Мультимодальная тематическая модель текстов и изображений на основе использования их векторного представления*

Н. Д. Смелик, А. А. Фильченков
smelik@rain.ifmo.ru; afilchenkov@corp.ifmo.ru
Университет ИТМО, Россия, г. Санкт-Петербург, Кронверкский проспект, 49

Целью данной работы является создание мультимодальной тематической модели для изображений и текстов. Предложен подход к построению такой модели на основе векторного представления текстов и изображений. Векторы значимых слов строятся за счет применения Word2Vec, для изображений — как выход последнего неполносвязного слоя сверточной нейронной сети. Предложены алгоритм обучения тематической модели по коллекции аннотированных изображений, а также алгоритмы аннотирования нового изображения и иллюстрирования нового текста. Эксперименты показали, что предложенная модель превосходит аналоги в задаче аннотирования изображений и иллюстрирования текстов.

Ключевые слова: *вероятностная тематическая модель; аннотирование изображение; иллюстрирование текста; сверточные нейронные сети; векторное представление слов*

DOI: 10.21469/22233792.2.4.05

1 Введение

Тематическое моделирование — активно развивающаяся с конца 1990-х гг. область машинного обучения, применимая к анализу текстов. Ее появление обусловлено необходимостью обрабатывать огромные объемы цифровых данных, которые стали доступны человеку после появления Интернета. Тематическая модель определяет, к каким темам относится каждый документ, а также из каких терминов состоит каждая тема. Это позволяет эффективно решать задачи тематического поиска [1], категоризации и классификации текстовых документов [2], а также аннотации разного вида данных. Перечисленные задачи находят применение в различных областях, где приходится иметь дело с поиском различной информации схожей тематики.

Изначально тематические модели учитывали при построении только слова, из которых состоят документы, однако в последнее время стали появляться модели, позволяющие учитывать также сопутствующую документу информацию, такую как теги, авторы [3], атрибуты [4] и др. К этому списку следует также отнести тематическую модель для текстов и изображений. Такая модель позволяет автоматически выделять темы из изображений на основе их описания и в дальнейшем использовать ключевые слова тем для описания новых изображений. Это может найти применение в таких задачах, как аннотирование изображений или поиск тематических иллюстраций для текста.

В последнее время появились инструменты, показавшие отличные результаты в области распознавания образов и обработки естественного языка. Это технология глубокого обучения, успешно применяющаяся для распознавания изображений, и модель векторного

*Работа выполнена при финансовой поддержке Правительства Российской Федерации, грант 074-U01.

представления слов, позволяющая учитывать контекст слова. Использование этих технологий может повысить качество построения тематических моделей для текстов и изображений.

Целью данной работы является повышение качества аннотирования изображений и иллюстрирования текста путем разработки тематической модели на основе совместного использования тематического моделирования, глубоких нейронных сетей и векторного представления слов.

В разд. 2 представлены основные понятия тематических моделей и описание исследований, посвященных этой тематике. В разд. 3 кратко описана идея использовать сверточные нейронные сети и векторные представления слов для построения мультимодальной тематической модели, а также описаны соответствующие технологии. В разд. 4 и 5 описаны алгоритмы соответственно обучения модели и ее применения для аннотирования изображений и иллюстрирования описаний. В разд. 6 приведены подробности реализации модели. В разд. 7 описаны вычислительные эксперименты, проведенные для сравнения модели с аналогами, а в разд. 8 — результаты экспериментов. Раздел 9 является заключительным.

2 Тематические модели

В данном разделе приведены необходимые понятия и обозначения, а также краткий обзор исследований в области тематического моделирования. Вначале будет дано определение тематической модели, а также будут сделаны основные предположения. Затем будет дано краткое описание существующих на данный момент тематических моделей.

2.1 Вероятностная модель коллекции документов

Вероятностная тематическая модель представляет собой модель коллекции текстовых документов, которая определяет темы для каждого документа [5]. Интуитивно можно высказать предположение, что документ, принадлежащий какой-либо конкретной теме, будет содержать много специфических для этой темы терминов. Тематическая модель строит на этом предположении математическую модель, которая позволяет, основываясь на статистических данных, предположить темы, к которым относится документ, а также соотношение этих тем в документе.

Основные предположения базовых вероятностных тематических моделей:

1) не важен порядок документов в корпусе;
2) не важен порядок слов в документе, документ — «мешок слов» [6];
3) слова, встречающиеся в большинстве документов (слова общей лексики), не важны для определения тематики документа;
4) слово в разных формах считается одним и тем же словом;
5) коллекцию документов можно рассматривать как случайную, однородную и независимую выборку пар документ–слово (d, w), $d \in D$, $w \in W_d$;
6) каждая тема $t \in T$ описывается неизвестным распределением $p(w|t)$ на множестве слов W, $w \in W$;
7) каждый документ описывается неизвестным распределением $p(t|d)$ на множестве тем T, $t \in T$;
8) гипотеза условной независимости: $p(w|t, d) = p(w|t)$;
9) $p(w|d) = \sum_{t \in T} p(t|d)p(w|t)$ — вероятностная модель порождения данных.

Построить тематическую модель — значит найти матрицы $\Phi = \|p(w|t)\|$ и $\Theta = \|p(t|d)\|$ для коллекции D. Для оценки параметров Φ и Θ используется принцип максимума правдоподобия:

$$p(D;\Phi,\Theta) = C\prod_{d\in D}\prod_{w\in d} p(d,w)^{n_d w} = \prod_{dinD}\prod_{w\in d} p(w|d)^{n_{dw}}\mathrm{Cp}\,(d)^{n_{dw}} \to \max_{\Phi,\Theta}.$$

где $n_d w$ — число вхождений термина w в документ d; $\mathrm{Cp}\,(d)^{n_{dw}}$ — константа; C — нормировочный множитель, зависящий только от чисел n_{dw}.

Чаще для удобства используется логарифм максимума правдоподобия:

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

$$\sum_{w\in W}\varphi_{wt} = 1;\varphi \geqslant 0; \quad \sum_{t\in T}\theta_{td} = 1;\theta_{td} \geqslant 0.$$

2.2 Обзор существующих тематических моделей

Тематическое моделирование берет свое начало из работы Пападимитриу, Томаки, Рагавана и Вемполы в 1998 г. [7]. В этой работе для построения тематической модели для коллекции текстовых документов использовался метод латентно-семантического анализа (Latent Semantic Analysis, LSA), который был разработан и запатентован в 1988 г. группой ученых, возглавляемых Скотом Дирверстером [8]. Определение слов, из которых состоят темы, а также определение тем, к которым принадлежит документ, осуществляется с помощью применения к матрице «Слова-на-Документы» *сингулярного матричного разложения* (Singular Value Decomposition, SVD) [9]. С помощью этого метода исходная матрица A представлялась в виде произведения трех матриц $A = USV^{\intercal}$, где матрицы U и V — ортогональные, а матрица S — диагональная, содержащая на главной диагонали числа, называемые сингулярными. Основная идея LSA заключалась в том, что матрица A', содержащая только первые k линейно независимые компоненты A, содержит основную структуру зависимостей, которые присутствуют в исходной матрице «Слова-на-Документы». В результате каждое слово и документ представляются в виде вектора в общем пространстве размерности k, и близость между комбинациями слов и/или документов вычисляется с помощью скалярного произведения векторов.

Основным недостатком этой модели является сильное уменьшение скорости вычислений с увеличением объема входных данных [8]. Также стоит отметить, что у SVD, как и у любого другого матричного разложения, отсутствует явное лингвистическое обоснование, поэтому оценка работы модели и интерпретация ее результатов не всегда понятны.

Как дальнейшее развитие LSA Томас Хоффман предложил в 1999 г. новую модель под названием *вероятностный латентно-семантический* анализ (Probabilistic Latent Semantic Analysis, PLSA) [10]. Данная модель основывается на статистической модели скрытых классов и может быть представлена как вероятностная тематическая модель, описанная в предыдущем пункте.

