Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»
при Вычислительном центре им. А. А. Дородницына РАН

Сафин Камиль Фанисович

Вариационная модель внимания в сетях глубокого обучения

010900 — Прикладные математика и физика

Магистерская диссертация

Научный руководитель:
к. ф.-м. н.
Чехович Юрий Викторович

Москва
2019 г.

Содержание

Аннотация

В данной работе рассматривается модель вариационного механизма внимания, которая является объединением двух популярных подходов, которые позволяют решать многие задачи с высокой точностью: вариационного кодировщика и механизма внимания. Предлагаемая модель объединяет в себе преимущества обоих методов, и, как результат, имеет более высокое качество работы. Для построения модели вводится дополнительное пространство скрытых векторов, которое будет отвечать за вариационный механизм внимания. В связи с этим, получена уточненная нижняя оценка правдоподобия модели. В качестве априорного распределения предлагается использовать распределения Гумбеля. Использование этого распределения позволяет получать интерпретируемые вектора механизма внимания. Предложенная модель была протестирована на задаче машинного перевода и показала лучшее качество, чем базовые модели.

1 Введение

Актуальность работы. Одна из важных задач в области обработки естественного языка — это генерация качественных и согласованных предложений в рамках решения задачи. Многие модели, используемые для решения задач обработки текстов, генерируют ответ слово за словом, что сказывается на качестве получаемых предложений.

В данной работе предлагается модель, объединяющая в себе два подхода, которые используются для решения вышеописанной проблемы. Один из подходов заключается в использовании вариационного кодировщика для получения векторного представления входной последовательности [12]. Данный метод позволяет получить векторное представление входящей последовательности в целом. Второй подход состоит в использовании механизма внимания при генерации выходной последовательности [16]. Этот метод позволяет учитывать контекст на этапе генерации. Как результат, получаемые предложения получаются более согласованными и качественными.

Цель работы. Целью данной работы является построение модели, объединяющей два эффективных метода решения задач обработки естественного языка: вариационный кодировщик и механизм внимания.

Методы исследования. Для достижения поставленной цели предлагается доработать модель вариационного кодировщика путем введения дополнительного пространства скрытых переменных, отвечающих за механизм внимания. С учетом этого, необходимо также получить уточненную нижнюю оценку правдоподобия модели.

Основные положения, выносимые на защиту.

1. Вариационный механизм внимания.

2. Сравнение моделей, использующих разные вариационные распределения.

3. Теорема о разложении правдоподобия модели при введении дополнительного вариационного пространтсва.

Научная новизна. Предложена модель вариационного внимания с семплируемыми весами состояний. Доказана теорема, позволяющая производить настройку параметров модели вариационного внимания.

Практическая значимость. Предложенный в работе алгоритм позволяет генерировать более качетсвенные и согласованные предложения и, как следствие, решать задачи обработки ествественного языка на более высоком уровне.

Степень достоверности и апробация работы. Достоверность результатов подтверждена экспериментальной проверкой предлагаемой модели на реальных задачах.

Публикации по теме дипломной работы. Результаты работ по схожей тематике представлены в следующих статьях:

1. R. Kuznetsova, O. Bakhteev, A. Ogaltsov, K. Safin *VBTA: Variational Bi-domain Triplet Autoencoder for learning across domains with the relative constraints.* In review at NeurIPS 2019.

2. K. Safin, A. Ogaltsov. *Detecting a change of style using text statistics.* Proceedings on CLEF, 2018.

3. K. Safin, R. Kuznetsova. *Style Breach Detection with Neural Sentence Embeddings.* Proceedings on CLEF, 2017.

Обзор литературы. Вариационный кодировщик [12] отображает входную последовательность данных в скрытое пространство случайных векторов заданной структуры. Полученный вектор из скрытого пространства, называемый *вектором контекста* подается на вход декодировщику, который генерирует выходную последовательность данных. Классические модели кодировщик-декодировщик [10], не используют пространств случайных векторов, т.е. являются детерминированными. Недостаток этих моделей состоит в том, что сгенерированные данные обладают достаточно малой дисперсией [8]. Это влечет за собой низкое интегральное качество итговой модели. Вариационные модели отображают входные данные не в конкретную точку скрытого пространства, а в некоторую окрестность, что позволяет получать более разнообразные и качественные выходные данные и даже контролировать процесс ге-

нерации данных [19].

В задачах обработки естественного языка в качества кодировщика и декодировщика обычно используют *реккурентные нейронные сети*. Существенным недостатком таких моделей является то, что при увеличении длины входной последовательности слов, качество декодирования существенно снижается[17], если модель имеет малый размер. Одним из способов устранить этот недостаток является использование в реккурентных сетях *механизма внимания*. Механизм внимания определяет степень релевантности каждого вектора входной последовательлности на каждом этапе декодирования. Механизм внимания [4], [16], [1] значительно повышает качество и на данный момент такие модели с механизмом внимания являются state-of-the-art решениями многих задач.

Довольно логично было бы объединить эти подходы. Однако построение таких моделей изучено слабо [14], [7]. Связано это в первую очередь с тем, что необходимо объединить детерминированный механизм внимания и случайную модель вариационного кодировщика. Показано [20], что при обучении совместной модели происходит «деградация» скрытого пространства и модель обучается как обычный кодировщик с механизмом внимания.

2 Постановка задачи

Пусть задана выборка:

$$\mathfrak{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\},\ i = \overline{1, N},$$

состоящая из множества пар объектов из разных доменов $\boldsymbol{x}_i \in \boldsymbol{X}$, $\boldsymbol{y}_i \in \boldsymbol{Y}$. Выборка разбита на обучающую $\mathfrak{D}_l$ и контрольную $\mathfrak{D}_t$. Объекты в паре соответсвуют друг другу, то есть представляют разные интерпретации одного и того же объекта, например:

- Переводы одной фразы на разные языки;
- Изображения одного предмета с разных ракурсови т.п.

