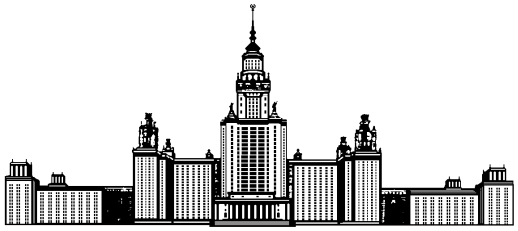

Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Дьяков Илья Андреевич

Тематические модели внимания для анализа связного текста

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
д.ф-м.н., профессор РАН
Воронцов Константин Вячеславович

Москва, 2025

Содержание

Аннотация

Тематическое моделирование — это быстрый и эффективный метод анализа корпуса текстов. В настоящее время наиболее популярными подходами к тематическому моделированию являются BPTM (байесовские вероятностные тематические модели) и ARTM (аддитивная регуляризация для тематического моделирования). Эти модели объединяет то, что они используют представление корпуса документов в виде мешка слов (bag of words). Такое преобразование теряет информацию об относительном положении слов в документах, что может быть важно для понимания смысла слова. С другой стороны, использование *контекста* слова в качестве отдельного «документа» для построения тематической модели позволяет сохранить информацию, утрачиваемую в мешке слов. В данной работе описывается тематическая EM-модель, использующая оконную функцию для учета близости слов при пересчете их тематических представлений. Данный подход имеет схожесть с более общей *моделью внимания* нейронных сетей, но использует оконные функции с фиксированными весами. Это делает его промежуточной моделью между примитивными моделями, построенными на основании представления мешка слов, и сложными нейронными сетями.

1 Вступление

Тематическое моделирование — это метод нежесткой кластеризации корпуса текстов. При этом кластеры называются темами, и кластеризуются как отдельные слова в документах, так и сами документы. В современных тематических моделях обычно принято принимать гипотезу мешка слов. Она заключается в том, что относительное положение слов в документе не влияет на тематику документа.

Данное допущение позволяет использовать компактное представление корпуса документов в виде матрицы (обычно, вообще говоря, разреженной), где по строкам располагаются документы, а по столбцам — слова. На пересечении документа и слова стоит число, равное количеству вхождений слова в документ. Такое представление позволяет свести задачу тематического моделирования к задаче матричной факторизации, заложенной в основу модели ARTM.

Однако гипотеза мешка слов кажется все менее состоятельной с ростом длины документа. Достаточно привести в пример слова-омонимы, которые могут использоваться в одном документе несколько раз в разном смысле.

Отказываясь от гипотезы мешка слов, мы теряем удобное представление корпуса документов и вынуждены работать с каждым словом в отдельности. Очевидно, что при таком переходе в асимптотике итоговых алгоритмов появится количество всех слов корпуса. Тем не менее, такой подход позволяет учитывать произвольный контекст слова, что потенциально делает её качественно лучшей моделью.

2 Постановка задачи

Введем следующие обозначения:

- W, D, W_d, C — количество слов в словаре, документов в корпусе, слов (термов) в документе d и слов (термов) в контексте соответственно. $I = \sum_d W_d$ — количество слов (термов) в корпусе.
- $\mathfrak{W} = \{w | w = \overline{1, \ldots, W}\}$ — множество токенов уникальных слов (термов) в корпусе.
- $\mathfrak{D} = \{\mathfrak{D}_d | d = \overline{1, \ldots, D}\}, \mathfrak{D}_d = \{w_{dk} | k = \overline{1, \ldots, W_d}\}$ — множество документов.
- $\mathfrak{C}_i = \{w_{i+c} | 0 < |c| \leq C, w_i \in \mathfrak{D}_d \Rightarrow w_{i+c} \in \mathfrak{D}_d\}$ — контекст слова (терма) на глобальной позиции i.
- $\mathfrak{C} = \{\mathfrak{C}_i | i = \overline{1, \ldots, I}\}$ — множество контекстов.
- $\mathfrak{T} = \{t | t = \overline{1, \ldots, T}\}$ — множество тем.

Обратим внимание на то, что, хотя в данной постановке само слово не входит в собственный контекст, при включении его в контекст все дальнейшие рассуждения остаются в силе.

Классическая постановка задачи тематического моделирования — это нахождение параметров $\phi_{wt} = p(w|t)$ и $\theta_{td} = p(t|d)$ по коллекции документов $\mathfrak{D}$:

$$p(w|d) = \sum_{t \in T} p(w|t, d) p(t|d) = \sum_{t \in T} p(w|t) p(t|d) = \sum_{t \in T} \phi_{wt} \theta_{td} \tag{1}$$

$$\sum_{w \in \mathfrak{W}} \phi_{wt} = 1, \quad \phi_{wt} \geq 0, \quad \sum_{t \in \mathfrak{T}} \theta_{td} = 1, \quad \theta_{td} \geq 0 \tag{2}$$

Так как в данной постановке задача представляет из себя задачу низкорангового стохастического матричного разложения, в общем случае она имеет бесконечное количество решений. Для получения единственного решения используют регуляризацию [17]. Аддитивная регуляризация тематических моделей (ARTM) [5] основана на максимизации линейной комбинации логарифма правдоподобия и регуляризаторов $R_i(\Phi, \Theta)$.

$$\sum_{d \in D} \sum_{w \in W} n_{dw} \log \sum_{t \in T} \phi_{wt} \theta_{td} + R(\Phi, \Theta) \to \max_{\Phi, \Theta} \tag{3}$$

$$R(\Phi, \Theta) = \sum_i \tau_i R_i(\Phi, \Theta), \quad \tau_i > 0 \tag{4}$$

Позже было показано [11], что можно уйти от оптимизации по матрице Θ, введя отношение-регуляризатор

$$\Theta = \Theta(\Phi) \tag{5}$$

Это позволило значительно ускорить итерации существующего EM-алгоритма, так как он требовал многократного прохода по всей коллекции на каждой итерации для оптимизации матрицы Θ.

