Генерация знаний в NLP
Генерация текста на основе знаний

Подготовили:
Тодосиев Н. Д.
Янковский В. И.

Проблема извлечения знаний из текста

- Философский вопрос о самом представлении знаний
- Какие части информации используются для генерации знаний (морфологическая, синтаксическая, семантическая)
- Хранение и обработка баз знаний
- Валидация результатов построения (как убедиться, что знания полноценно представляют исходные тексты)

История вопроса генерации знаний

1. Учения Хомского о семантическом представлении текста (“Синтаксические структуры”)
2. Введение первых стандартов генерации текста (RDF)
3. Появление понятия Text Mining
4. Векторные эмбеддинги, трансформеры

Задача обработки текстов

- Трансформеры (BERT, GPT, T5 [1])
- Библиотека инструментов DeepPavlov [2]
- Многое другое [3]

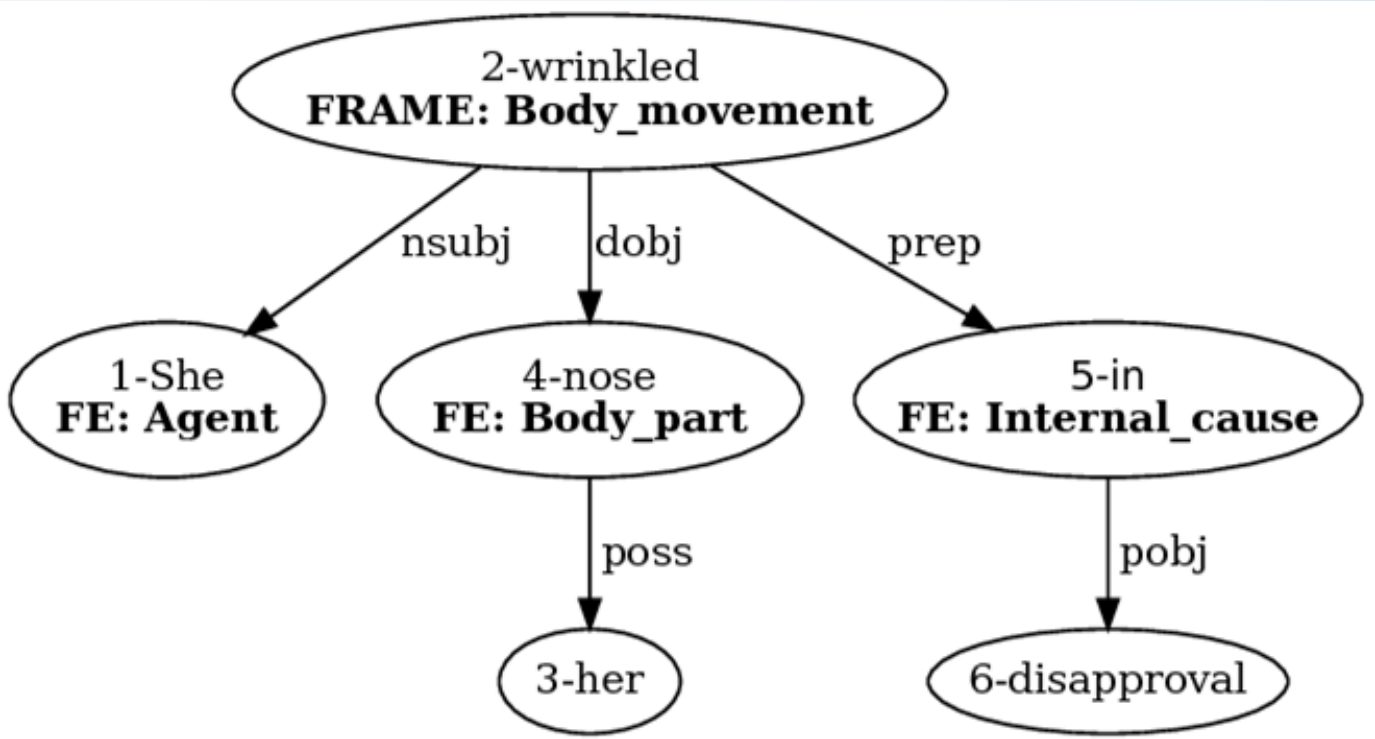
[1] https://github.com/google-research/text-to-text-transfer-transformer
[2] https://deeppavlov.ai/
[3] https://nlpub.ru/Обработка_текста

Фреймы как способ представления знаний 1

Фреймовая модель – это модель представления знаний, в основе которой лежат фреймы.

Фрейм состоит из конечного числа слотов (или составных ячеек), каждый из которых имеет имя и значение.
Последнее может быть ссылкой на другие слоты или фреймы, что и является ключевой особенностью данного формата концептуального представления.