Задача состоит в том, чтобы построить алгоритм F, который по заданному объекту $\boldsymbol{x} \in \boldsymbol{X}$ получает объект из другого кластера $\hat{\boldsymbol{y}} \in \boldsymbol{Y}$ так, чтобы полученный $\hat{\boldsymbol{y}}$ и истинный $\boldsymbol{y}$ объекты были наиболее похожи друг на друга:

$$F:\ \boldsymbol{x} \longrightarrow \hat{\boldsymbol{y}},\ S(\hat{\boldsymbol{y}}, \boldsymbol{y}) \to \max,\ \boldsymbol{x} \in \boldsymbol{X}, \hat{\boldsymbol{y}} \in \boldsymbol{Y}, \tag{2.1}$$

где мера сходства объектов S выбирается исходя из конкретной задачи. Например, это может быть:

- BLEU метрика для задач перевода фраз;
- попиксельное сравнение для задач перевода изображений и т.п.

3 Построение векторных представлений

Вариационный кодировщик позволяет получать векторные представления объектов в целом. Механизм внимания, в свою очередь, выделяет части входного объекта для более детальной обработки. Объяснить суть работы механизма внимания проще в случае, когда объекты $\boldsymbol{x}$ и $\boldsymbol{y}$ являются последовательностями векторов: $\boldsymbol{x} = \{\mathbf{x}_1\ \ldots, \mathbf{x}_{|\boldsymbol{x}|}\}$, $\boldsymbol{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{\boldsymbol{y}}\}$.

3.1 Механизм внимания

Механизм внимания — подход, который позволяет оценить степень релевантности каждого вектора входящей последовательности в процессе декодирования.

Простой кодировщик. Рассмотрим сначала простую модель кодировщика без использования механизма внимания. Эта модель выглядит следующим образом. Пусть входная последовательность векторов: $\boldsymbol{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\boldsymbol{x}|}\}$. Кодировщик отображает эту последовательность в последовательность векторов:

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}),\ t = \overline{1, |\boldsymbol{x}|}. \tag{3.1}$$

Тогда векторное представление всей последовательности, получаемое с помощью кодировщика есть:

$$\mathbf{c} = g\left(\{\mathbf{h}_1, \ldots, \mathbf{h}_{|\boldsymbol{x}|}\}\right),$$

где f и g — некоторые нелинейные функции. Чаще всего они моделируются нейронными сетями. Например, в качестве f используют LSTM (Long-Short Term Memory) [18] или GRU (Gated Reccurent Unit) архитектуру [15]. А в качестве g: $g\left(\{\mathbf{h}_1, \ldots, \mathbf{h}_{|\hat{\mathbf{x}}|}\}\right) = \mathbf{h}_{|\boldsymbol{x}|}$. То есть в качестве векторного представления всей последовательности $\boldsymbol{x}$ берется последний полученный вектор $\mathbf{h}_{\boldsymbol{x}}$

Декодировщик генерирует выходную последовательность $\boldsymbol{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_{|\boldsymbol{y}|})$. Каждый последующий вектор декодировщик генерирует на основе векторного представления входной последовательности и всех предыдущих сгенерированных векторов. Т.е.:

$$p(\boldsymbol{y}) = \prod_{t=1}^{T} p\left(\mathbf{y}_t | \{\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{c}\}\right).$$

В случае использования реккурентных нейронных сетей, каждое условное распределение есть некоторая нелинейная функция q:

$$p\left(\mathbf{y}_t | \{\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}, \mathbf{c}\}\right) = q(\mathbf{y}_{t-1}, \mathbf{c}). \tag{3.2}$$

Классический механизм внимания. При использовании механизма внимания, немного модернизируется процесс генерации выходной последовательности. Если в случае простого кодировщика 3.2 каждое условное распределение моделируется функцией предыдущего полученного вектора $\mathbf{y}_{t-1}$ и векторного представления последовательности $\mathbf{c}$, то при использовании механизма внимания, данная условная

вероятность записывается следующим образом:

$$p\left(\mathbf{y}_t|\{\mathbf{y}_1,\ldots,\mathbf{y}_{t-1},\mathbf{c}\}\right) = q(\mathbf{y}_{t-1},\mathbf{c}_t),$$

где $\mathbf{c}_t$ есть вектор контекста, рассчитываемый заново на каждом шаге декодирования. Вектор контекста $\mathbf{c}_t$ есть взвешенная комбинация векторов $\{\mathbf{h}_1,\ldots,\mathbf{h}_{|\boldsymbol{x}|}\}$, в которую кодировщик отображает входную последовательность 3.1

$$\mathbf{c}_t = \sum_{j=1}^{|\hat{\mathbf{x}}|} \alpha_{tj}\mathbf{h}_t. \tag{3.3}$$

Веса α_{tj}, с которыми суммируются вектора $\mathbf{h}_t$ рассчитываются как:

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum\limits_{i=1}^{|\boldsymbol{x}|}\exp(e_{ti})},$$

где e_{tj} — веса до нормализации, которые определяют модель выравнивания. Данная нормализация необходима для того, чтобы веса векторов на данном этапе суммировались в единицу. Одним из преимуществ этого является интерпретируемость весов. То есть на каждом шаге декодирования можно проследить, насколько каждый вектор входной последовательности $\boldsymbol{x}$ влияет на декодирование текущего вектора $\mathbf{y}_t$. Величина e_{tj} есть функция $e_{tj} = a(\mathbf{y}_{t-1},\mathbf{h}_j)$, которая оценивает, насколько элемент входящей последовательности на позиции j важен для декодирования вектора $\mathbf{y}_t$. Данная функция обычно моделируется нейронной сетью.
На рисунке 3.1 схематически представлена модель классического внимания.

