TEncDM: Понимание свойств диффузионной модели в пространстве кодировок языковых моделей

Лапиков Владислав

Аннотация

Представленная работа описывает Text Encoding Diffusion Model (TEncDM) — новаторский подход к моделированию текста с помощью диффузионной модели, работающей в пространстве кодировок языковой модели. В отличие от традиционных методов, основанных на эмбеддингах, TEncDM использует кодировки, которые содержат больше контекстной информации и улучшают качество предсказаний модели. Кроме того, в модели используется трансформерный декодер, специально разработанный для учета контекста при предсказании токенов, а также самоконтроль, что повышает точность генерации текста. Экспериментальные результаты на задачах перефразирования, суммаризации и упрощения текста подтверждают превосходство TEncDM над традиционными неавторегрессивными диффузионными моделями.

1 Введение

Авторегрессивные модели, такие как GPT-4 [21] и Llama 3 [5], демонстрируют высокое качество в задаче генерации текста. Они создают текст, проходя последовательно по каждому токену, что делает их подход надежным и естественным для создания длинных связных текстов. Однако эти модели имеют два значительных недостатка. Во-первых, они не могут корректировать ошибки, допущенные на ранних этапах генерации. Поскольку авторегрессивные модели генерируют текст слева направо, любая ошибка, допущенная на начальных шагах, будет "разрастаться искажая весь последующий текст. Во-вторых, авторегрессивный подход требует обработки каждого токена по отдельности, что замедляет процесс генерации, особенно на длинных последовательностях.

Диффузионные модели, активно развивающиеся в таких областях, как генерация изображений, аудио и видео, предлагают альтернативный метод. В диффузионных моделях процесс генерации происходит параллельно, что позволяет генерировать всю последовательность токенов одновременно. Это делает модель более гибкой и позволяет обрабатывать и корректировать любые части текста в процессе генерации, что ускоряет работу. Кроме того, диффузионные модели позволяют сократить количество необходимых вычислительных операций путем дистилляции, что также сокращает время генерации текста.

Существует множество подходов к адаптации диффузионных моделей для текстовой генерации, от замены гауссовского шума на категориальный до обучения на небольших латентных представлениях текста. Однако еще не найден оптимальный подход, который позволил бы эффективно моделировать текст с учетом особенностей его структуры и содержания. В данной работе исследуется возможность обучения диффузионной модели в латентном пространстве контекстуальных кодировок, предоставляемых языковыми моделями. Мы выделили и проанализировали влияние декодера, шумового расписания и метода самоконтроля на качество генерации текста, что позволило сформулировать основные рекомендации по разработке диффузионных моделей для текстовой генерации.

2 Постановка задачи и принципы диффузионного моделирования

Задача генерации текста заключается в построении текста, который удовлетворяет заданному условию, например, теме или стилю. Диффузионное моделирование предполагает преобразование случайного шума в структурированный текст. Для обучения модели требуется "прямой процесс диффузии в котором шум постепенно добавляется к латентному представлению текста, и "обратный процесс в ходе которого модель обучается восстанавливать исходный текст на основе зашумленного представления. При этом диффузионная модель должна учитывать контекст и структуру текста на всех этапах денойзинга, чтобы восстановить полноценную текстовую последовательность.