Фреймы как способ представления знаний 2



https://www.researchgate.net/figure/Example-FrameNet-annotation-FEs-and-FEEs-are-annotated-over-text-spans-Additional_fig1_268290484

Фреймы: современные решения

- FrameNet (1997) - фреймы представляют “сценарии” (e.g. “Body movement”, “Adjusting”, “Contacting”) [1,2]
- FrameBank - русская версия FrameNet [3]
- Фреймы Жаботинской - фреймы представляют конкректные семантические отношения (“Сущность”, “Действие”, “Владение”, “Определение”, “Сравнение”) [4]

[1]https://dl.acm.org/doi/10.3115/980845.980860
[2]https://framenet.icsi.berkeley.edu/fndrupal/
[3]https://www.ruslang.ru/doc/kashkin/2015/06.pdf
[4]Zhabotynska, S. A. 2010. “Principles of Building Conceptual Models for Thesaurus Dictionaries.” Cognition, Communication, Discourse 1: 75–92.

Фреймы: достоинства и недостатки

Достоинства

- Имеют четкую структуру
- Существует множество семантик с разным ядром построения фреймов

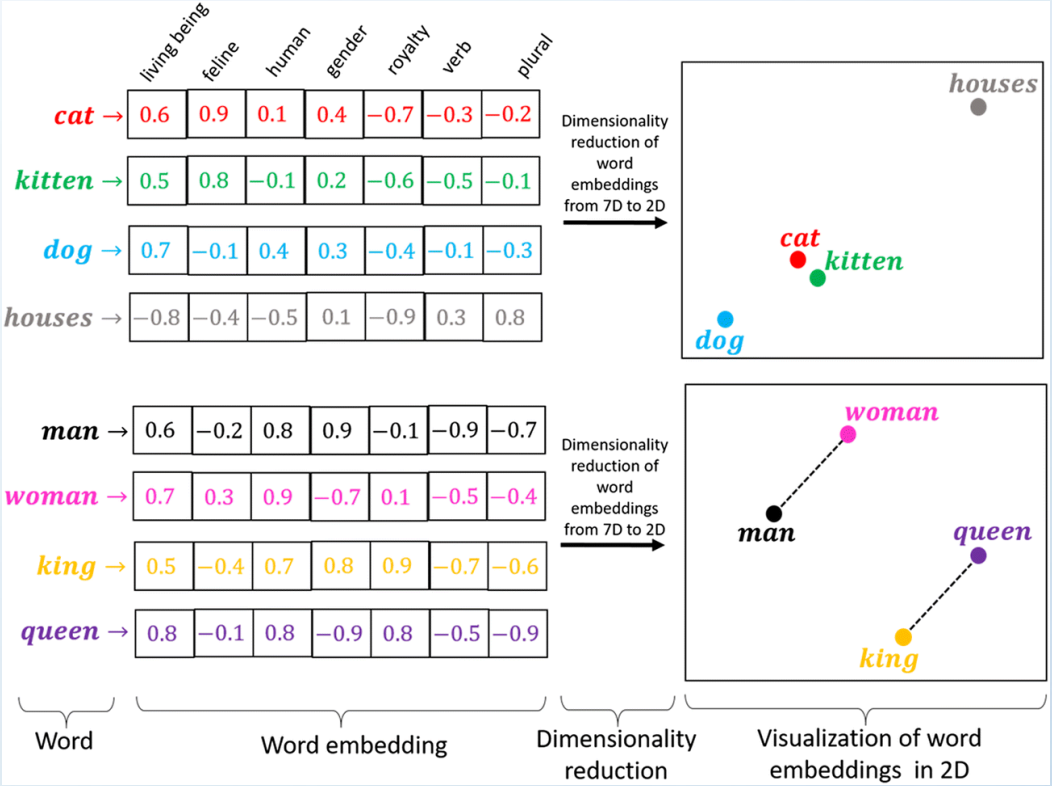
Недостатки

- Четкая структура ограничивает область знаний
- Необходимость прорабатывать фреймы предварительно, т.е. задавать точки опора

Эмбеддинг как способ представления знаний 1

- Присутствует в трансформерах state-of-the-art в задаче генераций
- Векторное представление знаний - слова представлены в формате массива чисел, текст - в формате матрицы чисел
- Поначалу использовался как end-to-end подход для text-to-text задач

Эмбеддинг как способ представления знаний 2



https://medium.com/@hari4om/word-embedding-d816f643140

Эмбеддинг: современные решения

- BERT
- GPT1, GPT2, GPT3 [1]
- T5 [2]

[1]https://arxiv.org/abs/2005.14165
[2]https://github.com/google-research/text-to-text-transfer-transformer

Эмбеддинг: достоинства и недостатки

Достоинства

- Достаточно легко обучить
- “end-to-end”: на вход текст, на выходе текст
- Возможность выполнять логические вычисления на основе

Недостатки

- Черный ящик беспросветной тьмы
- Большое количество ограничений к качеству выходящего текста [1]
- Качество логических вычислений оставляет желать лучшего [2]