В PLSA для максимизации логарифма правдоподобия используется EM (expectation-maximization) алгоритм [11]. На E-шаге для текущих значений параметров φ_{wt} и θ_{td} с помощью формулы Байеса вычисляются условные вероятности $p(t|d,w)$ для всех тем $t \in T$ для каждого термина $w \in d$ в каждом документе d:

$$H_{dwt} = p(t|d,w) = \frac{p(w|t)p(t|d)}{p(w|d)} = \frac{\varphi_{wt}\theta_{td}}{\sum_{s\in T}\varphi_{ws}\theta_{sd}}.$$

На M-шаге, наоборот, по условным вероятностям тем H_{dwt} вычисляется новое приближение параметров φ_{wt} и θ_{td}.

Преимуществом PLSA по сравнению с предыдущей моделью LSA является то, что PLSA основан на статистике и поэтому лучше подходит для применения на практике. Основным недостатком данного подхода считается невозможность управлять разреженностью матриц Φ и Θ. Также стоит отметить, что число параметров PLSA растет линейно с числом документов в коллекции, что приводит к переобучению модели.

С целью преодоления этих недостатков в 2003 г. Девидом Блеем, Эндрю Ыном и Майклом Джорданом была разработана модель, которая является логическим продолжением PLSA — *латентное размещение Дирихле* (Latent Dirichlet Allocation, LDA) [12]. На данный момент эта модель является самой популярной в тематическом моделировании, и на ее основе было создано большое число других моделей для разного рода задач.

Модель LDA основана на том же принципе максимизации логарифма правдоподобия что и PLSA, однако дополнительно было выдвинуто предположение, что векторы документов $\theta_d = (\theta_{td}) \in \mathbb{R}^{|t|}$ и векторы слов $\varphi_t = (\varphi_{wt}) \in \mathbb{R}^{|w|}$ порождаются распределением Дирихле с параметрами $\alpha \in \mathbb{R}^{|t|}$ и $\beta \in \mathbb{R}^{|w|}$ соответственно. Это позволяет управлять разреженностью матриц Φ и Θ, что приводит к получению более корректного набора тем по сравнению с PLSA.

Основным недостатком LDA является слабое лингвистическое обоснования использования распределения Дирихле в качестве порождающего распределения для θ_d и φ_t. На самом деле это предположение кажется весьма произвольным. Распределение Дирихле было выбрано для удобства вычислений, и вовсе необязательно соответствует реальной порождающей модели.

Последняя модель для определения тем текстовых документов была предложена сравнительно недавно и является альтернативой байесовскому подходу и графическим моделям, которые использовались в PLSA и LDA. Данная модель называется «Аддитивная регуляризация тематических моделей» (Additive Regularization of Topic Models, ARTM) и была предложена в 2014 г. Константином Воронцовым [13].

В основу создания модели легло решение проблемы неустойчивости и неединственности матричного разложения $\Phi\Theta$. Общий подход преодоления этой проблемы состоит в использовании регуляризации. Он заключается в введении ограничений на Φ и Θ, что в конечном итоге приводит к сужению пространства решений.

Соответственно подход ARTM основан на идее многокритериальной регуляризации. Он позволяет строить модели, удовлетворяющие многим ограничениям одновременно. Каждое ограничение формализуется в виде регуляризатора — оптимизационного критерия $R_i(\Phi, \Theta) \to \max$, зависящего от параметров модели. Взвешенная сумма таких критериев $R(\Phi, \Theta) = \sum_{i=1}^{k} \tau_i R_i(\Phi\Theta)$ максимизируется совместно с логарифмом правдоподобия

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

при тех же ограничениях нормировки и неотрицательности.

2.3 Мультимодальные тематические модели текстов и изображений

Тематическая модель, которая строится не только по словам, но и по любым терминам другой модальности, называется *мультимодальной*. К таким терминам могут относиться, например, авторство текста, время его написания или число людей, выразивших в социальной сети одобрение этому тексту. Для разных модальностей вероятностные распределения над терминами должны строиться раздельно. В качестве расширения ARTM в 2015 г. была предложена модель, являющаяся мультимодельным обобщением ARTM [14] с открытой

библиотекой BigARTM[1] [**?**]. Однако поскольку пространство изображений вычислительно неперечислимо, то любое построенное над ним вероятностное распределение будет характеризовать исключительно коллекцию, по которой оно было построено. В связи с этим применить мультимодальную ARTM к данной задаче напрямую невозможно.

Идея построения мультимодальных тематических моделей для текстов и изображений не нова. Первые работы в этом направлении были проделаны Девидом Блеем и Майклом Джорданом в работе [16], в которой была предложена модель на основе LDA для автоматической аннотации изображений под названием Correspondence LDA (CorrLDA). В данной модели изображения сначала сегментируются с помощью алгоритма N-cuts [17], затем для каждого сегмента извлекается вектор признаков, которыми выступают размер, позиция, цвет, текстура и форма, представленные в виде действительных значений. После этого формируются пары (вектор признаков изображения, набор слов описания). Вектор признаков изображения предполагается порожденным многофакторным гауссовым распределением с диагональной матрицей ковариации, а набор слов описания предполагается порожденным полиномиальным распределением на словаре. После этого модель можно рассматривать с точки зрения порождающего процесса, который сначала генерирует дескрипторы изображения, а затем генерирует подпись в виде слова. В частности, сначала создаются N дескрипторов с помощью модели LDA, а затем для каждого из M слов описания выбирается одна из областей изображения, и после этого слово привязывается к выбранному региону.

В работах [18, 19] были предложены модели MixLDA и sLDA, основным отличием от CorrLDA в которых стало использование алгоритма SIFT [20] для извлечения признаков из изображений. Признаки, извлеченные из изображений в ходе обучения моделей, кластеризовались с помощью алгоритма k-means, что позволяло представить изображения в виде «мешка визуальных слов» и объединить их с текстовой модальностью. Для извлечения тем в MixLDA используется вариант LDA, описанный в [12]. В sLDA используется усовершенствованная модель, описанная в [21].

3 Предлагаемый подход и описание технологий

В данной работе предлагается использовать подход к построению тематических моделей текстов и изображений, совмещающий в себе технологии, подробно описываемые в этой главе, а именно: глубокие нейронные сети и векторные представления слов. Предполагается, что для обучения на вход модели подается коллекция изображений, сопровождаемых описаниями. Для извлечения признаков изображения предлагается использовать сверточную нейронную сеть. Каждое значимое слово в аннотации предлагается заменить на его векторное представление, это позволит учитывать также контекст слова. Этот шаг позволит расширить словарь, так как будут учитываться также слова, употребляющиеся в схожих контекстах. Затем набор признаков изображения представляется в виде псевдодокумента, в котором словами будут векторные представления слов из описания к изображению. Нахождение скрытых тем в таких псевдодокументах позволит использовать полученную модель в задачах аннотирования изображений и иллюстрирования текста. Ожидается, что полученная модель повысит качество аннотирования изображения и иллюстрирования текста.

[1] `http://bigartm.org`.

3.1 Сверточные нейронные сети

Нейронные сети — мощная математическая модель, основанная на знаниях, полученных при изучении нейронных связей в мозге. Она представляет собой набор связанных и взаимодействующих между собой искусственных нейронов. Такие нейроны обычно очень просты, но, будучи соединенными в сеть, они способны решать довольно сложные задачи.

Одной из задач, решаемых нейронными сетями, является задача компьютерного зрения (computer vision, CV) [22], а именно: задача распознавания изображений. Наибольших успехов в решении этой задачи добились *сверточные нейронные сети* (Convolutional Neural Network, CNN) [23]. Сверточная нейронная сеть — глубокая нейронная сеть, для которой сделано явное предположение, что на вход подается изображение. Это позволяет добавить в архитектуру такой сети определенные свойства, которые улучшают эффективность и снижают число параметров по сравнению с обычными сетями.