Заметим, что формула 5 не подразумевает явного разделения термов матрицы Φ по документам. Это позволяет перейти к разделению слов по произвольным границам — словосочетаниям, абзацам и т.д. Введем следующее отношение на матрицы Φ и Θ:

$$\theta_{ti} = p(t|i) = \sum_{w \in \mathfrak{C}_i} \phi'_{tw} \alpha(w|i), \quad \sum_{w \in \mathfrak{C}_i} \alpha(w|i) = 1, \quad \alpha(w|i) \geq 0 \tag{6}$$

где ϕ'_{tw} — перенормировка строк матрицы Φ по формуле Байеса:

$$\phi'_{tw} = \underset{t \in \mathfrak{T}}{\text{norm}}(\phi_{wt} p(t)) = p(t|w) \tag{7}$$

В такой постановке роль «документа» играет произвольный контекст слова. Понятно, что контекстную модель можно свести к классической документоориентированной модели, взяв контекст каждого слова равным документу, в котором оно содержится, и приняв $\alpha(w|i) = \frac{1}{n_d}$.

Для вывода EM-алгоритма воспользуемся следующей теоремой.

Теорема 1. *Пусть функции $\theta_{td}(\Phi)$ и $R(\Phi, \Theta)$ непрерывно дифференцируемы. Тогда точка Φ_0 локального экстремума задачи 3 с ограничениями 2, 5 удовлетворяет системе уравнений с вспомогательными переменными $p_{tdw} = p(t|d, w), n_{td}$ и p'_{tdw}, если из решения исключить нулевые столбцы матриц Φ и Θ:*

$$\begin{cases} p_{tdw} = \underset{t \in \mathfrak{T}}{norm}(\phi_{wt} \theta_{td}) \\ n_{td} = \sum\limits_{w \in \mathfrak{D}_d} n_{dw} p_{tdw} \theta_{td} \dfrac{\partial R}{\partial \theta_{td}} \\ p'_{tdw} = p_{tdw} + \dfrac{\phi_{wt}}{n_{dw}} \sum\limits_{s \in \mathfrak{T}} \dfrac{n_{sd}}{\theta_{sd}} \dfrac{\partial \theta_{sd}}{\partial \phi_{wt}} \\ \phi_{wt} = \underset{w \in \mathfrak{W}}{norm} \left(\sum\limits_{d=0}^{D} n_{dw} p'_{tdw} + \phi_{wt} \dfrac{\partial R}{\partial \phi_{wt}} \right) \end{cases} \tag{8}$$

Используя теорему 1 и выражение 6, получим следующую итерационную схему:

$$\begin{cases} \phi'_{tw} = \underset{t\in\mathfrak{T}}{\text{norm}}(\phi_{wt}p(t)) \\ \theta_{ti} = \sum\limits_{w\in\mathfrak{C}_i} \phi'_{tw}\alpha(w|i) \\ p_{ti} = \underset{t\in\mathfrak{T}}{\text{norm}}(\phi_{w_i t}\theta_{ti}) \\ n_t = \sum\limits_{i=1}^{I} p_{ti} \\ \phi_{wt} = \underset{w\in\mathfrak{W}}{\text{norm}}\left(\sum\limits_{i=1}^{I} I[w_i = w]p_{ti} + \phi_{wt}\dfrac{\partial R}{\partial \phi_{wt}}\right) \end{cases} \tag{9}$$

3 Исследование связи контекстной и классической тематических моделей

Исследование связи контекстной и классической моделей необходимо сразу по нескольким причинам. Первая причина заключается в обосновании консистентности метрик, по которым будут сравниваться модели. Вторая причина — это исследование области применимости контекстной модели. Текущая постановка задачи и полученный EM-алгоритм 9 не имеют упоминаний документов как таковых, но важным преимуществом классической тематической модели было то, что она кластеризовала как слова, так и документы.

3.1 Эквивалентность метрик

Перед непосредственным сравнением моделей, использующих документ и произвольный контекст, необходимо обосновать, как будет изменяться вычисление метрик при таком переходе.

Сравним два подхода к вычислению перплексии: для модели, использующей мешок слов, и для модели, использующей контексты. Покажем, что в случае, когда контекст совпадает с документом, эти перплексии переходят одна в другую.

Перплексия для модели мешка слов вычисляется по следующей формуле:

$$P(D,p) = \exp\left\{-\frac{1}{I}\sum_{d=1}^{D}\sum_{w=1}^{W} n_{dw}\ln p(w|d)\right\}, \tag{10}$$

где n_{dw} — количество вхождений слова w в документ d.

Перплексия для контекстной модели равна

$$P(C,p) = \exp\left\{-\frac{1}{I}\sum_{i=1}^{I}\ln p(w_i|i)\right\} \tag{11}$$

Теорема 2. *Пусть каждый контекст представляет собой документ, т.е.* $\mathfrak{C} = \mathfrak{D}$. *Тогда формулы 10 и 11 равносильны.*

Доказательство. Формула перплексии для контекстной модели отличается от формулы для модели мешка слов только вычислением правдоподобия. Преобразуем его.

$$\sum_{i=1}^{I}\ln p(w_i|i) = \sum_{i=1}^{I}\mathcal{I}(w_i=w)\mathcal{I}(i=d)\ln p(w_i|i) =$$
$$= \sum_{d=1}^{D}\sum_{w=1}^{W}\sum_{k=1}^{n_{dw}}\ln p(w|d) = \sum_{d=1}^{D}\sum_{w=1}^{W} n_{dw}\ln p(w|d)$$

□

Таким образом, мы обобщили вычисление перплексии на случай произвольного контекста.

3.2 Восстановление кластеров документов в контекстной модели

Для начала рассмотрим одномерный случай. Правдоподобие контекстной модели будет записываться следующим образом

$$\mathcal{L}(\Phi) = \sum_i \log p(w_i|t)p(t|i) = \sum_i \log \phi_{wt}\theta_{ti} = \sum_i \log \phi_{wt} + \sum_i \log \theta_{ti} \tag{12}$$

Правдоподобие классической модели будет выглядеть схожим образом:

$$\mathcal{L}(\Phi) = \sum_{d,w} n_{dw}\log p(w|t)p(t|d) = \sum_{d,w} n_{dw}\log \phi_{wt} + \sum_{d,w} n_{dw}\log \theta_{td} \tag{13}$$

Теорема 3. *Преобразование*

$$p(t|d) = \sqrt[n_d]{\prod_{i\in\mathfrak{D}_d} p(t|i)} \tag{14}$$

сохраняет правдоподобие модели.