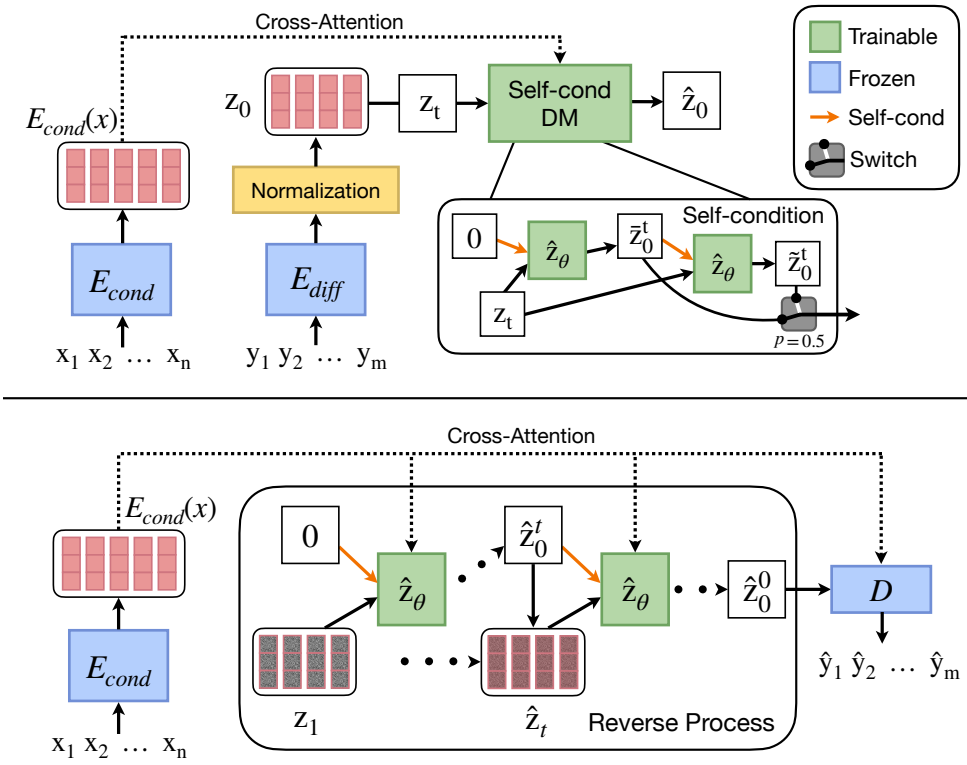

Рис. 1: Обзор фреймворка для условной генерации. Вверху - процесс обучения, внизу - процесс генерации.

3 Методология

Методология TEncDM включает несколько ключевых компонентов, которые позволяют повысить качество генерации текста за счет учета контекста и контроля шума на каждом этапе денойзинга.

3.1 Кодировщик

TEncDM использует предварительно обученную трансформерную модель для преобразования текста в латентное пространство кодировок, которое содержит контекстную информацию. В отличие от эмбеддингов, которые представляют токены фиксированными значениями, кодировки адаптируются к контексту, в котором находятся, что позволяет модели лучше понимать смысл текста. Кодировщик обрабатывает каждое пред-

ложение, преобразуя его в последовательность латентных переменных, выровненных по длине для корректной обработки в модели.

3.2 Диффузионная модель

Диффузионная модель TEncDM построена на базе трансформера с 12 слоями и обучается восстанавливать исходные латенты из зашумленных версий. Для этого используется шумовое расписание, в котором интенсивность шума контролируется параметром, что позволяет задать сложность восстановления на каждом шаге. Чем выше шум, тем сложнее диффузионной модели восстановить исходный текст, но это способствует более устойчивому обучению модели. Таким образом, контролируемое добавление шума помогает улучшить точность генерации текста.

3.3 Декодер

После этапа денойзинга декодер преобразует восстановленные латенты в текст. В TEncDM применяется трансформерный декодер, который учитывает контекст для каждого токена, что улучшает качество предсказаний. Декодер обучается независимо от диффузионной модели, чтобы его работа не зависела от ошибок, допущенных на предыдущих шагах. Таким образом, даже если диффузионная модель допускает неточность при восстановлении латентов, декодер способен скорректировать результат, улучшив итоговое качество текста.

3.4 Самоконтроль

Введение самоконтроля позволяет диффузионной модели повышать точность предсказаний на каждом шаге, ориентируясь на собственные предыдущие предсказания. В модели реализовано два сценария самоконтроля: с нулевым значением (для случаев, когда предыдущая информация отсутствует) и с учетом данных из предыдущего шага. Эксперимент показал, что самоконтроль позволяет модели сократить количество шагов денойзинга, что ускоряет процесс генерации.

4 Обзор литературы и открытых источников по теме генерации текстов с помощью диффузионных моделей

С момента появления диффузионных моделей и подтверждения их успешности для генерации изображений было предпринято немало попыток адаптации их для генерации текста. Все предоженные подходы можно разделить на две категории: дискретные диффузионные модели (Discrete Diffusion Models) и непрерывные диффузионные модели (Continuous Diffusion Models). Дискретные модели эксплуатируют дискретность текстовых данных и зашумляют их с помощью шума из категориального распределения. Непрерывные модели сперва переводят текст в непрерывный вид, например, с помощью замены токенов на их векторные представления, а после этого зашумляют данные стандартным гауссовским шумом.

На данный момент нельзя сказать однозначно, какое из этих направлений является более успешным. Обе области активно развиваются и каждый год пополняются новыми подходами. Далее мы рассмотрим подробнее методы каждой из этих категорий, чтобы сформировать представление о состоянии области.