Сверточная сеть использует три типа слоев: сверточные (convolutional), субдискретизирующие (max-pooling) и полносвязные (full connected). Чередование первых двух типов слоев позволяет получить из изображения набор карт признаков (feature map), полносвязный слой применяется для классификации полученных наборов.

Каждый нейрон сверточного слоя отвечает за применение операции свертки части изображения, поданного на вход нейрону, с фильтром, реагирующим на простые линии. Часть изображения, к которой применяется операция свертки, называется окном. За счет смещения этого окна нейрон обрабатывает все изображение. В результате применения фильтра к каждой позиции изображения на выходе сверточного нейрона получается карта признаков. При этом размер этой карты может быть как больше исходного изображения, так и меньше него. Это зависит от способа смещения окна фильтра.

Субдискретизирующие слои обычно находятся между сверточными. Основное назначение субдискретизирующего слоя заключается в уменьшении размерности представления, полученного на предыдущем сверточном слое. Субдискретизирующий нейрон также использует окно для обхода изображения. Для набора значений, попадающих в окно, применяется некоторая функция (например, функция, выбирающая максимум из значений в пределах окна), значение которой записывается на выход нейрона.

Сверточные нейронные сети обладают замечательной особенностью: если убрать полносвязные слои, которые используются для классификации изображений, получим набор признаков, которые характеризуют наше изображение. Также отбрасывание полносвязных слоев позволяет абстрагироваться от категорий изображений, используемых при обучении такой сети, и извлекать признаки из любых изображений. В работе [24] была экспериментально показана успешность применения такой техники извлечения признаков.

3.2 Векторное представление слов

Векторное представление слова (word embedding) [25] — параметризованное отображение слова на пространство большой размерности $\mathbb{R}^n$. Такое представление позволяет выразить семантическое и синтаксическое значение слова в виде вектора фиксированной длины.

В последнее время для порождения векторных представлений наибольшей популярностью пользуется набор алгоритмов, разработанный Томасом Миколовым под названием Word2Vec [26]. В основе Word2Vec лежит нейронная сеть, которая обучается на большом текстовом корпусе и на выходе генерирует для каждого слова из словаря его векторное представление большой размерности. Основной задачей Word2Vec является максимизация

расстояния между векторами слов, близких по смыслу, и минимизация расстояний между векторами слов, различных по смыслу [27].

В Word2Vec используются две модели для векторного представления слов: skip-gram и continuous bag of word (CBOW). Отличие этих моделей заключается в том, что модель skip-gram предсказывает контекст по слову, т. е. данная модель умеет по входному слову w_i предсказывать $(w_{i-3}, w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, w_{i+3})$ — набор слов, которые употребляются чаще всего вместе со словом, поданным на вход. Модель CBOW, наоборот, предсказывает слово по данному контексту, т. е. решает задачу, обратную задаче, которую решает skip-gram. Этот метод подходит наилучшим образом для создания вектора описания слов, так как на выходе выдает вектор контекста, в котором слово часто используется, что позволит учитывать этот контекст при построении тематической модели.

Одним из подходов к настройке параметров модели skip-gram является максимизация логарифма условной вероятности использования слова в контексте. Эту вероятность можно выразить в виде softmax функции:

$$(c|w;\Theta) = \frac{\exp(v_c v_w)}{\sum_{c^f \in C} \exp(v_c v_w)},$$

где v_w — вектор, представляющий слово w; v_c — вектор, который представляет контекст слова w; C — множество всех доступных контекстов. Однако так как контекстов может быть довольно много, то их полный перебор является довольно длительной задачей.

Для уменьшения количества вычислений в word2vec используется метод, разработанный Томасом Миколовым под названием *отрицательное сэмплирование* (negative sampling). В данном методе модель обучается на парах (w, c), где w — слово, а c — контекст этого слова. Вероятность $p(D = 1|w, c)$ означает вероятность того, что пара (w, c) пришла из данных для обучения. Вероятность $p(D = 0|w, c)$ означает вероятность того, что пара (w, c) не является парой из данных для обучения. Тогда задача метода negative sampling сводится к максимизации вероятности того, что пара (w, c) является парой из данных для обучения:

$$\arg\max_{\Theta} \sum_{(w,c)\in D} \log p(D = 1|w, c; \Theta),$$

где

$$p(D = 1|w, c; \Theta) = \frac{1}{1 + \exp(-v_c v_w)}.$$

Эта задача имеет тривиальное решение, если установить $p(D = 1|w, c; \Theta) = 1$ для всех пар (w, c). Однако тогда все векторы получат одно и то же значение, что крайне не желательно.

Один из способов усовершенствовать тривиальное решение — запрещать некоторые комбинации пар (w, c). Это достигается путем генерации множества D' — случайных некорректных пар (w, c), которые не принадлежат данным для обучения (название метода negative sampling происходит из использования множества случайно выбранных неправильных (negative) пар (w, c)). В итоге задача приобретает следующий вид:

$$\arg\max_{\Theta} \sum_{(w,c)\in D} \log p(D = 1|w, c; \Theta) + \sum_{(w,c)\in D'} \log p(D = 0|w, c; \Theta).$$

4 Алгоритм построения модели

Модель для обучения получает на вход корпус, состоящий из изображения и подходящего к нему описания. Описание может быть различной длины, а также их может быть несколько, относящихся к одному изображению.

После обучения модель возвращает матрицы Φ, описывающую распределение векторов слов на темы, и Θ, описывающую распределение тем на образы изображений.

4.1 Предварительная обработка данных

Так как на вход модели поступают сырые данные в виде набора пар изображение и его описание, необходимо подготовить эти данные для дальнейшей обработки. Основная задача предварительной обработки заключается в обработке описаний изображений.

Каждое описание разбивается на слова, при этом нет смысла различать разные формы одного и того же слова, так как это приведет к разрастанию словаря и ухудшению качества модели. Для приведения слов к нормальной форме используются лемматизация или стемминг.

Лемматизация — процесс приведения слова к нормальной форме. Применительно к русскому языку процесс лемматизации приводит существительные в любой форме к форме именительного падежа единственного числа, прилагательные в любой форме — к форме именительного падежа единственного числа мужского рода, а глаголы, причастия и деепричастия в любой форме — к форме глагола в инфинитиве. Лемматизация требует больших временны́х затрат, а также заставляет хранить большие словари. Это оправдано для языков, относящихся к агглютинативным или синтетическим.

На практике чаще всего применяется стемминг. *Стемминг* — это процесс нахождения основы слова. Самый простой способ стемминга заключается в отбрасывании окончания, также существуют и другие, более сложные варианты стемминга [28–31].

После процесса лемматизации или стемминга отбрасываются *стоп-слова* — слова, встречающиеся почти во всех документах. Этот шаг мотивирован тем, что такие слова бесполезны для тематического моделирования, так как они, скорее всего, будут встречаться во всех темах, но на самом деле они не будут характеризовать тему. Это так называемые слова общей лексики. К ним относятся предлоги, союзы, местоимения, числительные, прилагательные, некоторые глаголы и наречия. Число таких слов обычно варьируется в пределах нескольких сотен, так что для их идентифицирования заранее составлен словарь стоп-слов. Заметим, что отбрасывание стоп-слов практически не влияет на длину словаря.

В конечном итоге формируется словарь, состоящий из всех слов, которые встречались во всех описаниях и не были отброшены на предыдущем шаге. При этом каждому слову присваивается идентификатор — номер, под которым он записан в словаре.

