательностью слов $(w_1^d, \dots, w_{n_d}^d)$, n_d — длина документа. Число появлений слова w в документе d обозначим n_{dw}. Каждое слово $w \in W$ представляется бинарным вектором в формате one-hot-encoding длины $|W|$.

Контекстом слова w_i^d называется совокупность слов, отстоящих от данного слова не более чем на k позиций в документе, k — небольшое число от 2 до 10:

$$C(w_i^d) = \{w_{i-k}^d, \dots w_{i-1}^d, w_{i+1}^d, \dots w_{i+k}^d\}$$

Суммарное число появлений слова c во всех контекстах слова w обозначим n_{cw}. Согласно гипотезе дистрибутивности Харриса [9], слова, встречающиеся в схожих контекстах, имеют семантически близкие значения.

Задача построения векторных представлений слов (word embeddings) заключается в сопоставлении каждому слову $w \in W$ вектора $v_w \in \mathbb{R}^m$, $m \ll |W|$. Основное требование к отображению — соответствие близким по смыслу словам близких по расстоянию векторов.

В большинстве моделей, использующих гипотезу дистрибутивности, для каждого слова строится два векторных представления. Второе представление u_w соответствует слову w и используется при обучении модели в ситуации, когда необходимо задать векторные представления для слов в контекстах. Обозначим $V \in \mathbb{R}^{m \times |W|}$ и $U \in \mathbb{R}^{m \times |W|}$ — матрицы, столбцы которых соответствуют векторам слов и слов из контекстов соответственно. Согласно принятым обозначениям:

$$v_w = Vw \qquad u_w = Uw$$

Введём оператор softmax, который будет использоваться в дальнейшем для сопоставления векторам дискретных распределений. Оператор $\underset{t \in T}{\text{softmax}}(x_s)$ ставит в соответствие компоненте s вектора x компоненту s нормированного по всем $t \in T$ вектора $\exp(x)$:

$$\underset{t \in T}{\text{softmax}}(x_s) = \frac{\exp(x_s)}{\sum_{t \in T} \exp(x_t)}$$

2.2 Нейросетевые модели векторных представлений слов

Одна из первых работ по построению векторных представлений слов с использованием гипотезы дистрибутивности и нейронных сетей была предложена Бенджио в 2003 году [1]. Суть подхода заключается в моделировании совместного распределения всех слов в коллекции, используя предположение о том, что слово зависит не от всех предыдущих слов в корпусе, а только от некоторых k предшествущих (левый контекст слова):

$$p(w_1, \ldots, w_n) = \prod_{n=1}^{N} p(w_n | w_1, \ldots, w_{n-1}) \approx \prod_{n=1}^{N} p(w_n | w_{n-k}, \ldots, w_{n-1}) \tag{1}$$

Предложенная Бенджио нейронная сеть принимает на вход k слов $w_1, \ldots, w_k$ в формате one-hot-encoding, ставит им в соответствие их векторные представления, производит их конкатенацию и передаёт её на вход следующему слою. Выход сети — дискретное

вероятностное распределение на множестве всех слов словаря $p(w|w_1, \ldots, w_k)$:

$$L(V, H, b, d) = \sum_{n=1}^{N} \ln p(w_n|w_{n-k}, \ldots, w_{n-1}) \to \max_{V,H,b,d} \tag{2}$$

$$x = d + (v^T_{w_{n-1}}, \ldots v^T_{w_{n-k}}), \quad d \in \mathbb{R}^{mk \times 1} \tag{3}$$

$$p(w_n|w_{n-1}, \ldots, w_{n-k}) = \underset{w_n \in W}{\text{softmax}}\,(b + Htanh(x)), \quad b \in \mathbb{R}^{|W| \times 1},\ H \in \mathbb{R}^{|W| \times mk} \tag{4}$$

Модель обучается с помощью стохастического градиентного спуска и алгоритма обратного распространения ошибки, но из-за огромного количества настраеваемых параметров, наличия нелинейного преобразования и нормировки на последнем слое, обучение сети происходит очень медленно. Более простая, но гораздо более эффективная архитектура, получившая название CBOW, была предложена в 2013 году в статье [8]. Вместо моделирования статистической модели языка введена другая постановка задачи — предсказание слов по их контекстам. Вместо конкатенации векторов в модели используется их сумма, полностью отсутствуют нелинейности:

$$L(U, V) = \frac{1}{N} \sum_{n=1}^{N} \ln p(w_n|C(w_n)) \to \max_{U,V} \tag{5}$$

$$p(w_n|C(w_n)) = \underset{w_n \in W}{\text{softmax}} \left(\sum_{w_k \in C(w_n)} v^T_{w_n} u_{w_k} \right) \tag{6}$$

В этой же статье была предложена модель Skip-gram, в основе которой лежит идея противоположная идее CBOW — построение модели, предсказывающей по слову слова из его контекста. Skip-gram для входного слова w выдаёт дискретное распределение на множестве слов — вероятность появления каждого слова в контексте слова w. Функционал, который используется для обучения модели:

$$L(U, V) = \frac{1}{N} \sum_{n=1}^{N} \sum_{w_k \in C(w_n)} \ln p(w_k|w_n) \to \max_{U,V} \tag{7}$$

$$p(w_k|w_n) = \underset{w_k \in W}{\text{softmax}}(v^T_{w_k} u_{w_n}) = \underset{w_k \in W}{\text{softmax}}(\langle v_{w_k}, u_{w_n} \rangle) \tag{8}$$

Функционал модели можно записать проще, используя обозначение n_{cw}:

$$L(U, V) = \sum_{w \in W} \sum_{c \in W} n_{cw} \ln p(w|c) \to \max_{U,V} \tag{9}$$

Рассмотренные модели обучаются с помощью стохастического градиентного спуска и алгоритма обратного распространения ошибки. На каждом шаге обучения необходимо вычислять функцию $\exp(\langle v_w, u_c \rangle)$ для каждого слова w, что не позволяет эффективно

обучать модель на данных большого размера. В статье [6] предложено два способа модификации модели: hierarchical softmax и negative sampling. В Skip-gram hierarchical softmax (SGHS) вычисления softmax происходит с помощью бинарного дерева, в котором каждый лист соответствует конкретному слову. В этой модели не требуется обучать векторные представления для слов из контекстов, но необходимо вычислять векторные предствления для каждой внутренней вершины дерева. Вероятность $p(w|c)$ вычисляется с помощью произведения по всем элементам пути от корня до листа, соответствующего w:

$$p(w|c) = \prod_{j \in path(w)} \sigma([next\ node\ is\ left\ child]\langle v'_j, v_c\rangle) \tag{10}$$

Функция $[\cdot]$ возвращает 1, если выражение истинно, и -1 иначе. Функция сигмоида $\sigma(x) = 1/(1+e^{-x})$. За path(w) обозначены вершины, лежащие на пути от корня до вершины, соответствующей w. Исходное дерево строится исходя из частот появления слов в коллекции — частотные слова располагаются ближе к корню. Такая модификация вычисления softmax позволяет сократить сложность каждой итерации с $O(|W|)$ до $O(\ln|W|)$.