3.2 Вариационная оценка

Как говорилось выше, вариационный кодировщик позволяет получать качественные векторные представления объектов. Для простоты, сначала рассмотрим сначала задачу восстановления объекта $\boldsymbol{x} \in \boldsymbol{X}$ по его векторному представлению, полученному с помощью вариационного кодировщика.

Пусть данные $X = \bigcup\limits_i \boldsymbol{x}_i \subset \boldsymbol{X}$ имеют структуру, обусловленную некоторой скрытой переменной $\boldsymbol{z} \in Z$. Обозначим распределение этой переменной: $\boldsymbol{z} \sim P(z)$. Введем набор детерминированных функций, параметризованных вектором $\theta \in \Theta$, которые осуществляют отображения из пространства Z в пространство $\boldsymbol{X}$: $f(\boldsymbol{z},\theta):\ Z\times\Theta\rightarrow$

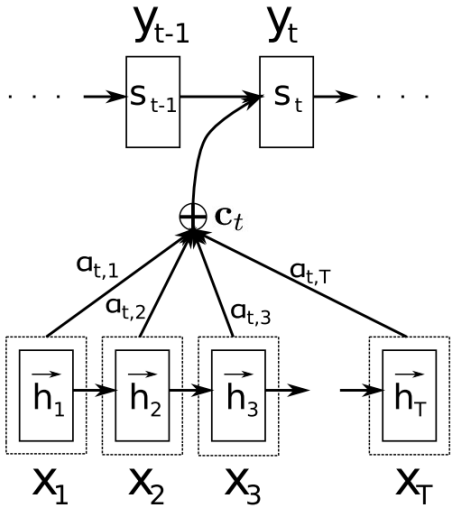

Рис. 1: Классический механизм внимания

$\boldsymbol{X}$. Если $\boldsymbol{z}$ — случайная переменная, то и $f(\boldsymbol{z}, \theta)$ — случайная переменная в пространстве $\boldsymbol{X}$.

С учетом этого, запишем правдоподобие данных X:

$$P(X) = \int P(X|\boldsymbol{z}, \theta) P(\boldsymbol{z}) d\boldsymbol{z}.$$

Здесь $f(\boldsymbol{z}, \theta)$ заменено на распределение $P(X|\boldsymbol{z}, \theta)$. Причем в качестве распределения берется: $P(X|\boldsymbol{z}, \theta) = \mathcal{N}(X|f(\boldsymbol{z}, \theta), \sigma^2 \cdot I)$. Введем $Q(\boldsymbol{z}|X)$ — функцию, которая по данным X выдает рапределение над переменными $\boldsymbol{z}$, из которых вероятнее сгенерируется X.

Дивергенция Кульбака-Лейблера выражается как:

$$\mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}|X)] = E_{\boldsymbol{z}\sim Q}[\log Q(\boldsymbol{z}) - \log P(\boldsymbol{z}|X)]. \tag{3.4}$$

Из правила Байеса можно получить:

$$\log P(\boldsymbol{z}|X) = \frac{P(X|\boldsymbol{z}) \cdot P(\boldsymbol{z})}{P(X)} = \log P(X|\boldsymbol{z}) + \log P(\boldsymbol{z}) - \log P(X). \tag{3.5}$$

Тогда, с учетом 3.4 и 3.5, для дивергенции получаем:

$$\mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}|X)] = E_{\boldsymbol{z}\sim Q}[\log Q(\boldsymbol{z}) - \log P(\boldsymbol{z}|X)] = E_{\boldsymbol{z}\sim Q}[\log Q(\boldsymbol{z}) - \log P(X|z) - \log P(\boldsymbol{z})] + \log P(X).$$

И далее:

$$\log P(X) - \mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}||X)] = E_{\boldsymbol{z}\sim Q}[\log P(X|\boldsymbol{z}) - \mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z})]].$$

В целом, Q — любое распределение над $\boldsymbol{z}$, но чтобы минимизировать $\mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}||X)]$, логично, чтобы оно зависело от X: $Q(\boldsymbol{z}) = Q(\boldsymbol{z}|X)$. В итоге получаем:

$$\log P(X) - \mathcal{D}[Q(\boldsymbol{z}||X)||P(\boldsymbol{z}||X)] = E_{\boldsymbol{z}\sim Q}[\log P(X|\boldsymbol{z})] - \mathcal{D}[Q(\boldsymbol{z}|X)||P(\boldsymbol{z})].$$

Из полученного равенства легко получить нижнюю оценку правдоподобия модели (Evidence Lower Bound):

$$\log P(X) \geqslant E_{\boldsymbol{z}\sim Q}[\log P(X|\boldsymbol{z})] - \mathcal{D}[Q(\boldsymbol{z}|X)||P(\boldsymbol{z})] = \mathcal{L}. \tag{3.6}$$