Для оценки веса каждого слова в описании используется неотрицательная мера TF-IDF (term frequency — inverse document frequency). Она описывается следующим выражением:

$$\text{tfidf}\,(t, d, D) = \text{tf}\,(t, d) \cdot \text{idf}\,(t, D).$$

Здесь

$$\text{tf}\,(t, d) = \frac{|t|}{\sum_{k \in d} |t_k|}; \quad \text{idf}\,(t, D) = \log \frac{|D|}{|d_i, t|},$$

где $|t|$ — число вхождений термина t в документ; $\sum_{k \in d} |t_k|$ — общее число слов в документе; $|D|$ — число документов в корпусе; $|d_i, t|$ — число документов в коллекции, в которых встречается термин t.

Вес каждого термина также добавляется в словарь.

В итоге после предварительной обработки для каждого изображения формируют взвешенный набор слов, которые наиболее важны при описании изображения:

$$I \leftrightarrow ((\sigma_1, w_1), (\sigma_2, w_2) \ldots, (\sigma_n, w_n)),$$

где σ_i — вес слова w_i.

4.2 Алгоритм обучения модели

Процесс обучения начинается с обучения моделей для векторного представления слов и векторного представления изображений. Для обучения модели векторного представления слов используется модель skip-gram, для обучении векторного представления изображений — CNN, описанные в предыдущем разделе.

Для построения тематических моделей текстовых документов необходимо сформировать матрицу вхождения каждого слова в каждый документ, т. е. набор векторов $w_j = p_1, p_2, \ldots, p_n$, $j = 1, \ldots, m$, где w_j — j-е слово из словаря; p_i — вероятность появления слова w_j в i-м документе; n — число документов в коллекции; m — число слов в словаре.

В рассматриваемой задаче вектор признаков изображения $\mathbf{i}$ будет представлен в виде псевдодокумента, в котором словами будут векторные представления слов $\mathbf{w}$ — из аннотации к изображению. Это позволит построить модель, которая сможет получать распределения тем для каждого изображения, основываясь на словах из его описания. В качестве вероятности появления слова в псевдодокументе будем использовать меру TF-IDF, описанную ранее.

Полученная матрица будет иметь вид:

$$F = (p_{\mathbf{wi}})_{|W| \times |I|}, \quad p_{\mathbf{weie}} = \text{tfidf}\,(\mathbf{w}, \mathbf{i}, I),$$

где $\mathbf{w}$ — векторное представление слова w из словаря W, $\mathbf{i}$ — векторное представление изображения из множества псевдодокументов I.

Для выделения тем будем использовать подход, аналогичный тематической модели ARTM, описание которой дается в разд. 2. Будем решать задачу приближенного представления матрицы F в виде произведения $F \approx \Phi\Theta$ двух матриц. Для данной задачи это будут матрица векторных представлений слов на темы Φ и матрица тем на векторы признаков изображений Θ.

Как уже было сказано в разд. 2, данная задача решается путем максимизации логарифма максимума при ограничениях нормированности и неотрицательности столбцов матриц Φ и Θ:

$$L(\Phi, \Theta) = \sum_{\mathbf{i} \in I} \sum_{\mathbf{w} \in i} p_{\mathbf{wi}} \ln \sum_{t \in T} \varphi_{\mathbf{w}t} \theta_{t\mathbf{i}} \to \max_{\Phi, \Theta},$$

где

$$\sum_{\mathbf{w} \in W} \varphi_{\mathbf{w}t} = 1; \varphi_{\mathbf{w}t} \geqslant 0; \quad \sum_{t \in T} \theta_{t\mathbf{i}} = 1; \theta_{t\mathbf{i}} \geqslant 0\,.$$

Для решения этой задачи применим EM-алгоритм. Перед первой итерацией задаем начальные приближения для параметров $\varphi_{\mathbf{w}t}$ и $\theta_{t\mathbf{i}}$.

Затем на E-шаге по текущим значениям параметров $\varphi_{\mathbf{w}t}$ и $\theta_{t\mathbf{i}}$ с помощью формулы Байеса вычисляются условные вероятности $p(t|\mathbf{i}, \mathbf{w})$ для всех тем $t \in T$ для каждого векторного представления слова $\mathbf{w} \in \mathbf{i}$ в каждом векторе признаков изображений $\mathbf{i}$:

$$H_{iewet} = p(t|\mathbf{i}, \mathbf{w}) = \frac{p(\mathbf{w}|t)p(t|\mathbf{i})}{p(\mathbf{w}|\mathbf{i})} = \frac{\varphi_{\mathbf{w}t}\theta_{t\mathbf{i}}}{\sum_{s \in T} \varphi_{\mathbf{w}s}\theta_{s\mathbf{i}}}.$$

На M-шаге по полученным условным вероятностям тем $H_{\mathbf{i}\mathbf{w}t}$ вычисляется новое приближение параметров $\varphi_{\mathbf{w}t}$ и $\theta_{t\mathbf{i}}$. Псевдокод данного EM-алгоритма приведен на листинге 1.

Алгоритм 1 EM-алгоритм обучения

Вход: коллекция изображений с описаниями I, число тем $|T|$, начальные приближения Φ и Θ
Выход: распределения Φ и Θ
повторять
обнулить $n_{\mathbf{w}t}, n_{\mathbf{i}t}, n_t$ для всех $\mathbf{i} \in I$, $\mathbf{w} \in W$, $t \in T$;
для всех $\mathbf{i} \in I$, $\mathbf{w} \in ie$
$Z = \sum_{t \in T} \varphi_{\mathbf{w}t}\theta_{t\mathbf{i}}$
для всех $t \in T$, что $\varphi_{\mathbf{w}t}\theta_{t\mathbf{i}} > 0$
увеличить $n_{\mathbf{w}t}, n_{t\mathbf{i}}, n_t$ на $\delta = n_{\mathbf{i}w}\varphi_{\mathbf{w}i}\theta_{ti}/Z$
пока Φ и Θ не сойдутся.

Однако, как показано в работе [32], искомое стохастическое матричное разложение $\Phi\Theta$ определено не единственным образом, а с точностью до невырожденного преобразования $\Phi\Theta = (\Phi S)(S^{-1}\Theta)$ при условии, что матрицы $\Phi' = \Phi S$ и $\Theta' = S^{-1}\Theta$ также стохастические, и задача тематического моделирования в общем случае имеет бесконечно много решений. Это ведет к неустойчивости EM-алгоритма. Для решения этой проблемы предлагается ввести дополнительные ограничения $R_i(\Phi, \Theta)$, $i = 1, \ldots, n$ (регуляризаторы) на Φ и Θ для того, чтобы сузить множество решений.

За счет этого задача сводится к максимизации линейной комбинации критериев L и R_i с неотрицательными коэффициентами регуляризации τ_i при все тех же ограничениях неотрицательности и нормировки матриц Φ и Θ:

$$R(\Phi, \Theta) = \sum_{i=1}^{n} \tau_i R_i(\Phi, \Theta); \quad L(\Phi, \Theta) + R(\Phi, \Theta) \to \max_{\Phi,\Theta}.$$

В итоге для решения задачи матричного разложения будем применять модифицированный EM-алгоритм, приведенный на листинге 2.

Алгоритм 2 EM-алгоритм для модели ARTM

Вход: коллекция изображений с описаниями I, число тем $|T|$, начальные приближения Φ и Θ
Выход: распределения Φ и Θ
повторять
обнулить $n_{\mathbf{w}t}, n_{\mathbf{i}t}, n_t$ для всех $\mathbf{i} \in I$, $\mathbf{w} \in W$, $t \in T$;
для всех $\mathbf{i} \in I$, $\mathbf{w} \in ie$
$Z = \sum_{t \in T} \varphi_{\mathbf{w}t}\theta_{t\mathbf{i}}$
для всех $t \in T$, что $\varphi_{\mathbf{w}t}\theta_{t\mathbf{i}} > 0$
увеличить $n_{\mathbf{w}t}, n_{t\mathbf{i}}, n_t$ на $\delta = n_{\mathbf{i}w}\varphi_{\mathbf{w}i}\theta_{ti}/Z$
$\varphi_{\mathbf{w}t} \propto (n_{\mathbf{w}t} + \varphi_{\mathbf{w}t}\partial R/\partial\varphi_{\mathbf{w}t})_+$ для всех $\mathbf{w} \in W$, $t \in T$
$\theta_{t\mathbf{i}} \propto (n_{\mathbf{i}t} + \theta_{t\mathbf{i}}\partial R/\partial\theta_{t\mathbf{i}})_+$ для всех $\mathbf{i} \in I$, $t \in T$
пока Φ и Θ не сойдутся.