Доказательство. Заметим, что мы можем поменять порядок суммирования слагаемого в итоговом выражении 12 так, чтобы оно совпало со слагаемым из 13:

$$\sum_i \log \phi_{w_i t} = \sum_i I[i = d] I[w_i = w] \log \phi_{wt} = \sum_{d,w} n_{dw} \log \phi_{wt}$$

Приравняем оставшиеся слагаемые для произвольного документа d:

$$\sum_w n_{dw} \log p(t|d) = \sum_{i \in \mathfrak{D}_d} \log p(t|i)$$

$$n_d \log p(t|d) = \log \prod_{i \in \mathfrak{D}_d} p(t|i)$$

$$\log p(t|d) = \frac{1}{n_d} \log \prod_{i \in \mathfrak{D}_d} p(t|i)$$

$$p(t|d) = \sqrt[n_d]{\prod_{i \in \mathfrak{D}_d} p(t|i)}$$

□

Таким образом, мы можем однозначно восстановить кластеры документов в контекстной модели в одномерном случае.

4 Эксперименты

Эксперименты проводились с целью изучения свойств новой модели, а также с целью сравнения ее с текущим SotA-подходом. На данный момент им является модель BigARTM с открытым исходным кодом. Основными метриками качества, используемыми в экспериментах, являются перплексия (perplexity), разреженность матрицы Φ (phi sparsity), разнообразие тем (topic variance) и когерентность (coherence). В метрике разнообразия тем считалось расстояние Жаккара между 10 самыми вероятными словами из каждой темы. В когерентности считалась попарная PMI для 10 наиболее вероятных слов в каждой теме.

На графиках яркими линиями изображены усредненные значения метрик на 20 запусках с различными начальными приближениями. Также изображается 95% доверительный интервал для каждой метрики. Параметры модели по умолчанию: размер контекста и параметр γ равны соответственно 10 и 0.6, размер батча $1e4$, шаг обучения 0.01, количество тем: 10. Слово участвует в вычислении собственного контекста, если не сказано иное.

Датасетом для проведения экспериментов был выбран «20 Newsgroups», представляющий собой коллекцию из 18846 статей на различные тематики. Количество слов в используемом словаре равно 107672.

Сравнительные эксперименты, измеряющие производительность, проводились на открытой платформе Kaggle. Процессор, использовавшийся для вычислений: Intel(R) Xeon(R) CPU @ 2.20GHz. Версия Python окружения: 3.10.12.

4.1 Сравнение моделей с целевым словом в контексте и без него

В данном эксперименте исследовалось влияние наличия слова в собственном контексте на качество модели и скорость сходимости.

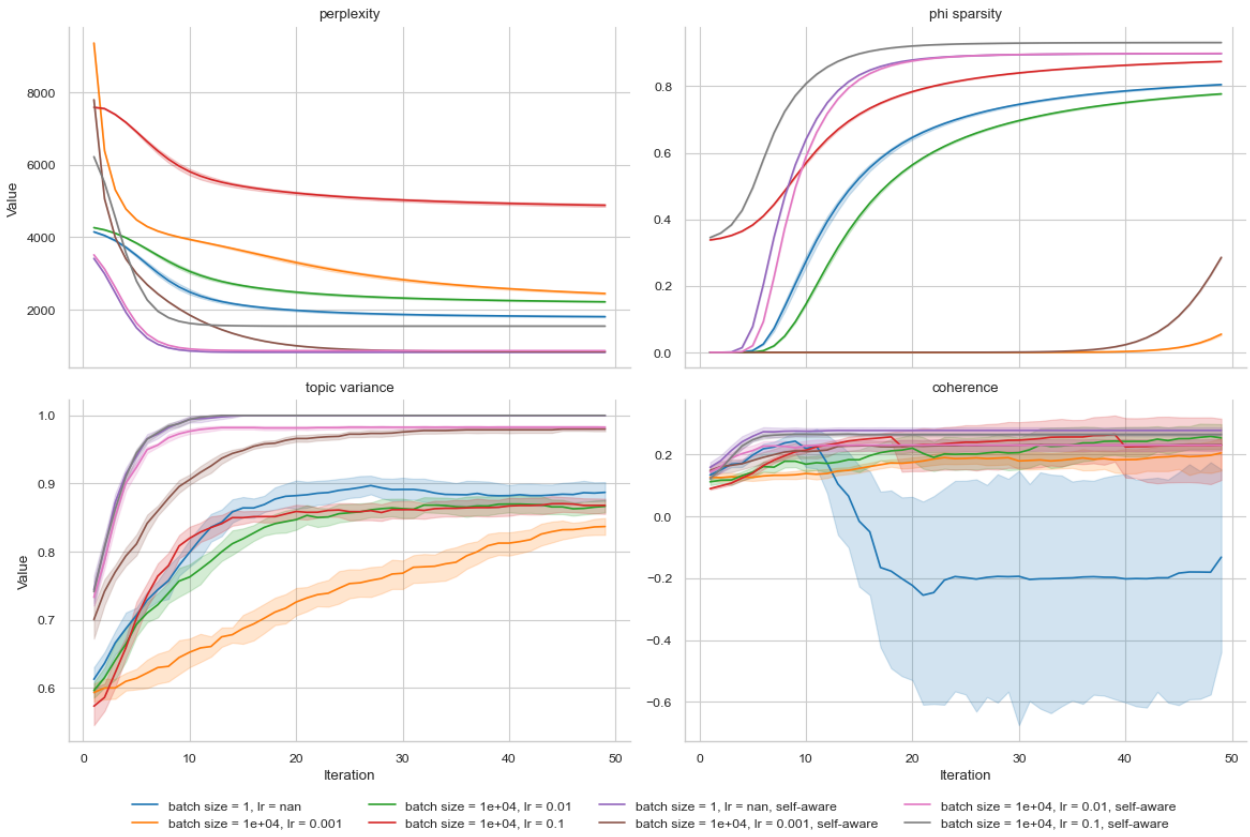

Рис. 1: Зависимость метрик от наличия слова в собственном контексте

Как видно из графиков, добавление слова в собственный контекст значительно улучшает метрики в целом. Сходимость становится быстрее, и алгоритм достигает локального минимума за конечное число итераций. Кроме того, при широком разбросе шага обучения (от 0.01 до 0.001), метрики сходятся к одним и тем же значениям, но с разной скоростью. Таким образом, подбор шага обучения требуется только для ускорения сходимости, но не для улучшения качества итоговой модели, что безусловно является плюсом.