В Skip-gram negative sampling (SGNS) вместо моделирования вероятности $p(w|c)$, моделируются более простая вероятность $p(\text{"yes"}|w,c)$ — вероятность встретить пару (w,c) в коллекции. Модель обучается по всем парам, встретившимся в обучающей выборке. На каждой итерации обучения случайно из распределения $p(w,c) = p(w)^{0.75}p(c)$, где $p(w), p(c)$ — частотные оценки вероятности встретить слово в коллекции, семплируются пары слов — отрицательные примеры (negative samples). Так как эти пары слов с большой вероятностью не встречались в исходной коллекции, вероятность встретить их вместе в коллекции необходимо понижать. Если обозначить множество негативных примеров за D', итоговый функционал модели записывается так:

$$\mathcal{L}(U,V) = \sum_w \sum_c n_{cw} \log p(\text{“yes”}|w,c) + \sum_{(w,c)\in D'} \log(1 - p(\text{“yes”}|w,c)) \longrightarrow \max_{U,V} \tag{11}$$

$$p(\text{“yes”}|w,c) = \sigma(\langle u_w, v_c\rangle) \tag{12}$$

Модели CBOW, Skip-gram, SGHS и SGNS составляют семейство моделей, известное под именем word2vec[1]. Модели SGHS и SGNS являются state-of-the-art в построении

[1]word2vec — лишь название программного продукта, реализующего алгоритмы CBOW, Skip-gram, SGHS и SGNS, а не название модели

word embeddings для самых различных задачах. Помимо рассмотренных вариаций обучения у моделей есть множество гиперпараметров. Например, с помощью процедуры subsampling можно с некоторой вероятностью отбрасывать слова при обучении, а процедура dynamic window позволяет на каждой итерации выбирать случайное значение размера окна для контекста.

2.3 Матричные разложения для векторных представлений слов

Одним из подходов к поиску векторных представлений слов являются матричные разложения. Одним из традиционных подходов является использование сингулярного разложения. В статье [14] показано, что при правильно сделанной предобработке коллекции, сингулярное разложение Shifted PPMI матрицы может давать результаты, сопоставимые с моделью word2vec. В этом случае, вектора и контексты задаются следующим образом:

$$SPPMI_{wc} \equiv \max(PMI_{wc} - \ln k, 0) \tag{13}$$

$$SPPMI_{wc} = U_d^* \Sigma_d V_d^* \qquad V = U_d^* \sqrt{\Sigma_d} \qquad U = V_d^* \sqrt{\Sigma_d} \tag{14}$$

Модель Glove [17], предложенная в 2014 году, представляет собой низкоранговое разложение матрицы логарифмов совместной встречаемости слов. В модели оптимизируется взвешенный квадратичный функционал:

$$\sum_{w \in W} \sum_{c \in W} f(n_{cw})(\langle v_w, u_c \rangle + b_w + \tilde{b}_c - \ln n_{cw})^2 \to \max_{V,U,b,\tilde{b}} \tag{15}$$

$$f(x) = \begin{cases} (x/x_{max})^{\varepsilon}, & \text{если } x < x_{max} \\ 1, & \text{если } x > x_{max} \end{cases} \tag{16}$$

Функция $f(x)$ нужна для штрафа слишком больших счётчиков, x_{max} и ε подбираются эмпирически, $b_w, \tilde{b}_c \in \mathbb{R}^m$ — вектора сдвига. Модель обучается методом AdaGrad по элементам входной матрицы. Качество полученных векторных представлений сопоставимо с качеством моделей SGHS и SGNS.

Интересно то, что модели SG и SGNS также можно трактовать как матричные разложения. В статье [13] показано, что модель SGNS можно интерпретировать, как разложение SPPMI матрицы. Модель Skip-gram представляет собой минимизацию суммы

взвешенных KL-дивергенций:

$$\sum_{c\in W}\sum_{w\in W} n_{cw}\ln p(w|c) = \sum_{c\in W} n_c \sum_{w\in W}\frac{n_{cw}}{n_c}\ln p(w|c) \to \max_{V,U} \Leftrightarrow$$
$$\Leftrightarrow -\sum_{c\in W} n_c \sum_{w\in W}\frac{n_{cw}}{n_c}\left(\ln p(w|c) - \ln\frac{n_{cw}}{n_c}\right) = \sum_{c\in W} n_c KL(\tilde{p}(w|c) \parallel p(w|c)) \to \min_{V,U} \quad (17)$$

2.4 Оценивание векторных представлений слов

Основное требование к построенным векторным представлениям слов — близость векторов, соответствующих семантически/синтаксически близким словам. Способом проверки этого свойства является задача близости (word similarity task) [19]. Каждая задача близости привязана к конкретном датасету, состоящему из списка пар слов и близости между каждой парой, измеренной асессорами. Для каждой пары слов из датасета вычисляется близость между построенными векторами. Итоговое значение задачи — корреляция Спирмена между модельными и асессорскими близостями.

В статье [8] был предложен новый способ оценивания векторных представлений, основанный на возможности проведения интерпретируемых алгебраических операции над векторами. Пример такой операции:

$$v_{\text{король}} - v_{\text{мальчик}} + v_{\text{девочка}} \approx v_{\text{королева}}$$

В статье был предложен способ оценивания интерпретируемости операций — задача аналогий (word analogy task). Задача аналогий привязана к конкретному датасету, состоящему из списка четвёрок слов. Для каждой четвёрки известно, что первое слово находится в таком же семантическом отношении со вторым словом, в каком третье слово находится в отношении с четвёртым (например, отношение столица — страна). Таким образом, для векторов, соответствующим четырём словам, должно выполняться тождество:

$$v_{2\text{ слово}} - v_{1\text{ слово}} + v_{3\text{ слово}} \approx v_{4\text{ слово}}$$

Задача заключается в нахождении по первым трём словам четвёртого слова:

$$w = \arg\min_{w\in W} \rho(v_{2\text{ слово}} - v_{1\text{ слово}} + v_{3\text{ слово}}, v_w) \quad \rho \text{ — функция близости}$$

Итоговое значение задачи — точность нахождения четвёртого слова. Заметим, что в задаче аналогий не обязательно использовать арифметические операции. В [12] показана эффективность использования других мер соотношения между четырьмя словами.

3 Тематические модели для обработки текстовых коллекций

3.1 Тематическая модель PLSA и её обобщения

Другим подходом для обработки текстовых коллекций является вероятностное тематическое моделирование. В основе тематического моделирования лежит гипотеза независимости, эквивалентная предположению, что порядок слов в документах коллекции не важен для выявления тематики, то есть тематику документа можно узнать даже после произвольной перестановки слов.

Предполагается, что появление каждого термина w в документе d связано с некоторой скрытой переменной t из конечного множества тем T. Тогда коллекция D представляет собой выборку троек (d, w, t), взятых независимо из дискретного распределения $p(d, w, t)$ на множестве $D \times W \times T$.

Гипотезой условной независимости называется предположение, что появление слов по теме t не зависит от документа:

$$p(w|t) = p(w|d, t)$$

С учётом этой гипотезы и формулы полной вероятности, а также обозначений $p(w|t) = \phi_{wt}$ и $p(t|d) = \theta_{td}$, тематическая модель коллекции представляется в виде:

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d) = \sum_{t \in T} \phi_{wt}\theta_{td} \tag{18}$$

Этой вероятностной моделью описывается порождение коллекции D по известным распределениям $p(w|t)$ и $p(t|d)$. Построение тематической модели — обратная задача: по коллекции D необходимо восстановить породившие коллекцию распределения.

Простейшей вероятностной тематической моделью является модель PLSA [10]. В PLSA для построения модели (18) максимизируется логарифм правдоподобия при ограничениях нормировки и неотрицательности:

$$L(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in W} n_{dw} \ln \sum_{t \in T} \phi_{wt}\theta_{td} \;\to\; \max_{\Phi, \Theta} \tag{19}$$

$$\sum_{w \in W} \phi_{wt} = 1,\ \phi_{wt} \geq 0; \quad \sum_{t \in T} \theta_{td} = 1,\ \theta_{td} \geq 0 \tag{20}$$

Стандартным методом решения этой задачи является EM-алгоритм — итерационный процесс, состоящий из двух шагов: E-шаг (expectation) и M-шаг (maximization).