В результате работы данного алгоритма будут получены матрицы Φ и Θ, которые выражают условные распределения на множестве векторных представлений слов для каждой темы и условные распределения на множестве тем для каждого вектора образов изображения. Это и есть искомая тематическая модель.

5 Алгоритмы применения модели

Обученная модель получает на вход либо только текст, по которому требуется подобрать иллюстрации, либо только изображение, к которому требуется подобрать описание.

5.1 Алгоритм генерации аннотаций по изображению

Имея разложение «матрицы векторные представления слов на векторы образов изображений» на матрицы Φ и Θ, можно относительно легко генерировать описания изображений. Последовательность шагов для нахождения слов, подходящих к описанию изображения, приведена на рис. 1. Опишем подробнее каждый шаг.

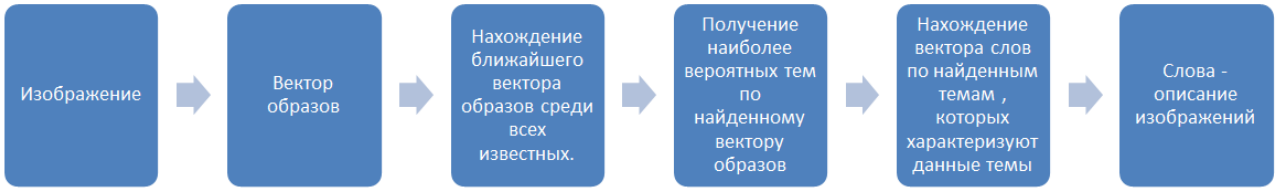


Рис. 1 Схема алгоритма генерации аннотаций по изображению

На вход модели подается изображение. Так как изображение может быть произвольное, а CNN принимает изображения определенного размера, необходимо изменить размер изображения. После этого изображение подается на вход CNN, которая выдает вектор признаков данного изображения. Далее модель начинает поиск вектора, ближайшего к тому, который был подан на вход, среди всех векторов, известных модели. Для найденного вектора из матрицы Θ извлекается распределение тем, полученное во время создания модели.

В результате по найденным темам с помощью матрицы Φ извлекаются векторы слов, которые наилучшим образом характеризуют контекст данной темы. С помощью этих векторов можно найти слова, которые чаще всего употребляются в этом контексте. Это и будет описание поданного на вход модели изображения.

5.2 Алгоритм поиска изображений по текстам

Также имея разложения матрицы F на матрицы Φ и Θ, можно реализовать алгоритм поиска изображений по тексту, или задачу иллюстрации текста.

Последовательность шагов, необходимых для нахождения изображений по текстовому описанию, приведена на рис. 2. Рассмотрим каждый шаг подробнее.

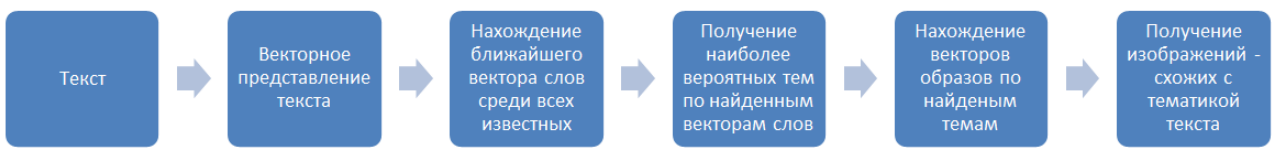


Рис. 2 Схема алгоритма поиска изображений по текстам

На вход модели подается текст. Текст проходит предобработку, описанную в п. 4.1, после чего для каждого термина в тексте находится его векторное представление. Далее текст представляется в виде нового документа, для которого нужно найти распределение тем, т. е. получить вектор θ распределений тем для нового документа. Это можно сделать с помощью алгоритма для обучения модели с той разницей, что матрица Φ уже известна.

После получения распределения тем для текстового документа для каждой темы, вероятность которой в документе не равна 0, находим изображение, максимально соответствующее данной теме. Этот набор изображений и будет иллюстрациями к тексту.

6 Особенности реализации обучения модели

Вся реализация была написана на языке Python версии 2.7. Здесь и далее все используемые библиотеки являются библиотеками для языка Python.

Для получения векторов признаков изображений была использована предобученная CNN под названием VGG-16 [33]. Эта сверточная нейронная сеть показала хорошие результаты в различных соревнованиях по классификации изображений. У этой CNN убирается последний softmax слой, в результате чего сеть выдает 4096-мерный вектор признаков. Использование предобученной сети обусловливалось тем, что процесс обучения такой сложной сети требует большого количества времени и вычислительных ресурсов, при этом качество полученной модели могло получиться хуже. Для взаимодействия с нейронной сетью была использована библиотека Keras[2] для построения нейронных сетей.

Для получения векторных представлений слов был использован также предобученый набор моделей под названием Word2Vec, разработанный кампанией Google [26]. В качестве ее реализации использовалась реализация, предоставленная библиотекой Gensim[3]. Так как для обучения Word2Vec так же, как и для CNN, требуется большое количество времени и вычислительных ресурсов, а также корпус порядка нескольких миллионов слов, было решено также использовать предобученную модель. В качестве предобученной модели для Word2Vec использовалась модель skip-gram с окном размера 10, построенная на основе корпуса Wikipedia[4]. На выходе модель отдает 1000-мерный вектор.

В качестве тематической модели было решено применять ARTM, описанную в разд. 2. В качестве ее реализации использовалась библиотека BigARTM. Выбор данной модели основывается на применении в ней регуляризаторов, использование которых позволяет сравнивать эту модель с моделями PLSA и LDA.

Для предобработки входных данных был написан модуль, осуществляющий предобработку входных данных, генерирующий необходимые словари, а также создающий специальные пакеты (batch), необходимые для библиотеки BigARTM. Описания для каждого изображения были объединены в одно, и ему присваивался уникальный идентификатор; этот же идентификатор присваивался изображению, к которому относились описания. Из описаний изображений были удалены все знаки препинания, а также все неалфавитные символы. После этого описание разбивалось на набор слов, в которые входили только существительные, глаголы и прилагательные. Также в словарь не входили некоторые слова общей лексики, которые могли бы помешать выделить темы. Для удаления таких слов был составлен словарь из 600 стоп-слов, не несущих никакой полезной нагрузки для определения темы. Из оставшихся слов был составлен словарь, в котором каждому слову был

[2] `http://keras.io/`.

[3] `https://github.com/piskvorky/gensim/`.

[4] `https://github.com/idio/wiki2vec`.

присвоен уникальный идентификатор, а в каждом описании слова были заменены на соответствующий идентификатор. При дальнейшем построении тематической модели применялись только идентификаторы слов. После этого каждое слово в словаре было заменено на соответствующее векторное представление, и словарь сохранялся. Для каждого идентификатора в описании производился подсчет меры TF-IDF. После этого формировался batch файл.

Каждое изображение сжималось до размеров 224×224, так как CNN обрабатывает изображения именно такого размера. После этого из изображения извлекался вектор признаков, которому присваивался идентификатор, полученный при обработке описаний этого изображения. Далее формировался и сохранялся словарь. При дальнейшем построении тематической модели использовались только идентификаторы признаков изображений.