4.2 Исследование влияния весов слов в контексте на сходимость модели

В данном эксперименте исследовалась устойчивость и скорость сходимости модели в зависимости от размера контекста и параметра γ, отвечающего за то, насколько быстро убывают веса термов при удалении от целевого слова. Напомним, что контекст терма представляет собой множество слов $\mathfrak{C}_i = \{w_{i+c} | 0 < |c| \leq C, w_i \in \mathfrak{D}_d \Rightarrow w_{i+c} \in \mathfrak{D}_d\}$. Будем называть число C размером контекста. Тогда, например, при размере контекста N количество термов, входящих в контекст, будет равно $2N$, если не учитывать само слово в собственном контексте, и $2N + 1$ иначе.

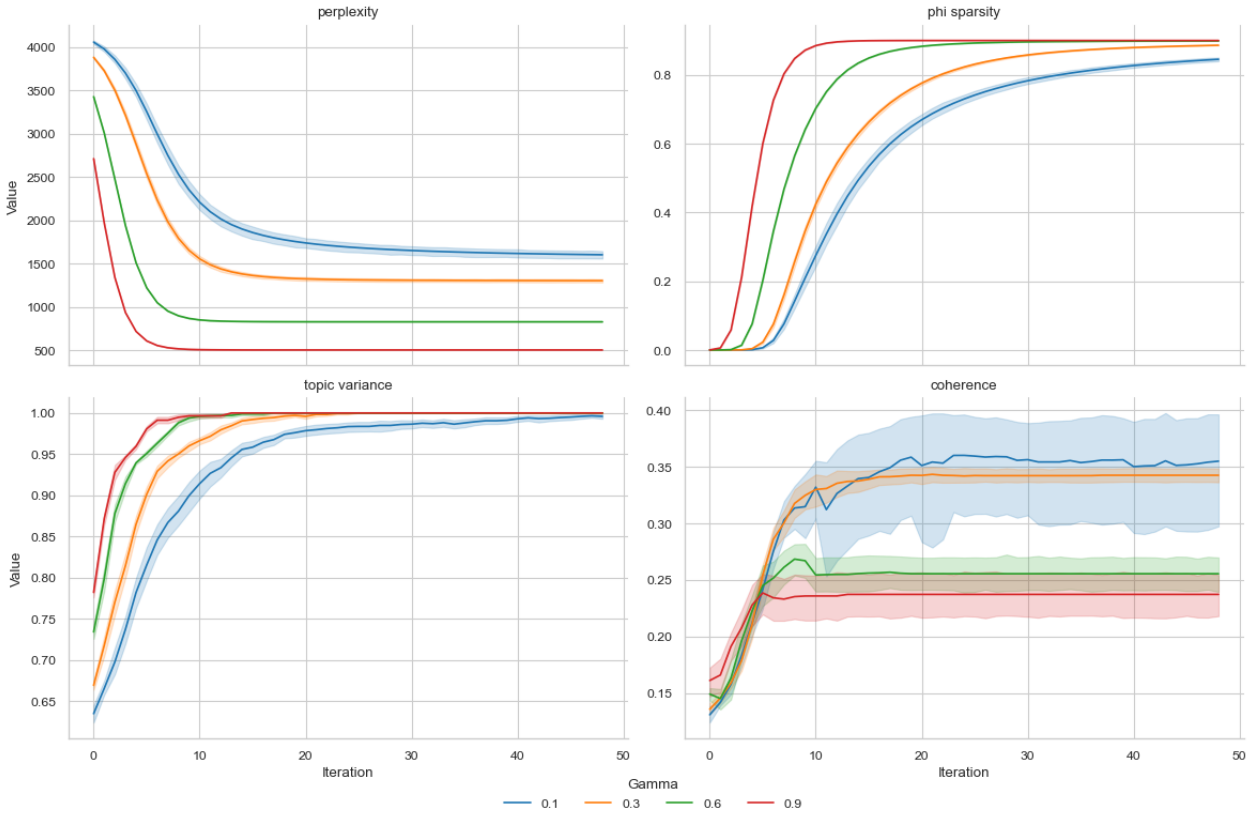

Рис. 2: Усредненные по размеру контекста значения метрик

На графике изображены усредненные метрики для 20 запусков модели с размером контекста $5, 10$ и 20 (60 запусков всего). Как можно судить по 95% доверительному интервалу, размер контекста не меняет общей тенденции изменения метрик, в отличие от параметра γ. Это наблюдение позволяет утверждать, что для оптимизации большинства метрик будет подходить модель с небольшим размером контекста. Так как асимптотика итоговой модели равна $O(ITC)$ (см. раздел 5.2), то можно утверждать, что данное свойство модели позволяет эффективно увеличивать производительность модели, при этом практически не теряя в качестве.

Одной из наиболее интересных зависимостей является то, что перплексия и когерентность с ростом γ уменьшаются. Это говорит о том, что направление оптимизации этих двух метрик является противоположным. В связи с этим был проведен вспомогательный эксперимент с целью оценки того, насколько каждая из упомянутых метрик согласуется с оценкой качества тем человеком. При визуальной оценке тем оказалось, что перплексия плохо коррелирует с оценкой тем человеком, в то время как более высокие значения метрики когерентности отвечали моделям с более интерпретируемыми темами. Было замечено, что уменьшение параметра γ (в большей мере) и увеличение размера контекста (в меньшей мере) приводит к улучшению интерпретируемости тем.

4.3 Исследование влияния размера батча на сходимость модели

Целью данного эксперимента было исследование сходимости модели при разном размере батча. Исследование проводилось для трех конфигураций: обучение без разбиения данных на батчи, обучение с размером батча $1e4$ и обучение с размером батча $1e5$. Как видно из графика 1, даже для небольшого батча модель сходится к очень близким с моделью, обучавшейся на всех данных, значениям при правильном выборе шага обучения. Поэтому в данном эксперименте исследовалось влияние размера батча на модели, обучающиеся без учета целевого терма в контексте.