В данном разеделе описан обзор и выводы в соответствии с анализом источников с 32 по 51, указанных в том числе в списке использованных источников.

4.1 Discrete Diffusion Models

Процесс зашумление – главное отличие между дискретными и непрерывными диффузионными моделями. В дискретных моделях каждый токен представляется в виде one-hot векторов (1 соответствует индексу токена, на всех остальных позициях стоят 0). Тогда, один шаг удаления информации записывается по формуле

$$q(x_t|x_{t-1}) = \mathrm{Cat}(x_t; p = x_{t-1}Q_t), \tag{1}$$

где x_t – набор one-hot векторов, соответствующих одному тексту на шаге зашумления t, а Q_t – стохастическая матрица перехода, переводящаяя каждый токен в вероятности перехода в другой токен. Для выбора матрицы Q_t существует несколько подходов. Два основные из них подробно разобраны в статье [1]:

1. **Равномерное распределение:** $Q_t = \alpha_{t|t-1}\boldsymbol{I} + \sigma_{t|t-1}/K$

2. **Поглощающее распределение:** $[Q_t]_{ij} = \begin{cases} 1 & \text{if} \quad i = j = m \\ \alpha_{t|t-1} & \text{if} \quad i = j \neq m \\ \sigma_{t|t-1} & \text{if} \quad j = m, i \neq m \end{cases}$,

где m – индекс токена маски [MASK]. Таким образом, каждый токен маскируется независимо с некоторой вероятностью.

Равномерное распределение применяется в работах [10, 16], а поглощающее – в [16, 9, 28, 22, 23].

Помимо способа удаления информации, методы дискретной диффузии различаются по предсказываемой сущности. В то время как [1, 9, 10, 22] предсказывают x_0, обучая модель с помощью кросс-энтропии, [19] показывает, что вместо этого можно предсказывать вектор из отношения плотностей всех токенов к плотности токена на текущем шаге, $s(x,t) = \left[\frac{p_t(y)}{p_t(x)}\right]_{y \neq x}$. В [16] эта идея развивается и предлагается новая функция ошибки на основе дивергенции Брегмана, улучшающая качество генерации текстов.

4.2 Continuous Diffusion Models

Непрерывные диффузионные модели повторяют идею диффузионных моделей, применяемых для изображений и аудио. В них объект зашумляется с помощью вливания в него гауссовского шума.

$$q(x_t|x_s) = \mathcal{N}(x_t; \alpha_{t|s}x_s, \sigma^2_{t|s}I) \tag{2}$$

Для текста в числовом формате чаще всего используются векторные представления токенов фиксированной длины [13, 7, 26, 14, 27, 25, 6]. Однако иногда модель обучают на симплексе [8, 18], представляя каждый токен в виде вектора $+k, -k^{|V|}$, где $+k$ стоит на месте индекса токена, а на всех остальных позициях стоит $-k$.

4.2.1 Расписание зашумления

Еще одной важной особенностью непрерывной диффузии является выбор расписания зашумления. В статьях [13, 6, 25] показано, что для текстовых диффузионных моделей требуется добавлять больше шума на ранних итерациях зашумления, чем для картиночных моделей. Данный феномен связан с дискретной сущностью текстов – векторные представления хорошо разделяют токены из-за проклятья размерности. Поэтому при добавлении небольшого количества шума каждый токен можно без труда восстановить и задача диффузионной модели становится слишком простой.

4.2.2 Self-conditioning

Также для непрерывных диффузионных моделей был предложен подход self-conditioning [2, 24]. Он модифицирует процесс генерации, добавляя в модель ее предсказание на предыдущем шаге в качестве условия $\hat{x}_0^{t-1} = p_\theta(x_{t-1}|x_t, \hat{x}_0^t)$. Несмотря на то, что такая модификация позволяет значительно улучшить качество генерации модели, почему именно это происходит – открытый вопрос. Для добавления self-conditioning процесс обучения тоже видоизменяется: с вероятностью 50% модель получает на вход пустое условие, $p_\theta(x_{t-1}|x_t, 0)$, и с вероятностью 50% свое предсказание для объекта x_t, $p_\theta(x_{t-1}|x_t, \hat{x}_0^t)$. Данная схема необходима для того, чтобы было возможно сделать первое предсказание на шаге T, когда никаких предсказаний модель еще не совершала.

4.2.3 Декодирование текста

В то время как самым распространенным способом декодирования текста является округление каждого сгенерированного вектора к токену с ближайшим векторным представлением [13, 7], были предложены и другие методы. В работах [27, 17] применяется авторегрессионная модель для генерации текста, которая получает на вход сгенерированные векторы в качестве условия.