После предобработки получалось три файла: batch-файл, представляющий входные данные для BigARTM; словарь соответствия идентификатора слова и его векторного представления; словарь соответствия идентификатора изображения и его векторного представления.

В конечном итоге после обработки batch-файла библиотекой BigARTM сохранялись матрицы Φ и Θ, полученные в результате построения тематической модели.

7 Вычислительные эксперименты

С помощью реализованной модели были проведены эксперименты по поиску изображений по тексту и текстов по описанию.

Для тестирования был использован набор данных Microsoft Common Object in Context[5]. Он содержит 21 000 изображений, каждое из которых сопровождается как минимум пятью описаниями. Словарь содержит 6000 слов.

Для оценки качества построения тематической модели использовались следующие оценки качества:

1) перплексия, определяемая следующей формулой [34]:

$$P = \exp\left(-\frac{1}{n}\sum_{\mathbf{i}\in I}\sum_{\mathbf{w}\in i} n_{\mathbf{iw}} \ln p(\mathbf{w}|\mathbf{i})\right),$$

   где n — длина коллекции в векторных представлениях слов. Перплексия зависит от мощности словаря и распределения частот слов в коллекции;
2) разреженность матриц Φ и Θ, определяемая как доля нулевых элементов в соответствующих матрицах;
3) контрастность, определяемая следующей формулой:

$$\text{Contrast}_t = \frac{1}{|W_t|}\sum_{\mathbf{w}\in W_t} p(t|\mathbf{W});$$

4) чистота ядра темы, определяемая следующей формулой:

$$\text{Purity}_t = \sum_{\mathbf{w}\in W_t} p(\mathbf{w}|t).$$

[5] `http://mscoco.org/`.

Для оценки качества использования модели для задачи аннотации изображений использовались следующие оценки качества: точность (precision), полнота (recall), а также F_1-мера [35].

Для оценки качества использования модели для задачи иллюстрирования текста использовалась точность (accuracy).

8 Результаты экспериментов

8.1 Качество построения модели

Были проведены эксперименты с использованием разных комбинаций регуляризаторов, а также с разным числом тем и итераций построения модели. Исследования проводились для числа тем, равного 50. Это число было выбрано на основе нескольких экспериментов. Следует отметить, что, как было показано в [36], в реальных задачах не существует оптимального числа тем. Набор регуляризаторов и их параметры подбирались также экспериментально. Наилучшими характеристиками обладает модель с использованием комбинации регуляризаторов разреженности для матриц Φ и Θ и декорреляции тем Φ с коэффициентами $-0{,}013$, $-0{,}25$ и $5{,}2 \cdot 10^5$ соответственно.

Для сравнения была выбрана модель без использования регуляризаторов, имитирующая работу PLSA.

Результаты экспериментов приведены в табл. 1 и на рис. 3. Для большей наглядности на рис. 3, *а* отображается логарифм перплексии.

Как видно из табл. 1, использование регуляризаторов дает прирост по всем видам метрик.

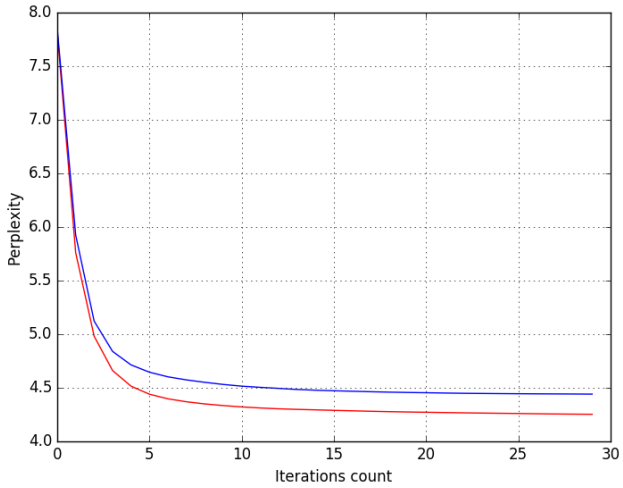
Таблица 1 Сравнение моделей ARTM и PLSA: метрики качества P — перплексия на выборке в 3000 изображений; S_Φ и S_Θ — разреженности матриц Φ и Θ; K_p и K_c — средняя чистота и контрастность ядер тем

Модель	P	S_Φ, %	S_Θ, %	K_p	K_c
ARTM	**70,312**	**96,5**	**88,6**	**0,889**	**0,831**
PLSA	84,597	82,1	84,6	0,461	0,656

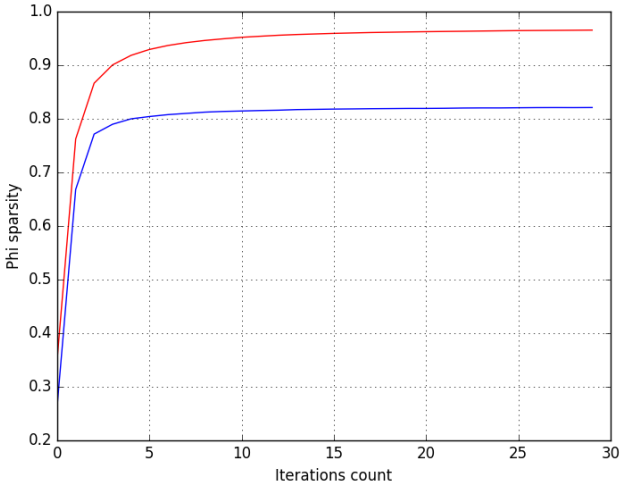
8.2 Аннотация изображений

Для тестирования применялась кросс-валидация на 20% набора данных для тестирования.

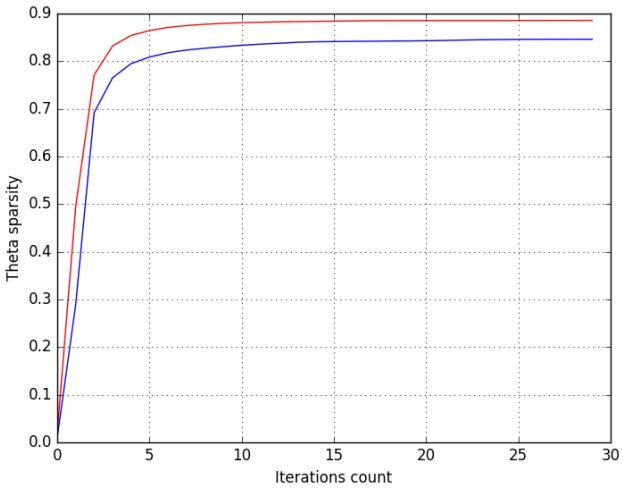
Процесс классификации выглядит следующим образом. На вход классификатору подается изображение без аннотации, на выходе классификатор выдает 10 лучших слов в качестве описания. Для валидации использовалась настоящая аннотация, разбитая на слова.



(*а*)



(*б*)



(*в*)

Рис. 3 График $\log P$ (*а*) и разреженностей матриц Φ (*б*) и Θ (*в*) для моделей PLSA (синие кривые) и ARTM (красные кривые)

Таблица 2 Результаты тестирования моделей в задаче аннотации изображений

Модель	Полнота	Точность	F_1-мера
CORRLDA	34,83	37,85	36,27
MIXLDA	35,20	37,98	36,54
sLDA	35,63	38,46	36,99
PLSA	35,94	38,02	36,92
ARTM	**40,43**	**43,37**	**41,85**

Сравнение проводилось с моделями CorrLDA[6], MixLDA[7] и sLDA[8], описанными в разд. 2. В качестве модели для тестирования использовались обе модели, описанные в предыдущем разделе. Результаты тестирования приведены в табл. 2.

8.3 Иллюстрирование текста

Для иллюстрирования текста на вход подавалось описание и пул изображений кандидатов. Модель находила изображение из пула, которое лучше всего подходит к описанию. Все модели оценивались с помощью лучшего значения точности, которая означает долю успешно подобранных пар в тестовом наборе. Результаты представлены в табл. 3.