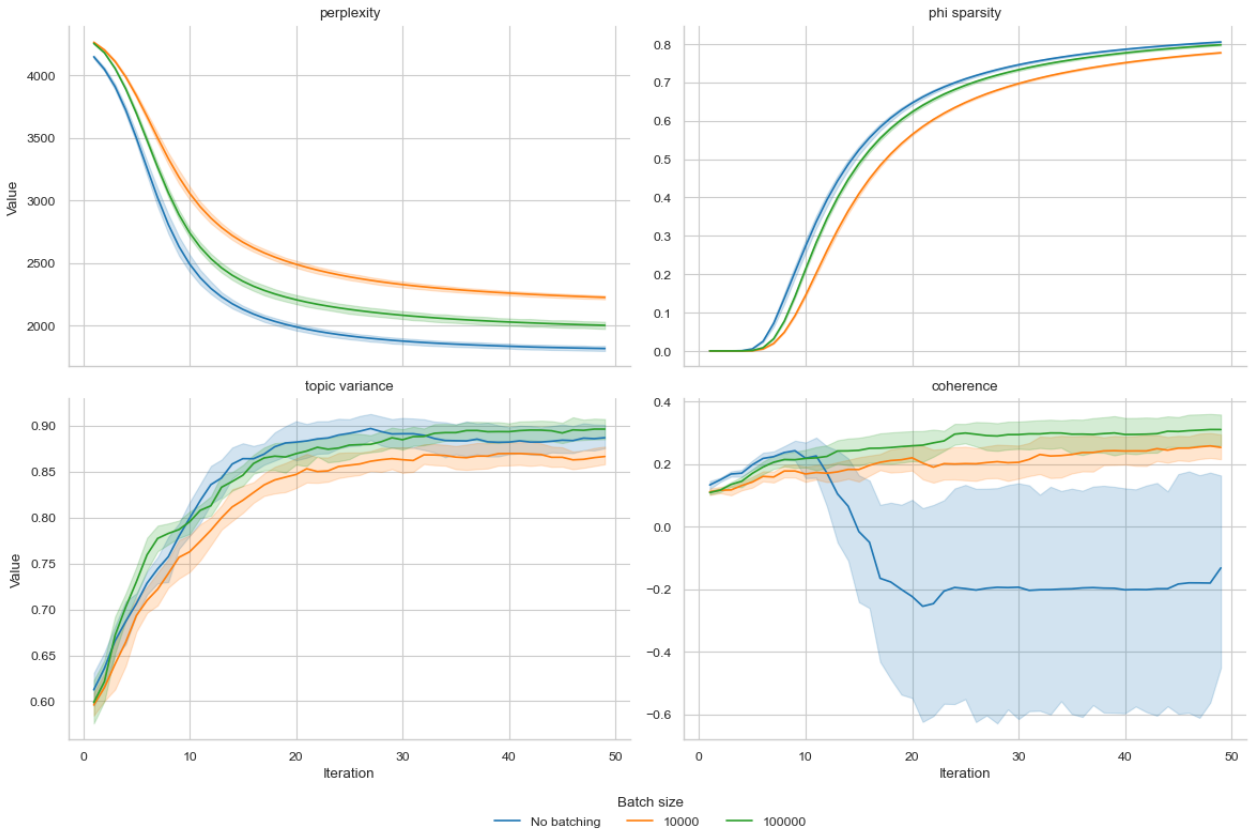

Рис. 3: Зависимость метрик от размера батча

Для данных, разбитых на батчи, подбиралось оптимальное значение шага обучения по логарифмической шкале. Исследование показало, что оптимальные значения шага обучения изменялись обратно пропорционально размеру батча. Для размера батча $1e4$ был выбран шаг обучения 0.01, для размера батча $1e5$ был выбран шаг обучения 0.1.

Исходя из данных, представленных на графиках, можно сделать вывод, что увеличение размера батча ведет к более стабильной и качественной сходимости. Тем не менее, обучение со слишком большим размером батча может негативно сказываться на метрике когерентности.

4.4 Сравнение контекстной модели и BigARTM

Сравнение данных моделей производилось с точки зрения качества и производительности. BigARTM автоматически подбирает оптимальное количество батчей для корпуса, поэтому количество батчей для него фиксировано и равно 10 на данном датасете. При обучении моделей не использовались регуляризаторы.

4.4.1 Значения метрик для разложения на 10 тем

Исходя из результатов предыдущих экспериментов (4.3, 4.1), сравнивались только контекстные модели, обучающиеся на полных данных, т.к. было показано, что модели, обучающиеся на батчах, могут достигать значений метрик, близких к значениям метрик моделей, обучающихся на полных данных, с помощью подбора размера батча и шага обучения.

Важно отметить, что на данном графике используется экспонированная когерентность для лучшей интерпретируемости результатов. Обе контекстные модели оказываются лучше модели мешка слов. При этом модель, использующая слово в собственном контексте, обходит BigARTM с большим отрывом.

4.4.2 Значения метрик для разложения на 100 тем

При разложении на 100 тем наблюдается аналогичная разложению на 10 тем ситуация. Модель, учитывающая слово в собственном контексте, почти на порядок обходит другие модели по метрике перплексии, при этом не деградируя по метрике разнообразия тем. То, что у данной модели когерентность становится хуже, чем у модели, не учитывающей слово в собственном контексте, можно объяснить большей разреженностью матрицы Φ и большим разнообразием тем.