Теперь рассмотрим случай, когда данные представлены двумя доменами. В этом случае рассуждения остаются теми же, за тем исключением, что распределение вектора $\boldsymbol{z}$ теперь зависит не только от X: $Q(\boldsymbol{z}) = Q(\boldsymbol{z}|X,Y)$, $Y = \bigcup\limits_i \boldsymbol{y}_i$. Это вносит некоторые трудности, так как в процессе обучения, мы знаем $\boldsymbol{y}$ и можем получить совместное распределение, однако в процессе переноса объекта из $\boldsymbol{X}$ в $\boldsymbol{Y}$, информация об $\boldsymbol{y}$ уже не известна. В этом случае делается предположение [21], что $\boldsymbol{y}$ есть функция $\boldsymbol{x}$: $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})$. Предположение, в целом, обоснованно, т.к. целью разрабатываемого алгоритма и является построение такой функции 2.1. В таком случае получаем: $Q(\boldsymbol{z}) = Q(\boldsymbol{z}|X,Y(X)) = Q(\boldsymbol{z}|X)$. А нижняя оценка правдоподобия видоизменяется следующим образом:

$$E_{\boldsymbol{z}\sim Q}[\log P(Y|\boldsymbol{z})] - \mathcal{D}[Q(\boldsymbol{z}|X)||P(\boldsymbol{z})] = \mathcal{L}. \tag{3.7}$$

$E_{\boldsymbol{z}\sim Q}[\log P(X|\boldsymbol{z})]$ заменяется на $E_{\boldsymbol{z}\sim Q}[\log P(Y|\boldsymbol{z})]$, потому что мы оцениваем правдоподобие получаемых данных в процессе переноса объектов из $\boldsymbol{X}$ в $\boldsymbol{Y}$, а не правдоподобие исходных данных.

3.3 Механизм внимания в вариационной оценке

Было отмечено [20], что добавление механизма внимания в неизменном виде в модель вариационного кодировщика приводит к тому, что скрытая переменная перестает аггрегировать информацию о входном объекте.

Теорема 3.1. *Оптимизация нижней оценки правдоподобия в модели вариационного кодировщика с механизмом внимания:*

$$\mathcal{L} = E_{\boldsymbol{z}\sim Q}[\log P(Y|\boldsymbol{z})] - \mathcal{D}[Q(\boldsymbol{z}|X)||P(\boldsymbol{z})],$$

равносильна минимизации первого слагаемого, при $Q(\boldsymbol{z}|X) \equiv P(\boldsymbol{z})$.

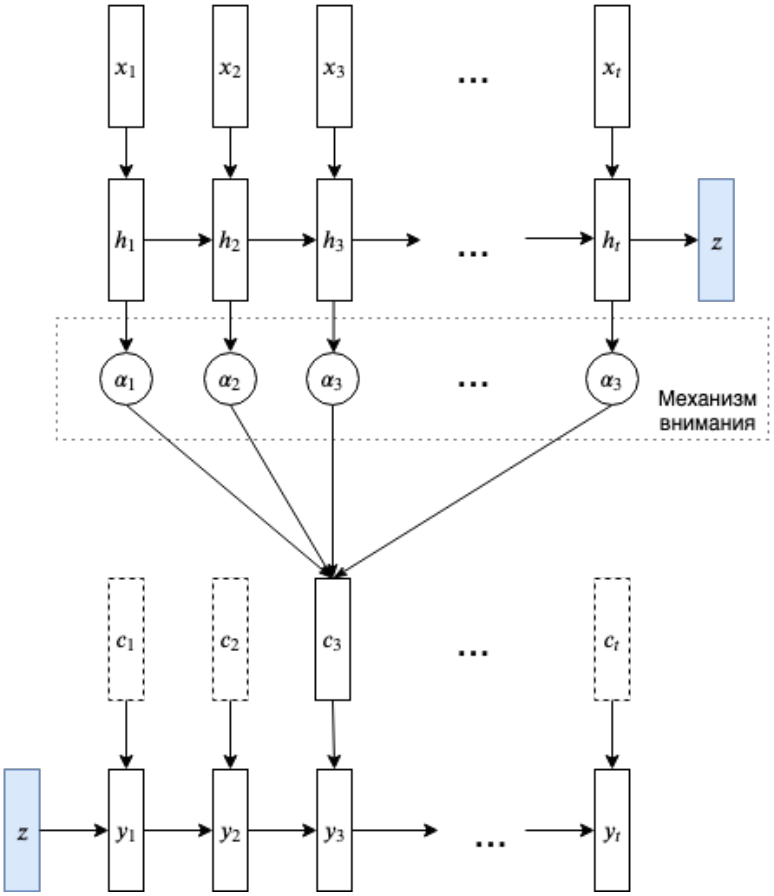

Рис. 2: Архитектура модели с механизмом внимания

При описании механизма внимания было получено, что:

$$p\left(\mathbf{y}_t|\{\mathbf{y}_1,\ldots,\mathbf{y}_{t-1},\mathbf{c}\}\right) = q(\mathbf{y}_{t-1},\mathbf{c}_t),$$

где $\mathbf{c}_t$ есть:

$$\mathbf{c}_t = \sum_{j=1}^{|\hat{\mathbf{x}}|} \alpha_{tj}\mathbf{h}_t,\ \ \mathbf{h}_t = f(\mathbf{x}_t).$$

То есть, можно сказать, что $\boldsymbol{y}$ есть некоторая функция $\boldsymbol{x}$: $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x})$. Эта функция моделируется с помощью механизма внимания, при этом скрытая переменная $\boldsymbol{z}$ в этой функции не играет никакой роли. Но тогда структура данных обусловлена не $\boldsymbol{z}$, а X. И правдоподобие данных записывается как: $\log[P(Y|X)]$. А так как распределение $\boldsymbol{z}$ не играет никакой роли, а дивергенция Кульбака-Лейблера неотрицательна, то можно положить $Q(\boldsymbol{z}|X) \equiv P(\boldsymbol{z})$, при котором второе слагаемое примет свое минимальное значение.

Для того, чтобы механизм внимания не оказывал воздействия на скрытую переменную, необходимо ввести дополнительное вариационное пространство, которое

будет содержать в себе параметры вариационного механизма внимания.