Полученные в ходе сравнения результаты показывают, что предложенная модель превосходит существующие аналоги на используемом наборе данных.

Таблица 3 Результаты тестирования моделей в задаче иллюстрирования текста

Модель	Точность
MixLDA	43,5
PLSA	55,8
ARTM	**60,4**

9 Заключение

В данной работе рассмотрена проблема построения мультимодальных тематических моделей текстов и изображений.

В работе были достигнуты следующие результаты.

1. Предложен новый метод построения тематической модели текстов и изображений, учитывающий также контекст слов с помощью их векторного представления.
2. Продемонстрировано применение полученной модели к задачам аннотирования изображений и иллюстрирования текста.

[6]Исходный код для построения модели CorrLDA был взят с сайта `http://home.in.tum.de/~xiaoh/`.

[7]MixLDA моделировался с помощью библиотеки BigARTM.

[8]Исходный код для построения модели sLDA был взят с сайта `http://www.cs.cmu.edu/~chongw/slda/`.

3. Проведены вычислительные эксперименты, показавшие превосходство предложенной модели над ранее известными на конкретном наборе данных.

Стоит отметить, что полученную модель можно использовать в различных задачах, таких как кластеризация текстов и изображений по темам, генерация изображений по описанию и др. Также модель можно легко расширить путем введения дополнительных модальностей для учета различных признаков.

В дальнейшем авторы планируют исследовать влияние методов, использованных для векторного представления, на качество получаемого результата.

Литература

[1] *Yi X., Allan J.* A comparative study of utilizing topic models for information retrieval // European Conference on Information Retrieval, 2009. P. 29–41.

[2] *Rubin T. N., Chambers A., Smyth P., Steyvers M.* Statistical topic models for multi-label document classification // Mach. Learn., 2012. Vol. 88. No. 1-2. P. 157–208.

[3] *Yang M., Hsu W. H.* Hdpauthor: A new hybrid author-topic model using latent Dirichlet allocation and hierarchical Dirichlet processes // 25th Conference (International) on Companion on World Wide Web Proceedings, 2016. P. 619–624.

[4] *Fu Y., Hospedales T. M., Xiang T., Gong S.* Learning multimodal latent attributes // IEEE Trans. Pattern Anal. Machine Intelligence, 2014. Vol. 36. No. 2. P. 303–316.

[5] *Daud A., Li J., Zhou L., Muhammad F.* Knowledge discovery through directed probabilistic topic models: A survey // Frontiers Computer Science China, 2010. Vol. 4. No. 2. P. 280–301.

[6] *Harris Z. S.* Distributional structure // Word, 1954. Vol. 10. No. 2-3. P. 146–162.

[7] *Papadimitriou C. H., Tamaki H., Raghavan P., Vempala S.* Latent semantic indexing: A probabilistic analysis // 17th ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems Proceedings, 1998. P. 159–168.

[8] *Deerwester S. C., Dumais S. T., Furnas G. W., et al.* Computer information retrieval using latent semantic structure, 1989. U.S. Patent 4,839,853.

[9] *Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.* Numerical recipes in C. The art of scientific computing. — 2nd ed. — Cambridge University Press, 1996. Vol. 2. 994 p.

[10] *Hofmann T.* Probabilistic latent semantic indexing // 22nd Annual ACM SIGIR Conference (International) on Research and Development in Information Retrieval Proceedings, 1999. P. 50–57.

[11] *McLachlan G., Krishnan T.* The EM algorithm and extensions. — 2nd ed. — Hoboken, NJ, USA: Whiley-Interscience, 2007. Vol. 382. 400 p.

[12] *Blei D. M., Ng A. Y., Jordan M. I.* Latent Dirichlet allocation // J. Mach. Learn. Res., 2003. Vol. 3. P. 993–1022.

[13] *Воронцов К. В.* Аддитивная регуляризация тематических моделей коллекций текстовых документов // Докл. РАН, 2014. Т. 456. №3. С. 268–271.

[14] *Vorontsov K., Frei O., Apishev M., Romov P., Suvorova M., Yanina A.* Non-Bayesian additive regularization for multimodal topic modeling of large collections // Workshop on Topic Models: Post-Processing and Applications Proceedings. — New York, NY, USA: ACM, 2015. P. 29–37.

[15] *Frei O., Apishev M.* Parallel non-blocking deterministic algorithm for online topic modeling // Analysis of Images, Social Networks and Texts, 2016.

[16] *Blei D. M., Jordan M. I.* Modeling annotated data // 26th Annual ACM SIGIR Conference (International) on Research and Development in Informaion Retrieval, 2003. P. 127–134.

[17] *Meghini C., Sebastiani F., Straccia U.* A model of multimedia information retrieval // J. ACM, 2001. Vol. 48. No. 5. P. 909–970.

[18] *Chong W., Blei D., Li F.-F.* Simultaneous image classification and annotation // IEEE Conference on Computer Vision and Pattern Recognition, 2009. P. 1903–1910.

[19] *Feng Y., Lapata M.* Topic models for image annotation and text illustration // Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010. P. 831–839.

[20] Object recognition from local scale-invariant features // 7th IEEE Conference (International) on Computer Vision Proceedings, 1999. Vol. 2. P. 1150–1157.

[21] *Mcauliffe J. D., Blei D. M.* Supervised topic models // Advances in Neural Information Processing Systems, 2008. P. 121–128.

[22] *Shapiro L., Rosenfeld A.* Computer vision and image processing. — San Diego, CA, USA: Academic Press, 1992. 662 p.

[23] *Krizhevsky A., Sutskever I., Hinton G. E.* Imagenet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems, 2012. P. 1097–1105.

[24] *Athiwaratkun B., Kang K.* Feature representation in convolutional neural networks. arXiv preprint, 2015. arXiv:1507.02313.

[25] *Bengio Y., Ducharme R., Vincent P., Jauvin C.* A neural probabilistic language model // J. Mach. Learn. Res., 2003. Vol. 3. P. 1137–1155.

[26] *Mikolov T., Chen K., Corrado G., Dean J.* Efficient estimation of word representations in vector space. arXiv preprint, 2013. arXiv:1301.3781.

[27] *Goldberg Y., Levy O.* Word2Vec explained: Deriving Mikolov *et al.*'s negative-sampling word-embedding method. arXiv preprint, 2014. arXiv:1402.3722.

[28] *Plisson J., Lavrac N., Mladenic D., et al.* A rule based approach to word lemmatization // 7th Multi-Conference (International) Information Society Proceedings, 2004. Vol. 1. P. 83–86.

[29] *Dolamic L., Savoy J.* Stemming approaches for East European languages // Workshop of the Cross-Language Evaluation Forum for European Languages, 2007. P. 37–44.

[30] *Smirnov I.* Overview of stemming algorithms // Mechanical Translation, 2008. Vol. 52.

[31] *Jongejan B., Dalianis H.* Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike // Joint Conference of the 47th Annual Meeting of the ACL and 4th Joint Conference (International) on Natural Language Processing of the AFNLP Proceedings, 2009. Vol. 1. P. 145–153.

[32] *Vorontsov K. V., Potapenko A. A.* Additive regularization of topic models // Machine Learn., Special Issue on Data Analysis and Intelligent Optimization with Applications, 2015. Vol. 101. No. 1. P. 303–323.

[33] *Simonyan K., Zisserman A.* Very deep convolutional networks for large-scale image recognition. arXiv preprint, 2014. arXiv:1409.1556.

[34] *Brown P. F., Pietra V. J. D., Mercer R. L., Pietra S. A. D., Lai J. C.* An estimate of an upper bound for the entropy of english // Comput. Linguistics, 1992. Vol. 18. No. 1. P. 31–40.