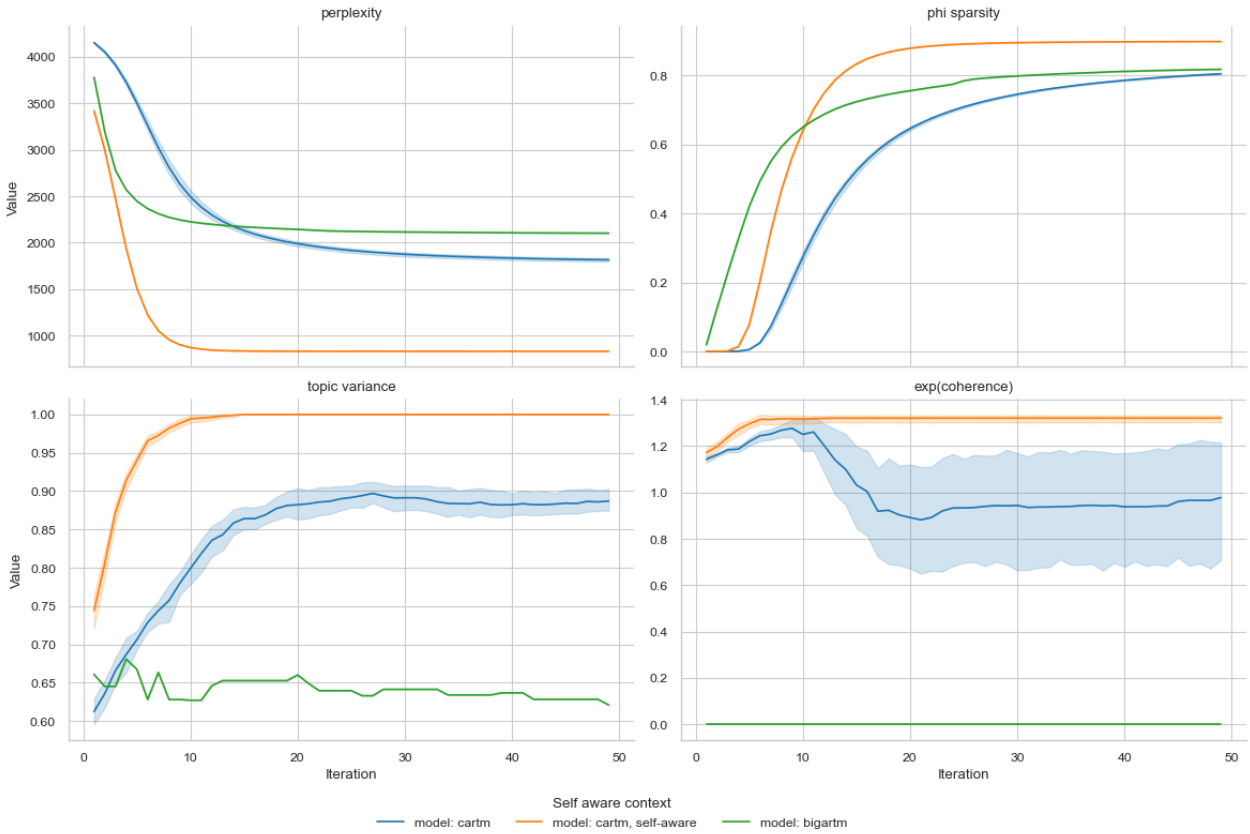

Рис. 4: Сравнение метрик моделей при разложении на 10 тем

4.4.3 Время работы

Скорость работы обеих моделей замерялась для 50 итераций. В таблице указаны усредненные значения, а также дисперсия результатов измерений (mean ± std. dev.). Всего производилось 10 запусков для каждой исследуемой конфигурации.

model	time@10 topics	time@30 topics	time@70 topics	time@100 topics
CARTM@CPU	4min 55s ± 1.7s	9min 20s ± 9.52s	20min 37s ± 2.36s	25min 52s ± 4.63s
BigARTM	1min 30s ± 1.6s	3min 5s ± 1.98s	4min 55s ± 653ms	6min 22s ± 5.81s

Таблица 1: Время работы 50 итераций моделей

Ожидаемо, время работы BigARTM ниже при сравнении производительности без графического ускорителя. Это обусловлено несколькими факторами. Во-первых, бекенд BigARTM написан на чистом C++, в то время как CARTM написан на Python с использованием фреймворка JAX. Это, с одной стороны, позволяет CARTM оставаться легковесным и переносимым пакетом, а с другой стороны, не позволяет добавлять какие-либо специфические аппаратные оптимизации в код. Во-вторых, так как у сравниваемых моделей различные подходы к оптимизации, они имеют различные временные асимптотики. Для BigARTM это $O(DWT)$, т.е. производительность модели зависит от размера словаря, количества документов и количества тем. Для CARTM это $O(ICT)$, т.е. производительность модели зависит от количества слов в корпусе, размера контекста и количества тем.