На E-шаге по текущим параметрам ϕ_{wt} и θ_{td} (начальное приближение — нормированные неотрицательные случайные вектора) вычисляются вероятности $p(t|d,w)$ для всех $t \in T$, $w \in W$, $d \in D$. На M-шаге при фиксированных вероятностях $p(t|d,w)$ вычисляются новые приближения для параметров ϕ_{wt} и θ_{td}.

Задачу построения тематической модели можно трактовать как задачу поиска приближения матрицы частот F произведением двух неизвестных стохастических (с нормированными и неотрицательными столбцами) матриц Φ и Θ:

$$F \approx \Phi\Theta \tag{21}$$

$$F = (\hat{p}(w|d))_{|W|\times|D|} = (n_{dw}/n_d)_{|W|\times|D|} \qquad \Phi = (\phi_{wt})_{|W|\times|T|} \qquad \Theta = (\phi_{wt})_{|T|\times|D|} \tag{22}$$

Задача поиска матричного разложения матрицы частот F имеет бесконечно много решений: $F = \Phi\Theta = \Phi(S^{-1}S)\Theta = \Phi'\Theta'$ (с условием, что Φ' и Θ' — стохастические). Таким образом задача тематического моделирование — некорректно поставленная. Общим методом решения таких задач является регуляризация — введение разумных дополнительных ограничений на решение задачи.

Модель ARTM [23] является обобщением модели PLSA. Для построения модели (18) максимизируется сумма логарифма правдоподобия и r дополнительных критериев $R_i(\Phi,\Theta)$, называемых регуляризаторами, с коээфицентами регуляризации τ_i. Ещё более общей моделью является мультимодальная ARTM[24], позволяющая учитывать различные модальности, встречающиеся в документе (например, время написания или автора). Пусть M — множество модальностей. Для каждой модальности строится своя матрица Φ^m, $\Phi = \bigcup\limits_{m\in M} \Phi^m$. Итоговый оптимизируемый функционал строится как сумма взвешенных с коеффициентами α_m логарифмов правдоподобий для каждой модальности и регуляризаторов:

$$L(\Phi,\Theta) = \sum_{m\in M} \alpha_m \sum_{d\in D} \sum_{w\in W^m} n_{dw} \ln \sum_{t\in T} \phi_{wt}\theta_{td} + \sum_{i=1}^{r} \tau_i R_i(\Phi,\Theta) \to \max_{\Phi,\Theta} \tag{23}$$

$$\sum_{w\in W} \phi_{wt} = 1,\ \phi_{wt} \geq 0; \quad \sum_{t\in T} \theta_{td} = 1,\ \theta_{td} \geq 0 \tag{24}$$

Эта задача также решается с помощью EM-алгоритма, изменения по сравнению с PLSA претерпевают формулы M-шага.

Большинство способов обобщения модели PLSA основано на использовании байесовских методов. Самой известной модификацией является модель LDA [3], в основе которой лежит предположение, что векторы документов $\theta_d \in \mathbb{R}^{|T|}$ и векторы тем $\phi_t \in \mathbb{R}^{|T|}$ порождаются из распределения Дирихле. Обучение модели заключается в

поиске апостериорных распределений на параметры Φ и Θ. Существуют реализации модели, основанные на вариационном байесовском выводе[21], и реализации, основанные на семплировании Гиббса [26].

Одной из отличительных особенностей тематических моделей является интерпретируемость. Каждая тема t характеризется набором терминов с самой большой вероятностью $p(w|t)$. Существует автоматическа мера измерения интерпретируемости тем в тематических моделях — когерентность темы (topic coherence), оценки которой хорошо коррелируют с асессорскими оценками интерпретируемости [2]. Оценка когерентности для темы t — среднее значение элементов подматрицы $PPMI$ матрицы, соответствующих k словам, имеющим наибольшее ϕ_{wt}:

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

Когерентность модели — средняя когерентность по всем темам.

3.2 Тематические модели дистрибутивной семантики

Существуют расширения тематических моделей, использующие гипотезу дистрибутивности. Модель Word Network Topic Model (WNTM) [27] была придумана для работы с короткими текстами социальной сети Twitter. WNTM — модель LDA, которая обучается не по исходной коллекции документов, а по перенарезанной псевдо-коллекции. Каждый документ псевдо-коллекции соответствует определённому слову словаря и состоит из всевозможных слов, встречавшихся в контекстах этого слова (рис. 1). Модель обучается с помощью семплирования Гиббса. Авторами показано, что модель WNTM превосходит LDA при работе с короткими текстами и работает наравне с LDA на текстах большого размера.

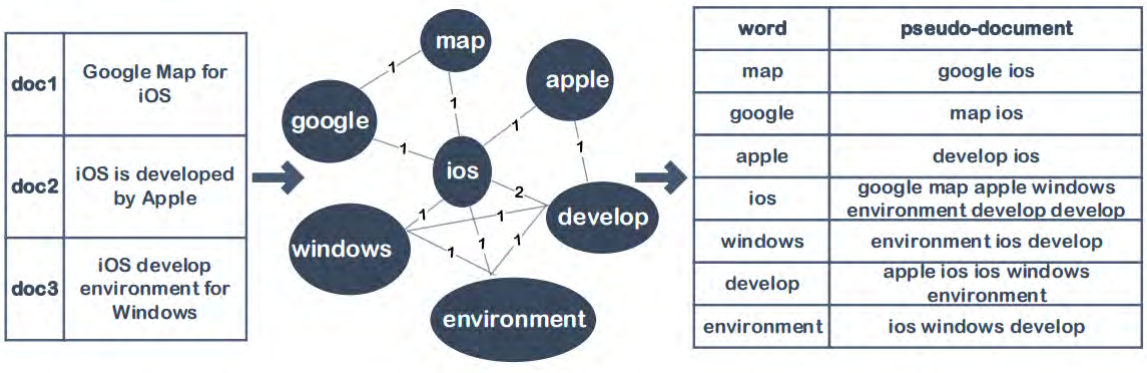

Рис. 1: Составление по коллекции псевдо-коллекции

Другой тематической моделью, основанной на гипотезе дистрибутивности является модель Biterm Topic Model (BTM) [4], также придуманная для работы с короткими текстами. Модель BTM моделирует вероятность совместного появления слова и контекста, B — множество всех пар слов и слов из их контекстов:

$$\ln L(B) = \sum_{(w,c)\in B} \ln p(w,c) \to \max_{\Phi,\pi} \tag{25}$$

$$p(w,c) = \sum_{t\in T} \phi_{wt}\phi_{ct}\pi_t \tag{26}$$

$$\sum_{w\in W} \phi_{wt} = 1,\ \phi_{wt} \geq 0; \quad \sum_{t\in T} \theta_{td} = 1,\ \theta_{td} \geq 0; \quad \sum_{t\in T} \pi_t = 1,\ \pi_t \geq 0 \tag{27}$$

В отличие от множества других моделей, в модели BTM нет явного задания представлений для слов, находящихся внутри контекстов. Как и WNTM, модель обучается с помощью семплирования Гиббса.

4 Тематические модели для векторных представлений слов

4.1 PLSA на псевдо-коллекции для векторных представлений слов

Несмотря на то, что в ходе обучения в тематических моделях строятся представления для слов, тематические модели никогда не рассматривались как средство для построения качественных векторных представлений сами по себе. Основное использование тематических моделей для построения векторных представлений слов — их соединение с моделью word2vec в одну гибридную модель[5][22].