[35] *Friedman J., Hastie T., Tibshirani R.* The elements of statistical learning. — 2nd ed. — Springer ser. in statistics. — Berlin: Springer, 2009. 745 p.

[36] *Vorontsov K. V., Potapenko A. A., Plavin A. V.* Additive regularization of topic models for topic selection and sparse factorization // 3rd Symposium (International) on Learning and Data Sciences. — London, U.K.: University of London, 2015. P. 193–202.

Поступила в редакцию 04.09.2016

Multimodal topic model for texts and images utilizing their embeddings*

N. D. Smelik and A. A. Filchenkov
smelik@rain.ifmo.ru; afilchenkov@corp.ifmo.ru
ITMO University, 49 Kronverksky Pr., St. Petersburg, Russia

A joint topic model for texts and images allows to extract image topics based on their text annotations and to suggest annotations for new images. A novel multimodal topic model for images and texts has been introduced. The proposed model utilizes vector representation of texts and images. Vector representation for a text is based on Word2Vec embedding. Vector representation for an image is convolutional neural network feature map. Then, vector of image is considered as a pseudodocument containing vectors of words instead of words. The proposed model is learnt on the resulting pseudodocument collection. An algorithm to learn the model as well as an algorithm for image annotating and an algorithm for text illustrating with a learnt model have been proposed. Microsoft Common Object in Context dataset was used for experiments. It contains 21,000 images, each has at least 5 annotations. The results show that usage of ARTM leads to much higher quality than the usage of PLSA. The present model was compared with CORRLDA, MIXLDA, and sLDA in image annotating problem and with MIXLDA in text illustrating problem. In both cases, the proposed model showed better results.

Keywords: *topic model; image annotation; text illustration; convolutional neural networks; word embedding*

DOI: 10.21469/22233792.2.4.05

References

[1] Yi, X., and J. Allan. 2009. A comparative study of utilizing topic models for information retrieval. *European Conference on Information Retrieval*. 29–41.

[2] Rubin, T. N., A. Chambers, P. Smyth, and M. Steyvers. 2012. Statistical topic models for multi-label document classification. *Mach. Learn.* 88(1-2):157–208.

[3] Yang, M., W. H. Hsu. 2016. Hdpauthor: A new hybrid author-topic model using latent Dirichlet allocation and hierarchical Dirichlet processes. *25th Conference (International) on Companion on World Wide Web Proceedings*. 619–624.

[4] Fu, Y., T. M. Hospedales, T. Xiang, and S. Gong. 2014. Learning multimodal latent attributes. *IEEE Trans. Pattern Anal. Machine Intelligence* 36(2):303–316.

[5] Daud, A., J. Li, L. Zhou, and F. Muhammad. 2010. Knowledge discovery through directed probabilistic topic models: A survey. *Frontiers Computer Science China* 4(2):280–301.

[6] Harris, Z. S. 1954. Distributional structure. *Word* 10(2-3):146–162.

[7] Papadimitriou, C. H., H. Tamaki, P. Raghavan, and S. Vempala. 1998. Latent semantic indexing: A probabilistic analysis. *17th ACM SIGACT–SIGMOD–SIGART Simposium on Principles of Database Systems Proceedings*. 159–168.

[8] Deerwester, S. C., S. T. Dumais, G. W. Furnas, *et al.* 1989. Computer information retrieval using latent semantic structure. U.S. Patent 4,839,853.

[9] Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 1996. *Numerical recipes in C. The art of scientific computing*. 2nd ed. Cambridge University Press. Vol. 2. 994 p.

*The research was supported by the Russian Government (grant 074-U01) and the Russian Foundation for Basic Research (project No. 16-37-60115).

[10] Hofmann, T. 1999. Probabilistic latent semantic indexing. *22nd Annual ACM SIGIR Conference (International) on Research and Development in Information Retrieval Proceedings*. 50–57.

[11] McLachlan, G., and T. Krishnan. 2007. The EM algorithm and extensions. 2nd ed. Hoboken, NJ: Whiley-Interscience. Vol. 382. 400 p.

[12] Blei, D. M., A. Y. Ng, and M. I. Jordan. 203. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022.

[13] Vorontsov, K. V. 2014. Additive regularization for topic models of text collections. *Dokl. Math.* 89(3):301–304.

[14] Vorontsov, K., O. Frei, M. Apishev, P. Romov, M. Suvorova, and A. Yanina. 2015. Non-Bayesian additive regularization for multimodal topic modeling of large collections. *Workshop on Topic Models: Post-Processing and Applications Proceedings*. New York, NY: ACM. 29–37.

[15] Frei, O., and M. Apishev. 2016. Parallel non-blocking deterministic algorithm for online topic modeling. *Analysis of Images, Social Networks and Texts*.

[16] Blei, D. M., and M. I. Jordan. 2003. Modeling annotated data. *26th Annual ACM SIGIR Conference (International) on Research and Development in Informaion Retrieval Proceedings*. 127–134.

[17] Meghini, C., F. Sebastiani, and U. Straccia. 2001. A model of multimedia information retrieval. *J. ACM* 48(5):909–970.

[18] Chong, W., D. Blei, and F.-F. Li. 2009. Simultaneous image classification and annotation. *IEEE Conference on Computer Vision and Pattern Recognition*. 1903–1910.

[19] Feng, Y., and M. Lapata. 2010. Topic models for image annotation and text illustration. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 831–839.

[20] Lowe, D. G. 1999. Object recognition from local scale-invariant features. *7th IEEE Conference (International) on Computer Mission*. 2:1150–1157.

[21] Mcauliffe, J. D., and D. M. Blei. 2008. Supervised topic models. *Advances in Neural Information Processing Systems*. 121–128.

[22] Shapiro, L., and A. Rosenfeld. 1992. *Computer vision and image processing*. San Diego, CA: Academic Press. 662 p.

[23] Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 1097–1105.

[24] Athiwaratkun, B., and K. Kang. 2015. Feature representation in convolutional neural networks. arXiv preprint. arXiv:1507.02313

[25] Bengio, Y., R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3:1137–1155.

[26] Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint. arXiv:1301.3781.

[27] Goldberg, Y., and O. Levy. 2014. Word2Vec explained: Deriving Mikolov *et al.*'s negative-sampling word-embedding method. arXiv preprint. arXiv:1402.3722.

[28] Plisson, J., N. Lavrac, D. Mladenic, *et al.* 2004. A rule based approach to word lemmatization. *7th Multi-Conference (International) Information Society Proceedings*. 1:83–86.

[29] Dolamic, L., and J. Savoy. 2007. Stemming approaches for East European languages. *Workshop of the Cross-Language Evaluation Forum for European Languages*. 37–44.

[30] Smirnov, I. 2008. Overview of stemming algorithms. *Mechanical Translation* 52.

[31] Jongejan, B., and H. Dalianis. 2009. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. *Joint Conference of the 47th Annual Meeting of the ACL and 4th Joint Conference (International) on Natural Language Processing of the AFNLP Proceedings*. 1:145–153.

[32] Vorontsov, K. V., and A. A. Potapenko. 2015. Additive regularization of topic models. *Machine Learn.* Special Issue on Data Analysis and Intelligent Optimization with Applications 101(1):303–323.

[33] Simonyan, K., and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556.

[34] Brown, P. F., V. J. D. Pietra, R. L. Mercer, S. A. D. Pietra, and J. C. Lai. 1992. An estimate of an upper bound for the entropy of english. *Comput. Linguistics* 18(1):31–40.

[35] Friedman, J., T. Hastie, and R. Tibshirani. 2009. *The elements of statistical learning*. 2nd ed. Springer ser. in statistics. Berlin: Springer. 745 p.

[36] Vorontsov, K. V., A. A. Potapenko, and A. V. Plavin. 2015. Additive regularization of topic models for topic selection and sparse factorization. *3rd Symposium (International) on Learning and Data Sciences*. London, U.K.: University of London. 193–202.

Received September 4, 2016