Теорема 3.2. *При введении дополнительного вариационного пространства скрытых векторов* $\mathbf{a}$*, нижняя оценка представима в виде:*

$$\mathcal{L} = E[\log p(Y|\boldsymbol{z}, \mathbf{a})] - \mathcal{D}_{KL}[q^{(\boldsymbol{z})}(\boldsymbol{z}|X)||p(\boldsymbol{z})] - \mathcal{D}_{KL}[q^{(\mathbf{a})}(\mathbf{a}|X)||p(\mathbf{a})].$$

Нижняя оценка присовместном распределении векторов $\boldsymbol{z}$ и $\boldsymbol{a}$ записывается следующим образом:

$$\mathcal{L} = E_{\boldsymbol{z},\boldsymbol{a}\sim Q(\boldsymbol{z},\boldsymbol{a}|X)}[\log P(X|\boldsymbol{z}, \boldsymbol{a})] - \mathcal{D}[Q(\boldsymbol{z}, \boldsymbol{a}|X)||P(\boldsymbol{z}, \boldsymbol{a})]$$

Т.к. $\boldsymbol{z}$ и $\boldsymbol{a}$ независимы друг от друга, то их совместное распределение факторизуется:

$$P(\boldsymbol{z}, \boldsymbol{a}) = P(\boldsymbol{z}) \cdot P(\boldsymbol{a})$$

$$Q(\boldsymbol{z}, \boldsymbol{a}) = Q(\boldsymbol{z}) \cdot P(\boldsymbol{a})$$

И нижняя оценка распадается на несколько слагаемых:

$$\mathcal{L} = E_{z\sim Q^{(z)}(z|X),a\sim Q^{(a)}(a|X)}[\log P(X|z, a)] - \mathcal{D}[Q^{(z)}(z|X)||P(z)] - \mathcal{D}[Q^{(z)}(a|X)||P(a)].$$

И в таком случае, нижняя оценка представляется в виде:

$$\mathcal{L} = E_{\boldsymbol{z},\boldsymbol{a}\sim Q(\boldsymbol{z},\boldsymbol{a}|X)}[\log P(X|\boldsymbol{z}, \boldsymbol{a})] - \mathcal{D}[Q(\boldsymbol{z}, \boldsymbol{a}|X)||P(\boldsymbol{z}, \boldsymbol{a})]$$

Механизм внимания с нормальным распределением Одним из способов задать структуру пространства векторов вариационного внимания является семплирование контекстного вектора $\mathbf{c}_t$ на каждом шаге декодирования, как это предлагается в [20]. В классическом механизме внимания контекстный вектор получается есть взвешенная сумма векторов $\left(\mathbf{h}_1, \ldots, \mathbf{h}_{|\hat{\mathbf{x}}|}\right)$, как показано в 3.3. При семплировании вектора контекста из нормального распределения, на данном этапе рассчитываются параметры этого распределения:

$$\boldsymbol{\mu}_t = u\left(\mathbf{h}_1, \ldots, \mathbf{h}_{|\hat{\mathbf{x}}|}\right), \quad \boldsymbol{\sigma}_t^2 = v\left(\mathbf{h}_1, \ldots, \mathbf{h}_{|\hat{\mathbf{x}}|}\right).$$

Как правило, параметры распределения $\boldsymbol{\mu}$ и $\boldsymbol{\sigma}^2$ являются выходами нейронной сети — как и в данной работе.

Однако, семплирование напрямую из нормального распределения с полученными параметрами $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2)$ [6], [2] не позволяет посчитать градиент, чтобы запустить

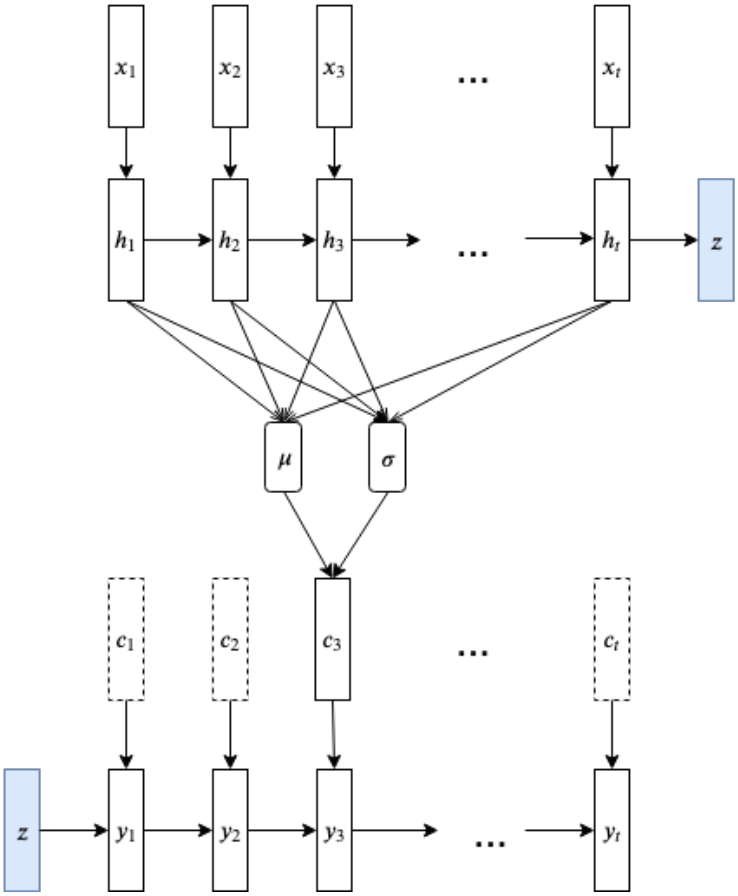

Рис. 3: Вариационный механизм внимания с семплированием из нормального распределения