Рассмотрим модель PLSA, обученную по псевдо-коллекции из модели WNTM:

$$L(\Phi,\Theta) = \sum_{c\in W}\sum_{w\in W} n_{cw} \ln \sum_{t\in T} \phi_{wt}\theta_{tc} \ \to \ \max_{\Phi,\Theta} \tag{28}$$

$$\sum_{w\in W} \phi_{wt} = 1,\ \phi_{wt} \geq 0; \quad \sum_{t\in T} \theta_{tc} = 1,\ \theta_{tc} \geq 0 \tag{29}$$

Различие между оптимизируемыми функционалами этой модели и Skip-gram (формула 9) заключается лишь в формировании вероятности $p(w|c)$. Модель не требует огромного количества вычислительных затрат связанных с вычислением softmax, как модель Skip-gram, так как нормировка по всем словам происходит лишь один раз за

проход всей коллекции. Таким образом, можно выдвинуть гипотезу, что с помощью тематической модели 28-29 можно строить качественные векторные представления, способные решать задачи близости и аналогий.

Предложенный подход легко модифицировать, заменив n_{wc} на любую другую меру совстречаемости двух слов, например $\ln n_{wc}$ или $PPMI_{wc}$. Однако, как показывают эксперименты, модификации такого вида практически не улучшают качество исходной модели. Другой модификацией является добавление регуляризаторов в функционал модели, например, с помощью регуляризатора разреживания можно повышать долю нулей в матрицах Φ и Θ.

Обучение модели можно проводить с помощью традиционного оффлайнового EM-алгоритма, онлайнового EM-алгоритма и гибридной модификации оффлайнового алгоритма, в которой матрица Θ на каждой итерации инициализируется случайно, а затем делается несколько проходов внутри итерации без изменения матрицы Φ до сходимости Θ. Эксперименты показывают, что наиболее эффективным вариантом является комбинирование онлайнового алгоритма (небольшое число итераций в начале) и гибридного алгоритма (дообучение полученной модели). Использование алгоритмов, не хранящих матрицу Θ, может в будущем сократить вычислительные затраты на составление псевдо-коллекции и позволит не использовать подсчитанную заранее матрицу совстречаемостей, а обрабатывать коллекцию последовательно.

Помимо выбора способа обучения модели, необходимо выбрать как с помощью полученных матриц Φ и Θ задавать векторные представления слов. В качестве векторного представления для слова $w \in W$ v_w для задачи близости предлагается использовать величины $p(t|w)$ или $\log(p(t|w) + 1)$, которые можно получить из матрицы Θ или с помощью теоремы Байеса из матрицы Φ. В качестве меры близости предлагается использовать скалярное произведение, которое показывает себя в экспериментах лучше остальных известных функций близости. Для задачи аналогий в качестве векторного представления для слова $w \in W$ предлагается использовать величину $\log(p(t|w)+\alpha)$, где $\alpha \in (0, 1]$. Заметим, что сложение логарифмов вероятностей, гораздо более естественная операция чем сложение вероятностей. В качестве меры близости предлагается использовать косинусную близость, которая по результатам экспериментов показывает себя в задаче аналогий лучше чем скалярное произведение.

4.2 Тематические модели для поиска разложения симметричной матрицы

Рассмотрим задачу поиска матричного разложения симметричной матрицы F. Так как матрица симметричная, разложение можно искать в виде:

$$F \approx V^T V \tag{30}$$

Предположим, что после обучения алгоритма было получено матричное разложение:

$$F \approx V^T S S^{-1} V = V'^T V' \tag{31}$$

Так как мы искали разложение в симметричном виде:

$$(V^T S)^T = S^{-1} V^T \Rightarrow S^T = S^{-1}$$

Рассмотрим косинусную близость между двумя векторами, полученными из второго разложения:

$$\cos(v'_w, v'_c) = \frac{\langle v'_w, v'_c \rangle}{\langle v'_w, v'_w \rangle \langle v'_c, v'_c \rangle} = \frac{v_c'^T v'_w}{v_w'^T v'_w v_c'^T v'_c} = \frac{v_c^T S S^{-1} v_w}{v_w^T S S^{-1} v_w v_c^T S S^{-1} v_c} = \cos(v_w, v_c)$$

Таким образом, если искать разложение в виде (30), любые трансорфмации разложения вида (31) не изменяют косинусную близость между векторными представлениями. Аналогичное свойство можно показать для скалярного произведения.

В тематическом моделировании разложение (30) неприменимо, так как матрицы должны быть нормированы, но можно искать разложение в виде:

$$F \approx \Phi \Phi^{bayes} \tag{32}$$

$$\Phi^{bayes} \in R^{|T| \times |W|} \qquad \Phi^{bayes}_{tw} = p(t|w) = \frac{\phi_{wt} p(t)}{p(w)} \tag{33}$$

Покажем, что для достижения такого разложения при решении задачи оффлайновым EM-алгоритмом достаточно правильно инициализацировать матрицы на первой итерации.

Теорема 1. *Пусть $n_{wc} = n_{cw}$, начальная инициализация матрицы $\Theta = \Phi_0^{bayes}$, где Φ_0 — начальная инициализация матрицы Φ. Тогда, матрицы Φ и Θ, полученные при решении задачи* (28)-(29) *оффлайновым EM-алгоритмом, удовлетворяют соотношению $\Theta = \Phi^{bayes}$.*

Доказательство. Пусть на первой итерации $\Phi = \Phi_0$, $\Theta = \Phi_0^{bayes}$. Рассмотрим итерационные формулы EM-алгоритма. Покажем симметричность вероятности $p(t|c,w)$ после E-шага:

$$p(t|c,w) = \frac{\phi_{wt}\theta_{tc}}{\sum_{s\in T}\phi_{ws}\theta_{sc}} = \frac{\phi_{wt}\phi_{ct}p(t)p(c)}{p(c)\sum_{t\in T}\phi_{ws}\phi_{cs}p(s)} = \frac{\phi_{wt}\phi_{ct}p(t)}{\sum_{t\in T}\phi_{ws}\phi_{cs}p(s)} =$$
$$= \frac{\theta_{tw}p(w)\phi_{ct}p(t)}{p(t)p(w)\sum_{t\in T}\theta_{sw}\phi_{cs}p(s)(p(s))^{-1}} = \frac{\theta_{tw}\phi_{ct}}{\sum_{t\in T}\theta_{sw}\phi_{cs}} = p(t|w,c)$$

На M-шаге:

$$\phi_{wt} = \frac{n_{wt}}{n_t} \qquad n_{wt} = \sum_c n_{cw}p(t|c,w) \qquad n_t = \sum_w n_{wt}$$

$$\theta_{tc} = \frac{n_{tc}}{n_c} \qquad n_{tc} = \sum_w n_{cw}p(t|c,w) \qquad n_c = \sum_t n_{tc}$$

В силу выявленной симметричности:

$$n_{tc} = \sum_w n_{cw}p(t|c,w) = \sum_w n_{wc}p(t|w,c) = n_{ct}$$

Тогда для элементов матрицы Θ верно тождество:

$$\theta_{tc} = \frac{n_{tc}}{n_c} = \frac{n_{ct}n_t}{n_c n_t} = \phi_{ct}\frac{n_t}{n_c} = \phi_{ct}\frac{p(t)}{p(c)} = \Phi_{tc}^{bayes}$$

■

Таким образом, предложенная схема инициализации позволяет обеспечить разложение (32). Интересно, что построение модели PLSA при такой инициализации эквивалентно построению модели BTM.

Теорема 2. *Пусть $n_{wc} = n_{cw}$, начальная инициализация матрицы $\Theta = \Phi_0^{bayes}$, где Φ_0 — начальная инициализация матрицы Φ. Тогда, задача* (28)-(29) *эквивалента задаче* (25)-(27).