5 Экспериментальный анализ

Для проверки эффективности TEncDM были проведены эксперименты на задачах условной генерации текста, включая перефразирование, суммаризацию и упрощение текста. Были использованы датасеты QQP [3] (вопросы-перефразирования), XSum [20] (экстремальная суммаризация) и Wiki-Auto [11] (упрощение текста). Модель TEncDM оценивалась на основе следующих метрик:

- **Perplexity** — показатель, который измеряет логарифмическое среднее расстояние между вероятностью предсказания модели и реальными данными. Низкий уровень Perplexity указывает на высокую точность генерации.
- **MAUVE** — метрика, оценивающая близость распределений сгенерированного и референтного текста. MAUVE измеряет, насколько тексты, сгенерированные моделью, похожи на реальные тексты, что позволяет оценить реалистичность и связность.

Encoder	**ppl** $\downarrow$	**mem** $\downarrow$	**div** $\uparrow$	**mauve** $\uparrow$
ROCStories				
BERT emb	$48.9_{.36}$	$.371_{.003}$	$.324_{.002}$	$.600_{.016}$
BERT	$29.1_{.89}$	$.453_{.003}$	$.295_{.002}$	$\mathbf{.762}_{.043}$
RoBERTa	$\mathbf{28.3}_{.33}$	$.443_{.003}$	$.302_{.002}$	$.647_{.019}$
T5	$31.3_{.54}$	$.427_{.003}$	$.312_{.004}$	$.706_{.024}$
BART	$34.1_{.52}$	$.441_{.006}$	$.299_{.005}$	$.705_{.030}$
Source text	21.7	.365	.403	.876
Wikipedia				
BERT emb	$156.1_{1.8}$	$.263_{.004}$	$.517_{.002}$	$.378_{.055}$
BERT	$\mathbf{104.4}_{2.1}$	$.286_{.002}$	$.504_{.003}$	$\mathbf{.874}_{.011}$
Source text	37.3	.122	.615	.957

Таблица 1: Сравнение энкодеров

Decoder	**ppl** $\downarrow$	**mem** $\downarrow$	**div** $\uparrow$	**mauve** $\uparrow$
ROCStories				
MLP	$39.7_{3.38}$	$.444_{.002}$	$.297_{.004}$	$.716_{.074}$
+ $Cor(z_0)$	$31.2_{.33}$	$.448_{.002}$	$.293_{.003}$	$.739_{.051}$
Transformer	$34.2_{.29}$	$.445_{.001}$	$.295_{.003}$	$.714_{.037}$
+ $Cor(z_0)$	$\mathbf{29.1}_{.89}$	$.453_{.003}$	$.295_{.002}$	$\mathbf{.762}_{.043}$
Wikipedia				
Transformer	$180.6_{3.2}$	$.261_{.001}$	$.511_{.001}$	$.526_{.025}$
+ $Cor(z_0)$	$\mathbf{104.4}_{2.1}$	$.286_{.002}$	$.504_{.003}$	$\mathbf{.874}_{.011}$

Таблица 2: Сравнение декодеров

- **Дивергенция** — метрика, которая измеряет разнообразие текстов, предотвращая дублирование и стандартизацию.
- **Меморизация** — метрика, которая измеряет, насколько модель запоминает данные из обучающего набора, что позволяет предотвратить чрезмерное повторение.

5.1 Результаты экспериментов

Результаты экспериментов показали, что модель TEncDM превосходит другие существующие подходы в генерации текста по всем основным метрикам. Например, на задаче перефразирования модель показала значительное преимущество перед конкурентами, демонстрируя более точные перефразы при сохранении смысла. На задаче суммаризации модель

смогла достичь высоких результатов, генерируя лаконичные и точные резюме. В задаче упрощения текста TEncDM также показала себя лучше других методов, успешно передавая основные мысли сложных текстов в простой форме.

Влияние отдельных компонентов на качество генерации

Кодировщик и декодер. Исследования показали, что использование кодировок вместо эмбеддингов позволяет модели учитывать больше контекста, а продвинутый декодер способен значительно улучшить качество текстов за счет адаптации к особенностям латентного пространства.

Самоконтроль. Анализ показал, что метод самоконтроля увеличивает точность предсказаний модели, особенно при генерации длинных текстов, когда важно учесть каждый шаг для сохранения связности.