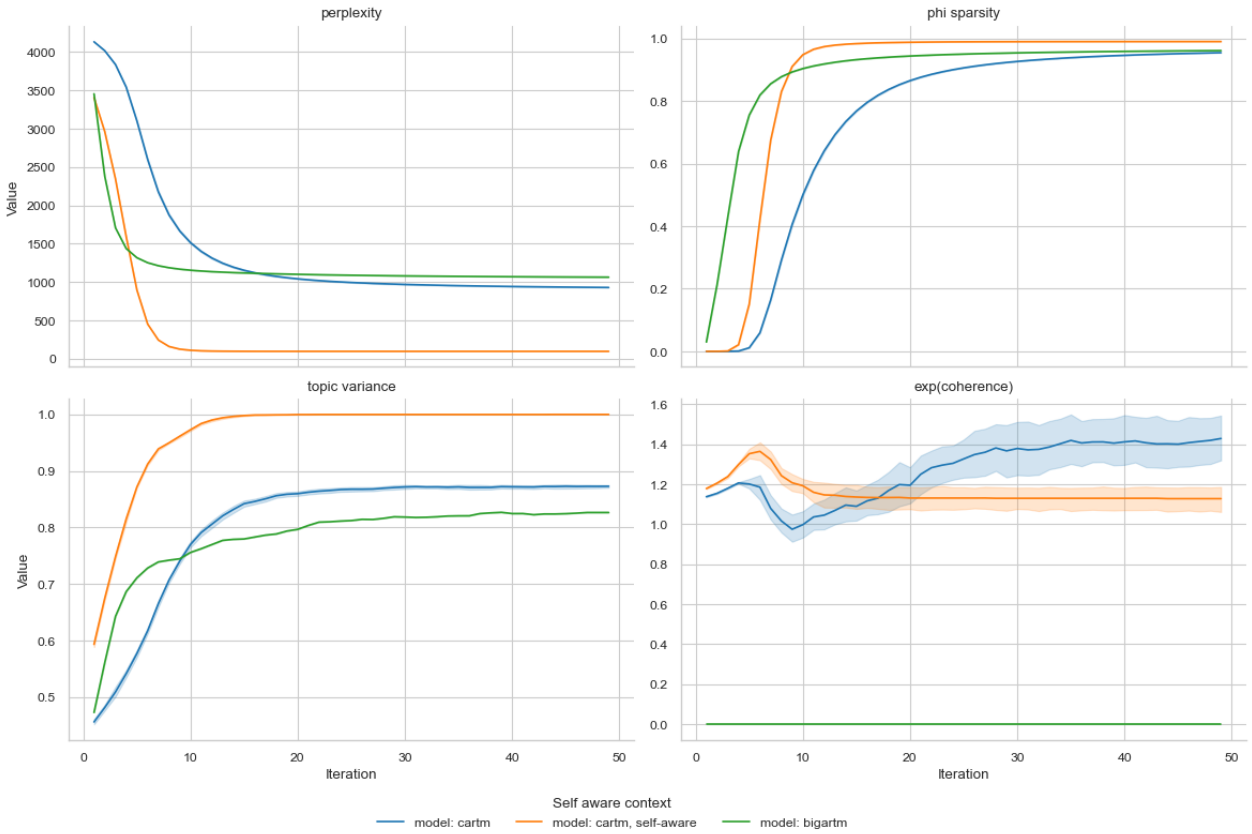

Рис. 5: Сравнение метрик моделей при разложении на 100 тем

Единственный общий множитель в этих асимптотиках — это количество тем, что оставляет большой простор для экспериментов в прикладных задачах при сравнении данных моделей. Например, как было показано в разделе 4.2, можно дополнительно уменьшать размер контекста, тем самым значительно ускоряя CARTM, при этом практически не теряя в качестве.

5 Архитектура модели

Модель написана на языке Python с использованием фреймворка JAX. Данный фреймворк поддерживает векторизацию вычислений и автоматическое вычисление градиента произвольной функции. Модель имеет открытый исходный код, размещенный по ссылке `https://github.com/revit3d/topic-modelling-attention`.

5.1 Формат данных

Модель использует формат фреймворка JAX, представляющий собой обертку над C-подобными массивами модуля numpy. Реализован предобработчик сырых данных `DatasetPreprocessor` для конвертации во внутреннее представление. Разбиение на батчи

также происходит с помощью специального класса `BatchLoader`. Данный класс автоматически разбивает предобработанные данные на батчи, причем можно регулировать размер одного батча.

Основной особенностью представления данных является то, что они хранятся в виде одномерного массива токенов. Это позволяет значительно ускорить вычисления за счет их векторизации. Восстановить границы документов позволяет вспомогательный одномерный массив, в котором хранятся индексы начала каждого документа.

5.2 Реализация итерационной схемы

Напомним, что в модели реализуется итерационная схема 9.

- Вычисление матрицы ϕ'_{tw} достигается покомпонентным умножением строк матрицы Φ на вектор $p(t)$ с последующей перенормировкой.
  Асимптотика: $O(WT)$.

- При вычислении матрицы θ_{it} для каждого терма в корпусе производится взвешенное усреднение тематических векторов слов, находящихся в его контексте.
  Асимптотика: $O(ICT)$.

- При вычислении матрицы p_{ti} строится в явном виде матрица $\phi_{w_i t}$, состоящая из строк матрицы Φ, отвечающих заданным термам w_i. Далее, покомпонентно перемножая матрицы и перенормируя результат, получаем итоговую матрицу.
  Асимптотика: $O(IT)$.

- Вектор n_t получается путем построчного сложения элементов матрицы p_{ti}.
  Асимптотика: $O(IT)$

- Обновление матрицы Φ происходит путем суммирования всех строк матрицы p_{ti}. Вычисление градиента происходит численно.
  Асимптотика: $O(IT)$.

Вычисление матрицы θ_{ti} является наиболее вычислительно-затратным шагом итерационной схемы. Для эффективного распараллеливания вычислений строится трехмерный тензор $I \times C \times T$, который представляет собой для каждой позиции i, для каждой темы t вектор контекста темы t слова на позиции i. Также именно на этом шаге происходит учет границы документов. Для этого строится трехмерный необучаемый тензор маски внимания $I \times C \times T$, с помощью которой затем вычисляются веса каждого слова в контексте. После этого тензоры покомпонентно умножаются и значения суммируются по оси контекста.

Рассмотрим построение маски внимания для случая $I = 10, C = 5$. Пусть имеется два документа, в первом документе 4 терма, во втором — 6. В начале строится одномерный ненормированный вектор весов термов в контексте.

Рис. 6: Пример ненормированных весов слов в контексте при $\gamma = 0.1$

Далее, с помощью массива границ документов строится бинарная маска внимания для учета границ документов. Теоретически, ничто не ограничивает использование данного массива лишь для обозначения границ документов. Возможно добавление произвольных границ. Это, в свою очередь, означает, что можно перейти от границ документов к границам предложений или к случайному разбиению на отрывки с фиксированным количеством слов.