Доказательство. Распишем совместную вероятность появления терминов w и c в модели (28)-(29) при выбранной инициализации:

$$p(w,c) = p(w|c)p(c) = \sum_{t\in T}\phi_{wt}\theta_{tc}\frac{n_c}{\sum_{c'}n_{c'}} = \sum_{t\in T}\phi_{wt}\phi_{ct}\frac{n_c n_t}{n_c\sum_{c'}n_{c'}} =$$
$$= \sum_{t\in T}\phi_{wt}\phi_{ct}\frac{n_t}{\sum_{t'}\sum_{c'}n_{t'c'}} = \sum_{t\in T}\phi_{wt}\phi_{ct}p(t) = \sum_{t\in T}\phi_{wt}\phi_{ct}\pi_t$$

Мы получили формулу (26), которой определялась модель BTM. ■

4.3 Мультимодальные тематические модели дистрибутивной семантики

Схему построения псевдо-коллекции документов из модели WNTM нетрудно обобщить на случай мультимодальной коллекции. Будем считать, что в коллекции можно выделить связи двух видов. Первый вид связи — внутремодальные, считаются по словам одной модальности. Слова связаны внутремодальной связью, если одно из них находится в контексте другого. Второй вид связи — внешнемодальные, считаются по словам из разных модальностей. Слова связаны внешнемодальной связью, если встречаются в одном документе. Очевидно, что второй тип связи гораздо хуже моделирует зависимости между словами, чем первый, поэтому каждое вхождение слов в один документ нужно учитывать с небольшим весом. Также при такой схеме можно строить псевдо-документы не для всех слов коллекции, а только для слов основной модальности.

doc1	Google map for iOS	Jobs
doc2	iOS is developed by Apple	Jobs
doc3	iOS develop enviroment for Windows	Gates

word	1 modality	2 modality
google	map ios	jobs
map	google ios	jobs
ios	google map develop apple develop enviroment windows	jobs jobs gates
develop	apple ios ios windows enviroment	jobs gates
apple	ios develop	jobs
enviroment	ios develop windows	gates
windows	ios develop enviroment	gates
jobs	google map ios ios develop apple	-
gates	ios develop windows enviroment	-

Рис. 2: Составление по мультимодальной коллекции псевдо-документов

Предложенная схема не только позволяет учесть мультимодальную структуру документа и использовать её для улучшения качества модели, но и строить векторные представления для токенов модальностей, содержащихся в документе. Для того, чтобы токены всех модальностей содержались в одном пространстве, векторное представление v_w необходимо строить на основе матрицы Θ.

5 Вычислительные эксперименты

Эксперименты по сравнению предложенного подхода и существующих проводились на коллекции английских статей Википедии (дамп 2016-01-13). Сначала из коллекции были удалены 25 стоп-слов, затем словарь был сокращён до оставшихся 100000 самых частотных слов. Далее по коллекции создавалась псевдо-коллекция, размер окна локальных контекстов — 5, коэффициент сабсемплинга — 10^5. При составлении псевдо-

документов коллекции были удалены пары слов, встречавшиеся меньше чем 5 раз. Использование динамического окна не давало серьёзных улучшений никакому из подходов при первоначальных экспериментов, поэтому не использовалось при составлении псевдо-коллекции. Отказ от использования сабсемплинга приводил к большом росту псевдо-коллекции и построение модели на персональном компьютере было невозможным.

5.1 Оптимальные параметры в задаче близости

Цель этого эксперимента — нахождение наилучших параметров для модели (28)-(29) в задаче близости. Параметрами являются:

1. Преобразование исходных матриц Φ и Θ
2. Функция близости
3. Преобразование исходных счётчиков n_{cw}
4. Алгоритм обучения модели

Для оценивания качества решения задачи близости использовались датасеты WordSim353 [18], дополнительно разделенный на две части — similarity и relatedness [20], MEN [7] и Mechanical Turk [25].

Исследуем представления $p(w|t)$, $p(t|w)$ и $log(p(t|w) + 1)$ и меры близости $cos(w, u), \langle w, u\rangle, -hel(w, u) = -\frac{1}{\sqrt{2}} \parallel \sqrt{w} - \sqrt{u} \parallel_2$. Качество измерялось на датасете WordSim, для обучения использовался оффлайновый EM-алгоритм. Для каждой модели проверялось три размерности (300, 400, 500), результат представленный в таблице — наилучшее из трёх значений.

Таблица 1: Качество ARTM на WordSim в зависимости от представления и функции близости

	cos	dot	-hel
$p(w\|t)$	0.59	0.5	0.25
$p(t\|w)$	0.61	**0.65**	0.58
$\ln(p(t\|w) + 1)$	0.61	**0.65**	0.6

Стандартная для SNGS косинусная мера близости и стандартное для модели LDA отрицательное расстояние Хеллингера работают хуже чем скалярное произведение(таблица 1). Единственная ситуация, когда косинусная мера близости работает луч-

ше скалярного произведения — представление $p(w|t)$. Это связано с тем, что представление $p(w|t)$ не является нормированным. В случаях, когда представление нормировано по норме l1, делать перенормировку по норме l2 не нужно. Заметим, что при вычислении скалярного произведения между представлениями $p(t|w)$ и $\log(p(t|w)+1)$ имеют значения только темы, имеющие ненуевые $p(t|w)$. В дальнейших экспериментах на задаче близости для предложенного подхода будет использоваться представление $p(t|w)$, а в качестве функции близости скалярное произведение.

Тематическую модель необязательно строить на счётчиках совстречаемостей слов, вместо них можно использовать любую другу меру взаимной близости слов. Однако, никаких заметных преимуществ по сравнению с использованием счётчиков совстречаемости другие меры совстречаемости не дают (таблица 2). При обучении всех моделей в этом эксперименте использовался гибридный оффлайновый EM-алгоритм, в котором на каждом шаге матрица Θ инициализируется случайно.

Таблица 2: Качество ARTM в зависимости от данных для построения модели

	WordSim Sim.	WordSim Rel.	WordSim	Bruni MEN	Radinsky M. Turk
n_{cw}	**0.709**	**0.635**	**0.654**	0.658	**0.607**
n_{cw}/n_c	0.642	0.58	0.576	0.679	0.517
$PPMI_{cw}$	0.694	0.618	0.636	0.685	0.577
$\max(PPMI_{cw}, \ln 5)$	0.697	**0.635**	0.648	**0.696**	0.6
$PPMI_{cw}/\sum_w PPMI_{cw}$	0.637	0.514	0.554	0.666	0.54

Сравним несколько стратегий обучения предложенного подхода EM-алгоритмом:

1. Оффлайновый алгоритм (40 итераций)
2. Оффлайновый алгоритм для симметричного разложения (40 итераций)
3. Онлайновый алгоритм (5 итераций)
4. Гибридный алгоритм (30 итераций)
5. Комбинированный алгоритм — онлайновый алгоритм (2 итерации) + гибридный алгоритм (30 итераций)

Число итераций выбиралось исходя из сходимости качества задач близостей. На некоторых датасетах качество начинает медленно падать с ростом итераций, это связано с увеличением числа нулей в исходных матрицах. Процесс останавливался, если

на четырёх датасетах качество переставало расти. Схема комбинированного применения онлайнового алгоритма и гибридного показывает себя лучше остальных подходов (таблица 3). Симметричное разложение эквивалентно по качеству несимметричному разложению, что подтверждает теорию о том, что различные представления для слов и контекстов в этой задаче не так уж необходимы.