алгоритм обратного распространения ошибки для настройки оптимальных параметров алгоритма. Для того, чтобы появилась возможность рассчитать градиент, используется так называемый reparametrization trick [12]. Его смысл заключается в том, чтобы семплировать не напрямую из распределения $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2)$, а сначала получить семпл из стандартного нормального распределения $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1)$. И для получения вектора из рассматриваемого распределения $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2)$, следует учесть $\boldsymbol{\mu}_t$, $\boldsymbol{\sigma}_t^2$, рассчитанные заранее:

$$\boldsymbol{c}_t = \boldsymbol{\mu}_t + \boldsymbol{\varepsilon} \cdot \boldsymbol{\sigma}^2 \tag{3.8}$$

Полученный таким образом вектор контекста $\boldsymbol{c}_t$ из распределения $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2)$ позволяет использовать алгоритм обратного распространения ошибки.

Механизм внимания с распределением Гумбеля Предложенный в [20] способ добавления механизма внимания в вариационный кодировщик устраняет проблему «деградации» вариационного пространства. Однако существуют другие способы задать структуру пространства векторов механизма внимания. К тому же, при семпли-

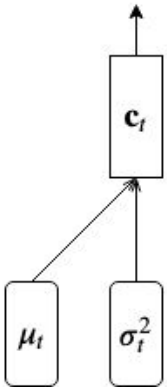

Рис. 4: Прямое семплирование

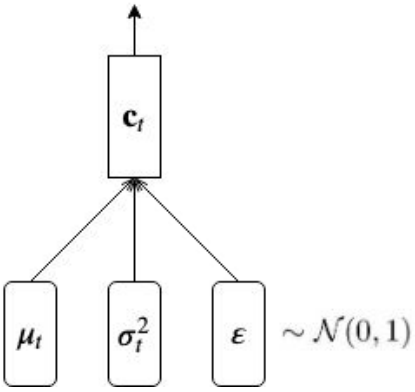

Рис. 5: Семплирование с репараметризацией

ровании из нормального распределения теряется возможность интерпретации векторов контекста, как это делается в классическом механизме внимания.

Поэтому предлагается семплировать не вектора контекста $\boldsymbol{c}_t$, а сами веса α_{tj} из 3.3. Нужно использовать распределение, при семплировании из которого можно получать вектора, компоненты которого интерпретируются как нормализованные веса. При этом, это распределение должно иметь свойство, позволяющее прозвести преобразование, аналогичное reparametrization trick 3.8 В качестве такого распределения в вариационных кодировщиках часто используется распределение Гумбеля [9], [11]. Свойства распределения Гумбеля, которые важны для последующих выводов:

- Связь с равномерным распределением:

$$G(0,1) = -\log(-\log U(0,1)),$$

  где $G(0,1)$ – стандартное распределение Гумбеля, $U(0,1)$ — стандартное равномерное распределение;

- Reparametrization trick:

$$\alpha_t = \frac{\exp((\log \pi_t + g_t)/\tau)}{\sum\limits_{j=1}^{k} \exp((\log \pi_j + g_j)/\tau)}, \; g_i \in G(0,1),$$

  где π_i — ненормализованные веса, τ — параметр семплирования, от которого зависит вид распределения.

Таким образом, используя распределения Гумбеля, можно получать веса векторов, с которыми они складываются для получения вектора контекста 3.3. Если

в случае в семплированием из нормального распределения, рассчитываемыми параметрами были $\boldsymbol{\mu_t}$ и $\boldsymbol{\sigma}_t^2$, то в данном случае, при использовании распределения Гумбеля, рассчитываемыми параметрами будут являться ненормализованные веса π_{jt}.

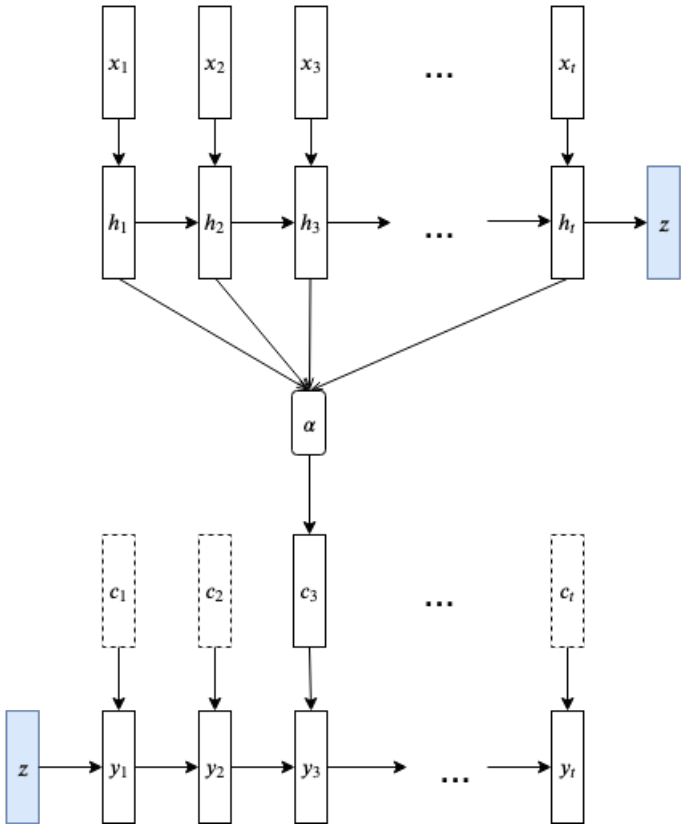

Рис. 6: Вариационный механизм внимания с семплированием из распределения Гумбеля

4 Вычислительные эксперименты

Для иллюстрации работы предложенных алгоритмов была выбрана задача машинного перевода для пары языков английский-французский и проведен вычислительный эксперимент на данных Europarl [13]. Корпус состоит из параллельных реплик заседаний Европарламента на различных языках, в данном случае используются реплики на английском и французском языках.