Шумовое расписание. Контроль интенсивности шума на каждом этапе улучшил процесс обучения модели и позволил улучшить качество предсказаний, что особенно важно для обработки текстов с неоднородной структурой.

6 Разработка стратегии увеличения большой языковой диффузионной модели на задаче генерации текста порядка 100 млн. параметров до 1 миллиарда параметров

6.1 Описание задачи

Данный отчет описывает стратегию масштабирования диффузионной языковой модели генерации текста с порядка 100 млн. параметров до 1 миллиарда параметров. В ходе выполнения были рассмотрены различные подходы, проведены эксперименты с масштабированием различных частей модели (энкодера и диффузионной части), а также изучены особенности использования претрейна для предотвращения переобучения. В основе работы лежат результаты исследований и референсные данные по уже известным моделям, таким как BERT [4], RoBERTa [15] и GPT-Neo [12].

6.2 Цели разработки алгоритма

- Разработать стратегию увеличения числа параметров модели до 1 миллиарда, не ухудшая ее качество и устойчивость.

- Протестировать поведение моделей с различным количеством параметров при решении задач генерации текста.
- Определить ключевые параметры, которые необходимо оптимизировать для успешного масштабирования.

6.3 Методы и материалы

6.3.1 Исходные данные

Для разработки и масштабирования диффузионной модели были использованы следующие компоненты:

- Открытые модели вроде GPT-Neo.
- **Таблицы параметров**: Для сравнения были взяты размеры различных моделей, таких как BERT (108M и 334M параметров) и XLM-RoBERTa (278M, 560M и 43GB). Это помогло ориентироваться на размеры моделей и задавать параметры масштабирования диффузионной модели.

6.3.2 Параметры моделей

В рамках стратегии масштабирования был установлен ряд базовых параметров, таких как количество слоев (n_{layers}), размер модели (d_{model}), количество голов (n_{head}), размер головы (d_{head}) и другие. Это позволило формализовать размеры диффузионных моделей, которые использовались в экспериментах. Примерная структура моделей до и после масштабирования:

Таблица 3: Структура моделей до и после масштабирования

Diffusion	**Параметры модели**	n_{layers}	d_{model}	n_{head}	d_{head}	**Эндодер**
small	158M	12	768	12	64	bert-base
medium	352M	24	768	16	64	bert-base
large	1B	32	1024	24	64	bert-large

6.4 Эксперимент 1: Генерация текста без претрейна

6.4.1 Описание

Первый эксперимент был проведен с целью понять, как больший размер модели влияет на производительность без использования предварительного обучения (pretrain). Модели различного масштаба (small, medium)

обучались на задачах генерации текста с прямым использованием тестовых данных.

6.4.2 Результаты

Таблица 4: Результаты первого эксперимента

Метрика (QQP)	small	medium-352M	medium-366M
BLEU ↑	0.317	0.320	0.315
PPL ↓	61.3	60.9	64.0

Результаты показали незначительное улучшение качества при увеличении параметров модели до 352M, однако при дальнейшем увеличении до 366M наблюдалось переобучение. Большая модель показала относительно худшие результаты, несмотря на увеличение числа параметров.

Вывод: Для успешного масштабирования необходимо использовать претрейн, так как без него модели с большим количеством параметров склонны к переобучению и не демонстрируют значительного прироста качества.

6.5 Эксперимент 2: Масштабирование на WikiPedia

6.5.1 Описание

Данный эксперимент был направлен на увеличение модели до большего объема и масштабирование на задаче безусловной генерации текста (например, написание статей на основе данных из Википедии). В этом эксперименте мы проводили несколько циклов обучения с различными конфигурациями модели (small, medium, large).

6.5.2 Результаты

Также были проанализированы графики зависимости метрик от количества шагов обучения, что позволило определить оптимальный момент для остановки обучения во избежание переобучения.

Вывод: Увеличение количества параметров модели до уровня medium и large показало улучшение метрик качества (например, уменьшение PPL и увеличение Mauve).

Таблица 5: Результаты второго эксперимента

Diffusion	**Параметры**	PPL ↓	Mauve ↑	Div ↑	Mem ↓
small (129M)	base	77.06	0.941	0.556	0.273
medium (276M)	base	67.40	0.947	0.564	0.282
large (493M)	base	67.90	0.936	0.560	0.277

6.6 Эксперимент 3: Масштабирование на Fineweb

6.6.1 Описание

В этом эксперименте были предприняты попытки увеличить модель до 1B параметров с акцентом на увеличение диффузионной части модели, так как это показало лучшие результаты в предыдущем эксперименте. Были увеличены как энкодер, так и основная диффузионная часть модели. Пропорционально увеличивалось количество слоев и количество голов внимания (head).