Таблица 3: Сравнение алгоритмов обучения на задаче близости

Model	WordSim Sim.	WordSim Rel.	WordSim	Bruni MEN	Radinsky M. Turk
Оффлайновый	0.71	0.62	0.65	0.67	0.59
Оффлайновый + симметр.	0.71	0.62	0.64	0.66	0.59
Онлайновый	0.715	**0.675**	**0.685**	0.668	0.638
Гибридный	0.709	0.635	0.654	0.658	0.607
Комбинированный	**0.723**	**0.675**	0.682	**0.672**	**0.642**

Одна из попыток улучшения модели заключалась в использовании негативных семплов при обучении модели. На каждой итерации во все псевдо-документы добавлялось некоторое число случайных слов с отрицательным весами (по аналогии с моделью SGNS). Однако, такая схема приводила только к ухудшению модели, а также сильному замедлению обучения.

5.2 Сравнение моделей на задаче близости

Таблица 4: Сравнение моделей на задаче близости

Model	Метрика	WordSim Sim.	WordSim Rel.	WordSim	Bruni MEN	Radinsky M. Turk
SGNS	cos	**0.752**	0.632	0.666	**0.745**	**0.661**
SVD of PPMI	cos	0.711	0.648	0.672	0.236	0.616
LDA	hel	0.53	0.455	0.474	0.583	0.483
ARTM (комб.)	dot	0.723	**0.675**	**0.682**	0.672	0.642

Сравним на задаче близости предложенный подход с тематической моделью LDA и с традиционными моделями для получения word embeddings SGNS и сингулярным разложением SPPMI. Для каждой из базовых моделей будем использовать общепри-

нятую в литературе меру близости: косинусную для SGNS и SVD, -hel для векторных представлений $p(t|w)$ модели LDA.

Предложенный способ построения тематических моделей, сильно превосходит по качеству LDA (таблица 4) и сопоставим по качеству с моделями SGNS и SVD.

5.3 Оптимальные параметры в задаче аналогий

Аналогично задаче близости, исследуем влияние параметров модели (28)-(29) на решение задачи аналогий. Будем исследовать следующие параметры:

1. Преобразование исходных матриц Φ и Θ
2. Функция близости
3. Схема построения аналогий
4. Алгоритм обучения модели

Как и в предыдущем эксперименте на задаче близости, замена исходных счётчиков на более сложные величины не даёт прироста качества, поэтому детальный анализ этой модификации не вошёл в работу. Для оценивания качества решения задачи аналогий использовались датасеты Google и Microsoft Research (MSR).

Исследуем четыре различных представления: $p(w|t)$, $p(t|w)$, $log(p(t|w) + 1)$, $log(p(t|w) + \alpha)$. Заметим, что если $\alpha \in (0, 1]$, то при последнем представлении учитываются не только нулевые $p(t|w)$. Исследуем две функции близости: $cos(w, u)$ и $\langle w, u \rangle$. Рассмотрим стандартную схему построения аналогий через алгебраические опперации. Исследовать качество будем на части датасета Google — Google semantics.

Таблица 5: Качество ARTM на Google semantics в зависимости от представления и функции близости

	cos	dot
$p(w\|t)$	0.11	0.045
$p(t\|w)$	0.113	0.001
$\ln(p(t\|w) + 1)$	0.128	0.003
$\ln(p(t\|w) + \alpha)$	**0.343**	0.005

Лучший результат у косинусной близости на представлении $\ln(p(t|w) + \alpha)$ (таблица 5). Использование логарифма в операциях сложения выглядит достаточно естественным. Уменьшение сглаживающей константы с 1 до α связано с тем, что в задаче анало-

гий играют роль нулевые компоненты, их важно учитывать. Скалярное произведение неприменимо в задаче аналогий.

Помимо вариаций представлений и функций близости, можно также менять постановки задачи аналогий. Например вместо сложения и вычитания представлений логично в случае вероятностей делать умножение и деление:

$$w = \arg\min_{w \in W} \rho\left(\frac{p(t|2\text{ слово})p(t|3\text{ слово})}{p(t|1\text{ слово}) + \alpha},\ p(t|w)\right) \quad \rho \text{ — функция близости}$$

Такая схема, несмотря на её идейную близость к сложению логарифмов, показывает плохое качество — 0.015. Рассчитывание близости только по ненулевым координатам не приводит к улучшению результата.

Рассмотрим разные вариации EM-алгоритма для обучения (таблица 6). Как и в задаче близости, лучше всего себя показывают гибридная и комбинированная схемы. Успех схем основанных на инициализации Θ случайными значениями на каждой итерации можно объяснить тем, что действуя таким образом мы добавляем в модель больше случайности. EM-алгоритм очень сильно зависит от начальной инициализации, подобные модификации дают более широкое пространство для поиска решений.

Таблица 6: Сравнение алгоритмов обучения на задаче аналогий

	Google semantic	Google syntactic	MSR
Оффлайновый	0.324	0.268	0.142
Оффлайновый + симметр.	0.339	0.310	0.160
Онлайновый	0.315	0.310	0.153
Гибридный	**0.366**	0.323	0.166
Комбинированный	0.318	**0.370**	**0.208**

5.4 Сравнение моделей на задаче аналогий

Сравним на задаче аналогий предложенный подход с тематической моделью LDA и с традиционными моделями для получения word embeddings SGNS и сингулярным разложением SPPMI. Для всех моделей будем использовать косинусную меру близости. Для модели LDA используем представление $\ln(p(t|w) + \alpha)$.

По результатам сравнения видим (таблица 7), что тематические модели дистрибутивной семантики превосходит традиционную тематическую модель LDA. Результаты для предложенного подхода сопоставимы с представлениями из SVD разложения

Таблица 7: Сравнение моделей на задаче аналогий

	Google semantic	Google syntactic	MSR
LDA	0.1	0.186	0.1
ARTM (гибрид.)	0.366	0.323	0.166
ARTM (онлайн. + гибрид.)	0.318	0.370	0.208
SVD SPPMI	0.35	0.331	0.135
SGNS	**0.704**	**0.578**	**0.351**

SPPMI, но сильно проигрывают по качеству представлениям SGNS. Таким образом, в предложенной модели операции сложения и вычитания являются интерпретируемыми, хотя интерпретируемость и ниже чем в SGNS. Сопоставление некоторых аналогий моделей ARTM и SGNS можно наблюдать в таблице 8.

Таблица 8: Примеры аналогий для ARTM и SGNS

Модель ARTM	
Операция	Результат
king + *girl* − *boy*	*queen*, *princess*, *lord*, *prince*
moscow + *spain* − *russia*	*madrid*, *barcelona*, *aires*, *buenos*
india + *ruble* − *russia*	*rupee*, *birbhum*, *pradesh*, *madhya*
better + *bad* − *good*	*really*, *something*, *thing*, *nothing*
cars + *computer* − *cars*	*computers*, *software*, *servers*, *implementations*

Модель SGNS	
Операция	Результат
king + *girl* − *boy*	*queen*, *princess*, *regnant*, *kings*
moscow + *spain* − *russia*	*madrid*, *barcelona*, *valladolid*, *malaga*
india + *ruble* − *russia*	*rupee*, *rupiah*, *devalued*, *debased*
better + *bad* − *good*	*worse*, *easier*, *prettier*, *funnier*
cars + *computer* − *cars*	*computers*, *software*, *hardware*, *microcomputers*

5.5 Интерпретируемость компонент моделей

Для автоматического оценивания интерпретируемости компонент была использована мера когерентности по 10 и 100 топ-словам для каждой компоненты. Модель SNGS в отличие от тематических моделей не выдаёт вероятностные распределения, поэтому

Таблица 9: Топ-слова некоторых тем тематической модели дистрибутивной семантики

art	airport	carolina	moscow
painting	airlines	florida	dynamo
museum	airways	texas	pfc
painters	flights	georgia	sofia
gallery	airports	alabama	spartak
sculpture	destinations	tenessee	lokomotiv
exhibition	international	mississippi	levski

для оценивания когерентности исходные векторы преобразовывались в распределения. Было использовано два подхода: в первом функция softmax применялась к матрице V по столбцам, а во втором softmax применялся к V по строкам, а затем производилась перенормировка по Байесу. Вероятность слова $p(w)$ оценивалась как $(n_w/N)^{3/4}$.