Меры качества. В качестве меры качества машинного перевода используется показатель BLEU (Bilingual Evaluation Understudy) [3]. BLEU-n показывает мощность

пересечения множества n-грамм истинного перевода (T) и машинного перевода (M):

$$\text{BLEU-n} = \frac{|set(ngram_M)|}{|set(ngram_T)|}$$

Для измерения качества сгенерированных предложений используются показатели энтропии (Entropy) и разнообразия (Distinct) [5]. Энтропия расчитывается как:

$$\text{Entropy} = \sum_{\omega} p(\omega) \log p(\omega),$$

где $p(\omega)$ — вероятность (частота) появления слова ω. Distinct метрика рассчитывается как доля различных униграмм (одиночных слов) и биграмм (последовательных двух слов) — Dist-1 и Dist-2 соответсвенно:

$$\text{Distinct-1} = \frac{|\text{set}(unigram_i)|}{|\bigcup_i unigram_i|}, \ \text{Distinct-2} = \frac{|\text{set}(bigram_i)|}{|\bigcup_i bigram_i|}$$

Модели. Для сравнения предлагаемой архитектуры с существующими подходами, было выбрано несколько моделей для обучения:

- Простая модель кодировщик-декодировщик (Simple AE);
- Модель кодировщик-декодировщик с классическим механизмом внимания (AE with Classic Attention);
- Кодировщик с вариационным механизмом внимания с семплированием по Гауссу (Var Attention, Normal dist);
- Кодировщик с вариационным механизмом внимания с семплированием по Гумбелю (Var Attention, Gumbel dist);

Все модели были подробно описаны в предыдущих разделах работы.

Результаты. В таблице 1 приведены показатели BLEU для всех моделей. Видно, что кодировщики с вариационным механизмом внимания демонстрируют более высокое качество перевода по всем показателям. Причем модель кодировщика с вариационным механизмом внимания, в котором сеплирование весов происходит по Гумбелю показывает лучшие результаты.

Такие же выводы можно сделать из таблицы 2 относительно показателей энтропии и разнообразия. Добавление механизма внимания повышает разнообразие генерируемых предложений, что напрямую сказывается на качестве перевода.

Таблица 1: BLEU качество перевода

	BLEU-4	BLEU-3	BLEU-2	BLEU-1
Simple AE	24,3	13,5	8,1	4,9
AE with Classic Attention	34,2	22,3	15,14	10,17
Var Attention, Normal dist.	34,7	22,6	15,3	10,2
Var Attention, Gumbel dist.	35.3	23.4	16.1	11.0

Таблица 2: Показатели Entropy и Distinct

	Entropy	Distinct-1	Distinct-2
Simple AE	2,2	0,073	0,085
AE with Classic Attention	2,4	0,083	0,093
Var Attention, Normal dist.	2,5	0,116	0,173
Var Attention, Gumbel dist	2,6	0.126	0.195

Интерпретируемость компонент. Как было сказано выше, при семплировании по Гумбелю в вариационном механизме внимания, компоненты вектора можно интерпретировать, как и в классическом механизме внимания. То есть, каждая из компонент вектора отображает то, с каким весом данное слове вносит вклад в генерацию текущего слова. На рис.7 представлены визуализации векторов внимания на двух примерах.

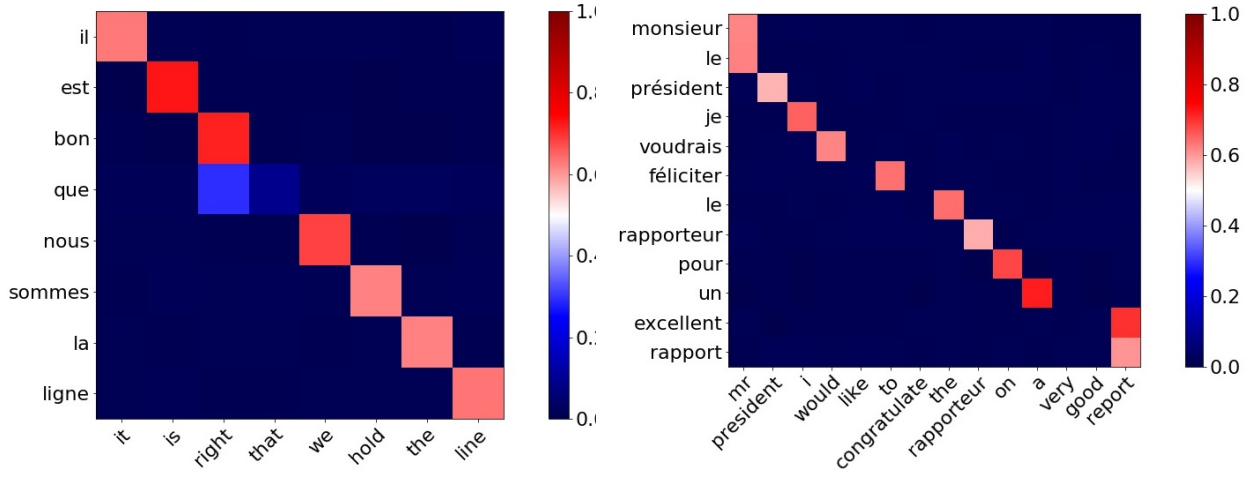

Рис. 7: Примеры визуализации векторов внимания при семплировании из распределения Гумбеля

Для сравнения на рис.8 приведены примеры визуализации весов при использовании классического механизма внимания.