Таблица 6: Параметры диффузионной модели для эксперимента 3

Diffusion	**Параметры модели**	n_{layers}	d_{model}	n_{head}	d_{head}
small	150M	12	768	12	64
XL	1B	48	768	24	128

6.6.2 Результаты

Таблица 7: Результаты увеличения диффузионной модели

Diffusion	PPL ↓	Mauve ↑	Div ↑
small	64.5	0.886	0.528
XL	46.5	0.912	0.338

Результаты показывают, что при увеличении модели до уровня XL качество текстов заметно улучшилось, однако снизилось разнообразие (Div).

6.6.3 Увеличение энкодера

Для того чтобы оставить количество параметров диффузионной части модели примерно на том же уровне, были добавлены два слоя проектора. Первый слой преобразует входной вектор размерности 1024 в вектор 768,

второй слой преобразует выходной вектор из размерности 768 в 1024. Остальная часть модели (трансформерные слои) осталась неизменной.

Таблица 8: Результаты увеличения энкодера

Encoder	PPL ↓	Mauve ↑	Div ↑
base	**64.5**	0.886	0.528
large	118.3	0.887	0.578

Результаты показывают, что перплексия (PPL) увеличилась вдвое, хотя остальные метрики (Mauve и Div) остались практически теми же. Это говорит о том, что тексты, генерируемые моделью, усложнились, но прироста качества не было.

Вывод: Применение претрейна на Википедии выявило ограниченность размера этого датасета, что сдерживает дальнейшее улучшение модели. Fineweb как более крупный датасет дает лучшие результаты при масштабировании, и эксперименты показывают, что увеличение диффузионной модели лучше сказывается на метриках, чем увеличение энкодера.

6.7 Перспективы и ограничения

Несмотря на высокие результаты, TEncDM имеет ряд ограничений. В частности, процесс генерации замедляется для длинных текстов, так как размерность латентного пространства возрастает. В будущем возможно создание автоэнкодера, который позволит сжимать латенты, улучшая производительность модели. Кроме того, разные задачи требуют настройки параметров модели, что предполагает необходимость в проведении дополнительных исследований для нахождения оптимальных значений для каждого типа задач.

7 Заключение

TEncDM демонстрирует высокие результаты в задачах генерации текста, превосходя классические диффузионные и авторегрессивные модели. Работа показала, что переход от эмбеддингов к контекстуальным кодировкам улучшает качество предсказаний и способствует созданию более натуральных текстов. Предложенная модель успешно справляется с различными задачами на реальных данных, показывая универсальность и точность. В дальнейшем планируется улучшить производительность

модели за счет уменьшения латентного пространства и оптимизации параметров для различных задач.

Список литературы

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc., 2021.

[2] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning, 2023.

[3] Zihang Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. Quora question pairs. 2017.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and et al. The llama 3 herd of models, 2024.

[6] Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Empowering diffusion models on the embedding space for text generation. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4664–4683, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[7] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control, 2023.

[9] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models, 2022.

[10] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions, 2021.

[11] Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. Neural crf model for sentence alignment in text simplification. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2020.

[12] Rohan Kashyap, Vivek Kashyap, and Narendra C. P. Gpt-neo for commonsense reasoning – a theoretical and practical lens, 2023.

[13] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation, 2022.

[14] Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[16] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution, 2024.

[17] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 56998–57025. Curran Associates, Inc., 2023.

[18] Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew E. Peters, and Arman Cohan. Tess: Text-to-text self-conditioned simplex diffusion, 2024.

[19] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data, 2023.

[20] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[21] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[22] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models, 2024.

[23] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data, 2024.

[24] Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. Self-conditioned embedding diffusion for text generation, 2022.

[25] Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. Dinoiser: Diffused conditional sequence learning by manipulating noises, 2024.

[26] Hongyi Yuan, Zhengyuan Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Seqdiffseq: Text diffusion with encoder-decoder transformers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[27] Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua M. Susskind, and Navdeep Jaitly. PLANNER: Generating diversified paragraph via latent language diffusion model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[28] Kun Zhou, Yifan Li, Wayne Xin Zhao, and Ji-Rong Wen. Diffusion-nat: Self-prompting discrete diffusion for non-autoregressive text generation, 2023.