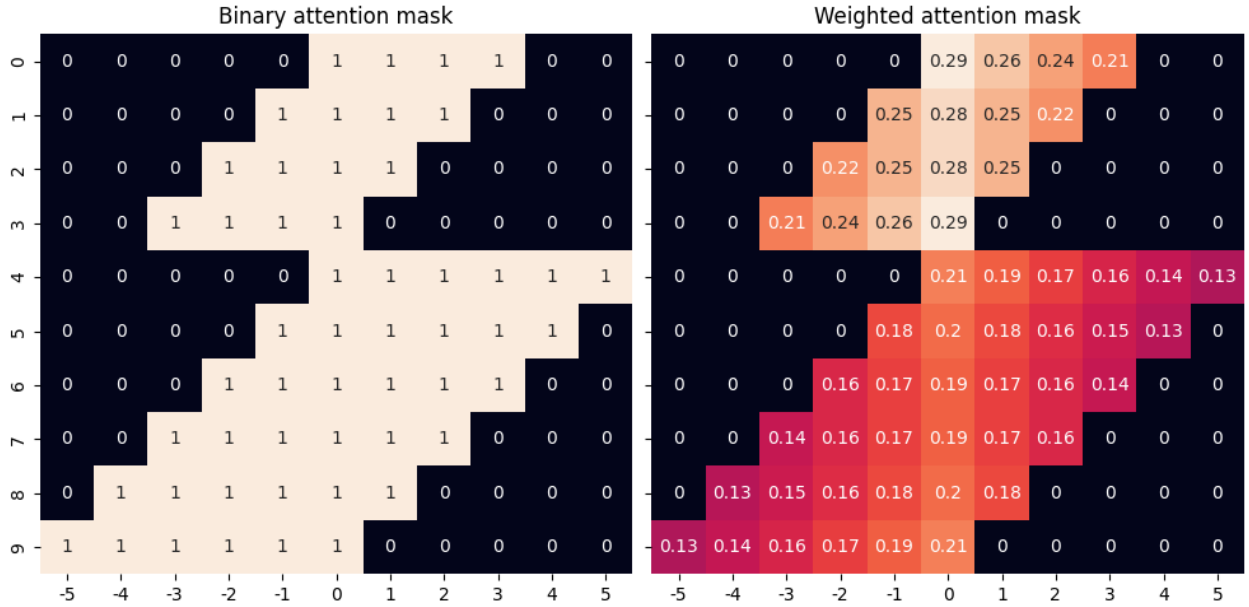

Рис. 7: Бинарная маска внимания (слева) и взвешенная маска внимания (справа)

Далее, бинарная маска внимания построчно перемножается с вектором весов и перенормируется. В результате получается взвешенная маска внимания, учитывающая как границы документов, так и расстояние между словами в контексте.

5.3 Метрики и регуляризаторы

Модель имеет встроенные метрики перплексии, разнообразия тем, разреженности матрицы Φ и когерентности. Реализована поддержка регуляризаторов, и написаны разреживающий и декоррелирующий регуляризаторы. Также модель поддерживает создание пользовательских метрик и регуляризаторов.

Метрики и регуляризаторы реализованы по схожему принципу. Имеются базовые классы, от которых можно унаследоваться и описать функционал метрики/регуляризатора. При этом, благодаря автоматическому вычислению градиентов, не требуется в явном виде дифференцировать функционал регуляризатора.

6 Выводы

В работе была предложена и исследована контекстная тематическая модель, основанная на механизме внимания с фиксированными весами. Модель позволяет учитывать локальный контекст слов при построении тематических распределений, что является преимуществом по сравнению с классическими подходами, основанными на гипотезе мешка слов. В отличие от классических моделей, предложенная модель учитывает относительное положение слов в тексте, что позволяет более точно определять их тематическую принадлежность.

Доказана эквивалентность перплексии между классической и контекстной моделями при совпадении контекста с документом. Показано, что кластеры документов могут быть восстановлены из контекстной модели с помощью преобразования тематических распределений. Экспериментально исследованы свойства модели. Установлено, что включение целевого слова в собственный контекст значительно улучшает качество модели (перплексия, когерентность). Показано, что уменьшение параметра затухания весов γ и увеличение размера контекста улучшают интерпретируемость тем. Эксперименты показали, что обучение на батчах позволяет ускорить сходимость без значительной потери качества.

Проведено сравнение с BigARTM: описанная модель продемонстрировала лучшее качество по метрикам перплексии и когерентности. Несмотря на более высокую вычислительную сложность, модель сохраняет практическую применимость благодаря возможности регулирования размера контекста. Реализована эффективная архитектура на основе JAX, обеспечивающая векторизацию вычислений и автоматическое дифференцирование.

Таким образом, предложенная модель представляет собой эффективную альтернативу классическим тематическим моделям, сочетающую преимущества контекстного подхода с интерпретируемостью и масштабируемостью.

7 Перспективы дальнейших исследований

- Оптимизация вычислительной эффективности модели (использование GPU, более эффективные алгоритмы обработки контекста).
- Исследование адаптивных механизмов внимания.
- Исследование применимости модели для задач классификации, кластеризации и генерации текстов.

Список литературы

[1] Blei D.M., Ng A.Y., Jordan M.I. Latent Dirichlet Allocation // Journal of Machine Learning Research, 2003. Vol. 3. P. 993-1022.

[2] Воронцов К.В. Аддитивная регуляризация тематических моделей // Доклады Академии наук, 2014. Т. 455, № 3. С. 268-271.

[3] Mimno D., Wallach H.M., Talley E. et al. Optimizing Semantic Coherence in Topic Models // Proceedings of EMNLP, 2011. P. 262-272.

[4] Devlin J., Chang M.W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // NAACL-HLT, 2019.

[5] Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need // NIPS, 2017. P. 5998-6008.

[6] Hofmann T. Probabilistic Latent Semantic Analysis // UAI, 1999. P. 289-296.

[7] Wallach H.M., Murray I., Salakhutdinov R., Mimno D. Evaluation Methods for Topic Models // ICML, 2009. P. 1105-1112.

[8] Bianchi F., Terragni S., Hovy D. et al. Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence // ACL, 2021.

[9] Dieng A.B., Ruiz F.J.R., Blei D.M. Topic Modeling in Embedding Spaces // TACL, 2020. Vol. 8. P. 439-453.

[10] Arora S., Ge R., Halpern Y. et al. A Practical Algorithm for Topic Modeling with Provable Guarantees // ICML, 2013. P. 280-288.

[11] Zhao H., Phung D., Huynh V. et al. Topic Modeling Meets Deep Neural Networks: A Survey // IJCAI, 2021.

[12] Николенко С.И., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. СПб.: Питер, 2017. 480 с.

[13] Tu Y., Wang W., Zeng M. et al. A Novel Context-Aware Topic Model for Statistical Machine Translation // AAAI, 2019. Vol. 33. P. 7143-7150.

[14] Boyd-Graber J., Blei D.M. Syntactic Topic Models // arXiv preprint arXiv:1202.3903, 2012.

[15] Gerlach M., Peixoto T.P., Altmann E.G. A Network Approach to Topic Models // Science Advances, 2018. Vol. 4, no. 7.