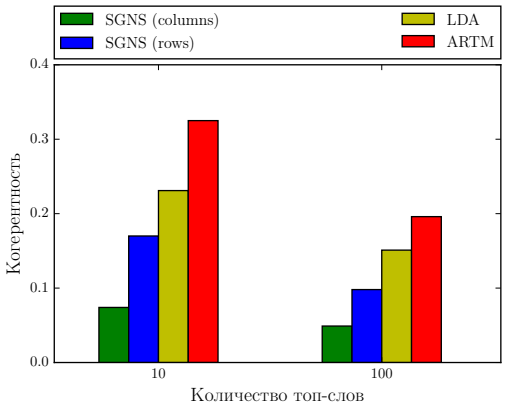

Рис. 3: Сравнение когерентностей моделей

По оценкам когерентности видно, что предложенная модель превзошла и LDA, и модель SGNS. Несмотря на грубость данного подхода, его результаты согласуются с визуальным наблюдениями за топ-словами компонент моделей. Компоненты тематических моделей дистрибутивной семантики являются интерпретируемыми (таблица 9), более того, темы дистрибутивной семантики несколько отличаются от тем традиционных тематических моделей. Некоторые из тем состоят из однородных понятий: названий штатов, названий футбольных клубов.

Визульная интерпретируемость компонент SGNS гораздо ниже чем у тематической модели (таблица 10). Однако, следует отметить, что некоторые закономерности в компонентах всё же прослеживаются. В первой компоненте собрано много восточных муж-

Таблица 10: Топ-слова самых когерентных компонент модели SGNS

rana	membranes	biscuits	geneticist
walnut	absorbed	isip	hereditary
rashid	dental	sponge	ylw
malek	transmembrane	cadbury	genetics
aziz	oral	obedience	dominates
khalid	gre	biscuit	torch
yemeni	strands	nestle	biologist

ских имён, во второй компоненте много биологических терминов, в третьей встречаются слова, связанные со сладостями, а в четвёртов слова, связанные с генетикой.

Когерентность можно считать не по топ-словам модели, а по анти-топ-словам, имеющим самые низкие значения в компоненте. В тематическом моделировании такой подход абсолютно бессмысленен, однако для модели SGNS он имеет смысл (таблица 4). Когерентность, посчитанная по анти-топ-словам, практически не отличается от когерентности по топ-словам.

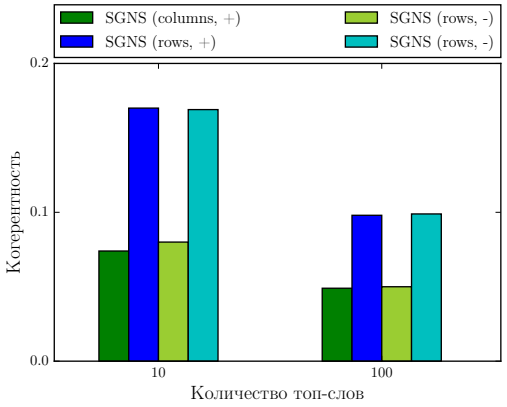

Рис. 4: Сравнение когерентностей модели SGNS при различном учёте топ-слов

Визуальный анализ компонент (таблица 11) также показывает, что в анти-топ-словах компонент можно находить некоторые закономерности. Таким образом можно сделать вывод, что компоненты SGNS являются слабо интерпретируемыми. Также можно сделать вывод о разумном центрировании в модели, положительные и отрицательные значения компонент обладают одинаковой интерпретируемостью.

Таблица 11: Топ-слова самых когерентных компонент модели SGNS (анти-топ-слова)

skywalker	banknotes	martins	stereotypes
darth	signage	italo	cropped
jedi	banknote	wim	ylw
spock	transmembrane	accra	disestablished
maximus	announcements	franck	chr
elijah	wrapper	biscuit	terminated
fir	streak	nestle	reclassified

5.6 Разреженность компонент моделей

Векторы, полученные с помощью модели SGNS, не являются разреженными. Векторы, полученные с помощью тематических моделей, являются разреженными сами по себе, но их разреженность можно увеличить за счёт использования регуляризатора разреживания [24]. Исследуем, как использование регуляризатора разреживания влияет на качестве построенных представлений.

Без использования регуляризатора разреживания, разреженность матрицы Φ тематической модели составляет 80% (таблица 12). С использованием регуляризаторов разреживания матриц Φ и Θ разреженность можно повысить до 93%. Разница на задаче близости сильно разреженных представлений и исходных незначительна.

Таблица 12: Сравнение исходной модели и модели с регуляризатором разреживания

	Доля нулей	WordSim Sim.	WordSim Rel.	WordSim	Bruni MEN	Radinsky M. Turk
ARTM	80%	0.723	**0.675**	**0.682**	0.672	**0.642**
ARTM (+разр.)	93%	**0.728**	0.672	0.68	**0.675**	0.635

5.7 Мультимодальные представления на коллекции новостей

Эксперименты по исследованию мультимодального расширения тематических моделей дистрибутивной семантики проводились на коллекции 100 000 новостей с сайта Lenta.ru. Словарь коллекции после выбрасывания редких слов составил 54693 слова. Каждый документ коллекции состоит из четырёх модальностей: текст, время написания, категория и подкатегория. При создании коллекции использовался размер окна 5.

Таблица 13: Сравнение моделей на задаче близости

Model	WS-sim	WS-rel	MC	RG	ALL
SGNS	0.63	0.53	0.377	0.415	0.567
PWE	0.612/0.649	0.54/0.565	0.648/0.605	**0.63**/0.594	0.583/0.604
Multi-PWE. (sym)	0.646/0.677	0.537/0.576	0.668/0.612	0.618/0.585	0.579/0.608
Multi-PWE. (tokens)	0.646/**0.682**	0.55/**0.58**	**0.675**/0.607	0.617/0.584	0.583/**0.611**

Для оценивания качества задачи близости использовался датасет HJ [11], составленный из переведённых датасетов близости английских слов: MC citeMiller:1991, RG [19] and WordSim353 [18]. В качестве базовых моделей обучались модель SGNS и одномодальная модель (28)-(29). Исследовалось два способа построения мультимодальных моделей:

1. Тематическая модель, использующая все модальности, но строящая псевдодокументы только для основной модальности текстов (Multi-ARTM tokens)

2. Тематическая модель, использующая все модальности и строящая псевдодокументы для токенов всех модальностей (Multi-ARTM sym)

Для тематических моделей провеялось два параметра:

1. представление $p(w|t)$ и косинусная мера близости (первое число)

2. представление $p(t|w)$ и мера близости скалярное произведение (второе число)

По результатам измерения близости (таблица 13) можно сделать вывод, что на маленьких коллекция векторные представления тематических моделей дистрибутивной семантики работают лучше чем SGNS. Это согласаются с привычными представлениями о том, что алгоритмам word2vec для хорошей работы нужна коллекция достаточно большого размера. Использование дополнительных модальностей повышает качество, использование дополнительных псевдо-документов не влияет на качество мультимодальной модели. Однако, как показывает визуальный анализ, представления для модальностей, полученные от псевдо-документов, лучше представлений, полученных из матрицы Ф. Интерпретируемость полученных представлений можно наблюдать на таблице 14, в которой изображены самые близкие слова к соответствующим представле-

Таблица 14: Интерпретируемые векторные представления для моментов времени

премьера star wars 2015-12-18	церемония оскар 2016-02-29	день победы 2015-05-09
джедай	статуэтка	великий
ситх	кинонаграда	годовщина
фетт	номинироваться	фотопортрет
энакин	кинопремия	нормандия
чубакка	линклейтер	парад
киносага	оскар	демонстрация
хэмилл	бёрдмен	шествие
кэрри	удостоиться	владимир
приквел	award	празднование
соло	критик	концентрационный
пробуждение	отрочество	освенцим

ниям для временных промежутков слова. Таким образом, используя аппарат модальностей, можно улучшать качество на задаче близости векторных представлений слов.