Видно, что механизм внимания с распределением Гумбеля выборочен. То есть для каждого слова выбирается одно слово, которое оказывает наибольшее влияние на перевод. В случае с классическим механизмом внимания рпассматриваются слова в некоторой окрестности переводимого слова.

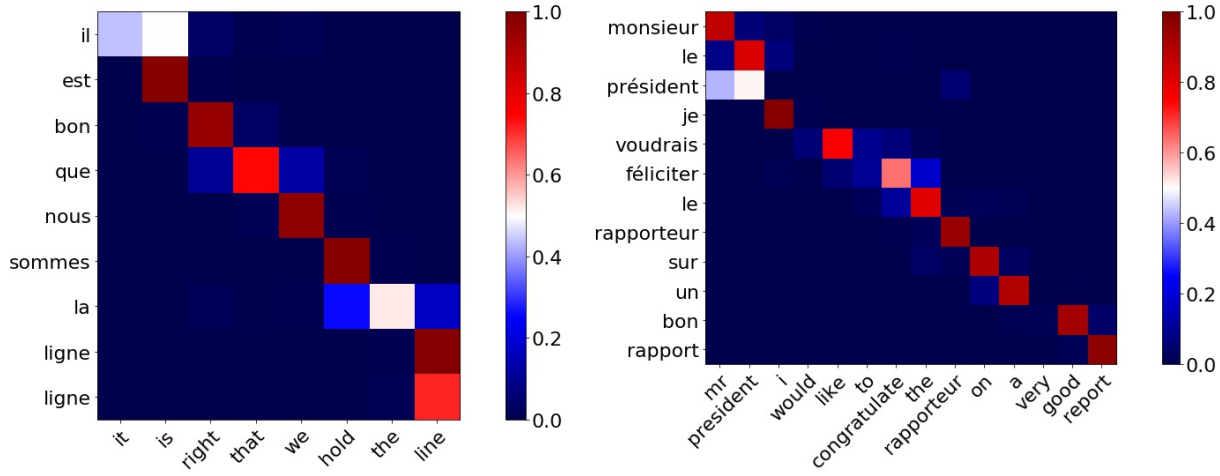

Рис. 8: Примеры визуализации векторов внимания при использовании классического механизма внимания

5 Заключение

Основные результаты работы

- Предложен и протестирован вариационный механизм внимания с семплированием весов из распределения Гумбеля.
- Произведено сравнение моделей, использующих разные вариационные распределения.
- Доказана теорема о разложении правдоподобия модели при введении дополнительного пространства.

Список литературы

[1] Attention-based models for speech recognition / J. Chorowski, D. Bahdanau, D. Serdyuk et al. // NIPS. — 2015.

[2] Automatic differentiation variational inference / A. Kucukelbir, D. Tran, R. Ranganath et al. // *Journal of Machine Learning Research.* — 2017. — Vol. 18. — Pp. 14:1–14:45.

[3] Bleu: a method for automatic evaluation of machine translation / K. Papineni, S. Roukos, T. Ward, W. jing Zhu. — 2002. — Pp. 311–318.

[4] *Cho K., Courville A. C., Bengio Y.* Describing multimedia content using attention-based encoder-decoder networks // *IEEE Transactions on Multimedia.* — 2015. — Vol. 17. — Pp. 1875–1886.

[5] A diversity-promoting objective function for neural conversation models / J. Li, M. Galley, C. Brockett et al. // HLT-NAACL. — 2016.

[6] *Doersch C.* Tutorial on variational autoencoders // *CoRR.* — 2016. — Vol. abs/1606.05908.

[7] *Dupont E.* Learning disentangled joint continuous and discrete representations. — 2018. — 03.

[8] Generating sentences from a continuous space / S. R. Bowman, L. Vilnis, O. Vinyals et al. // CoNLL. — 2016.

[9] *Gumbel E. J.* Statistical theory of extreme values and some practical applications: a series of lectures // *US Govt. Print. Office.* — 1954. — Vol. Number 33.

[10] *Hinton G., Salakhutdinov R.* Reducing the dimensionality of data with neural networks // *Science (New York, N.Y.).* — 2006. — 08. — Vol. 313. — Pp. 504–7.

[11] *Jang E., Gu S., Poole B.* Categorical reparameterization with gumbel-softmax // *CoRR.* — 2017. — Vol. abs/1611.01144.

[12] *Kingma D. P., Welling M.* Auto-encoding variational bayes // *CoRR.* — 2014. — Vol. abs/1312.6114.

[13] *Koehn P.* Europarl: A parallel corpus for statistical machine translation.

[14] Latent alignment and variational attention / Y. Deng, Y. Kim, J. Chiu et al. // NeurIPS. — 2018.

[15] *Cho K., Gulcehre C., MontreMĺal U. . D. et al.* Learning phrase representations using rnn encoderвЂ“decoder for statistical machine translation.

[16] *Luong T., Pham H. Q., Manning C. D.* Effective approaches to attention-based neural machine translation // EMNLP. — 2015.

[17] On the properties of neural machine translation: Encoder-decoder approaches / K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio // Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), 2014. — 2014.

[18] *Sherstinsky A.* Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network // *CoRR.* — 2018. — Vol. abs/1808.03314.

[19] Toward controlled generation of text / Z. Hu, Z. Yang, X. Liang et al. // ICML. — 2017.

[20] Variational attention for sequence-to-sequence models / H. Bahuleyan, L. Mou, O. Vechtomova, P. Poupart // COLING. — 2018.

[21] *Zhou C., Neubig G.* Morphological inflection generation with multi-space variational encoder-decoders // CoNLL Shared Task. — 2017.