6 Заключение

В работе предложены способы построения векторных представлений слов с помощью тематических моделей дистрибутивной семантики. Качество решения задач близости этих представлений сопоставимо с качеством решения задачи близости модели SGNS, качество решения задач аналогии превосходит стандартные подходы в тематическом моделировании. Компоненты полученных представлений являются интерпретируемыми и разреженными.

Исследовано применение оффлайнового EM-алгоритма для построения тематической модели по симметричной исходной матрице F. Показано, что с помощью специальной начальной инициализации матриц $\Phi = \Phi_0$ и $\Theta = \Phi_0^{bayes}$, можно добиться симметричного разложения $F \approx \Phi\Phi^{bayes}$. Показана связь предложенного подхода и тематической модели дистрибутивной семантики Biterm Topic Model.

Предложен способ использования не основных модальностей документа для улучшения качества векторных представлений на задаче близости и построения интерпретируемых векторных представлений для токенов не основных модальностей.

Список литературы

[1] A Neural Probabilistic Language Model. / Y. Bengio, R. Ducharme, P. Vincent, C. Janvin // *Journal of Machine Learning Research.* — 2003. — Vol. 3. — Pp. 1137–1155.

[2] Automatic evaluation of topic coherence / D. Newman, J. H. Lau, K. Grieser, T. Baldwin // Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. — HLT '10. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. — Pp. 100–108.

[3] *Blei D. M., Ng A., Jordan M.* Latent Dirichlet allocation // *JMLR.* — 2003. — Vol. 3. — Pp. 993–1022.

[4] BTM: Topic Modeling over Short Texts / X. Cheng, X. Yan, Y. Lan, J. Guo // *IEEE Transactions on Knowledge and Data Engineering.* — 2014. — Dec. — Vol. 26, no. 12. — Pp. 2928–2941.

[5] *Das R., Zaheer M., Dyer C.* Gaussian lda for topic models with word embeddings // Proceedings of the 53rd Annural Meeting of the Association for Computational Linguistics / Association for Computational Linguistics. — Vol. 53. — 2015. — Pp. 795–804.

[6] Distributed Representations of Words and Phrases and their Compositionality. / T. Mikolov, I. Sutskever, K. Chen et al. // *CoRR.* — 2013. — Vol. abs/1310.4546.

[7] Distributional semantics in technicolor / E. Bruni, G. Boleda, M. Baroni, N.-K. Tran // Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1. — ACL '12. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. — Pp. 136–145.

[8] Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado, J. Dean // *arXiv preprint arXiv:1301.3781.* — 2013. `http://www.bibsonomy.org/bibtex/24a3db34a5744ad8a2704d42f0ef00905/scheuerpflug`.

[9] *Harris Z.* Distributional structure // *Word.* — 1954. — Vol. 10, no. 23. — Pp. 146–162.

[10] *Hoffmann T.* Unsupervised learning by probabilistic latent semantic analysis // *Machine Learning.* — 2001. — Vol. 42, no. 1. — Pp. 177–196.

[11] Human and machine judgements for russian semantic relatedness / A. Panchenko, D. Ustalov, N. Arefyev et al. // Analysis of Images, Social Networks and Texts (AIST'2016). — Springer, 2016.

[12] *Levy O., Goldberg Y.* Linguistic regularities in sparse and explicit word representations // Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Baltimore, Maryland, USA, June. Association for Computational Linguistics. — 2014.

[13] *Levy O., Goldberg Y.* Neural Word Embedding as Implicit Matrix Factorization // Advances in Neural Information Processing Systems 27 / Ed. by Z. Ghahramani, M. Welling, C. Cortes et al. — Curran Associates, Inc., 2014. — Pp. 2177–2185. http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf.

[14] *Levy O., Goldberg Y., Dagan I.* Improving distributional similarity with lessons learned from word embeddings // *TACL*. — 2015. — Vol. 3. — Pp. 211–225. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570.

[15] *Miller G. A., Charles W. G.* Contextual correlates of semantic similarity // *Language and Cognitive Processes*. — 1991. — Vol. 6, no. 1. — Pp. 1–28.

[16] *Pennington J., Socher R., Manning C. D.* Glove: Global vectors for word representation // Empirical Methods in Natural Language Processing (EMNLP). — 2014. — Pp. 1532–1543. http://www.aclweb.org/anthology/D14-1162.

[17] *Pennington J., Socher R., Manning C. D.* Glove: Global vectors for word representation // Empirical Methods in Natural Language Processing (EMNLP). — 2014. — Pp. 1532–1543. http://www.aclweb.org/anthology/D14-1162.

[18] Placing search in context: The concept revisited / L. Finkelstein, E. Gabrilovich, Y. Matias et al. // *ACM Trans. Inf. Syst.* — 2002. — Vol. 20, no. 1. — Pp. 116–131.

[19] *Rubenstein H., Goodenough J. B.* Contextual correlates of synonymy // *Commun. ACM*. — 1965. — Vol. 8, no. 10. — Pp. 627–633.

[20] A study on similarity and relatedness using distributional and wordnet-based approaches / E. Agirre, E. Alfonseca, K. Hall et al. // Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the

Association for Computational Linguistics. — NAACL '09. — Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. — Pp. 19–27.

[21] *Teh Y. W., Newman D., Welling M.* A collapsed variational bayesian inference algorithm for latent dirichlet allocation // In NIPS. — P. 2006.

[22] Topical word embeddings. / Y. Liu, Z. Liu, T.-S. Chua, M. Sun // AAAI. — 2015. — Pp. 2418–2424.

[23] *Vorontsov K.* Additive regularization for topic models of text collections // *Doklady Mathematics*. — 2014. — Vol. 89, no. 3. — Pp. 301–304. http://dx.doi.org/10.1134/S1064562414020185.

[24] *Vorontsov K., Potapenko A.* Additive regularization of topic models // *Machine Learning*. — 2015. — Vol. 101, no. 1-3. — Pp. 303–323.

[25] A word at a time: Computing word relatedness using temporal semantic analysis / K. Radinsky, E. Agichtein, E. Gabrilovich, S. Markovitch // Proceedings of the 20th International World Wide Web Conference. — Hyderabad, India: 2011. — March. — Pp. 337–346.

[26] *Xiao H., Stibor T.* Efficient collapsed gibbs sampling for latent dirichlet allocation // Asian Conference on Machine Learning (ACML). — Vol. 13 of *JMLR W&CP*. — Japan: 2010. — (AR: 31https://www.sec.in.tum.de/assets/Uploads/XiaoStiborACML2.pdf.

[27] *Zuo Y., Zhao J., Xu K.* Word Network Topic Model: A Simple but General Solution for Short and Imbalanced Texts // *CoRR*. — 2014. — Vol. abs/1412.5404. http://arxiv.org/abs/1412.5404;http://dblp.uni-trier.de/rec/bib/journals/corr/ZuoZX14.