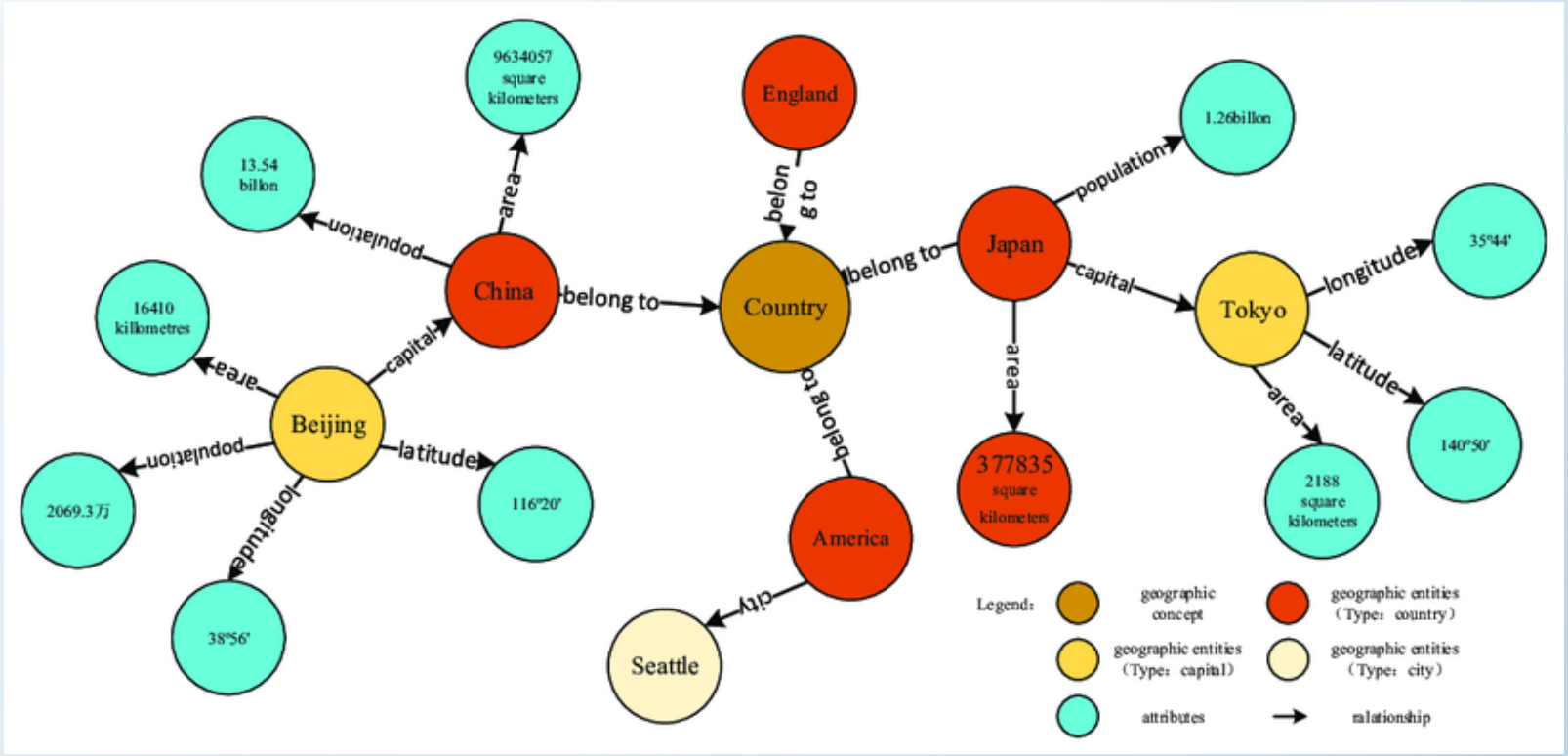
[1]https://www.cambridge.org/core/journals/natural-language-engineering/article/gpt3-whats-it-good-for/0E05CFE68A7AC8BF794C8ECBE28AA990
[2]https://blog.esciencecenter.nl/king-man-woman-king-9a7fd2935a85

Графы знаний как способ представления знаний 1

Графы знаний - семантическая технология и база знаний, используемая Google для повышения качества своей поисковой системы с семантическо-поисковой информацией, собранной из различных источников.

Отличается от фреймовой модели отсутствием отдельной базы знаний о структурах фреймов и большей гибкостью построения

Графы знаний как способ представления знаний 2



https://www.researchgate.net/figure/Example-of-a-geographic-knowledge-graph_fig1_336038833

Графы знаний: достоинства и недостатки

Зачем работать с графами:

- Natural-language understanding
- Уход от концепта “черный ящик”
- Гибкость представленных данных

Недостатки:

- Сложность подготовки тестовой выборки для обучения
- Сложность построения алгоритмов по генерации и получению графов знаний

Хранилище графов знаний

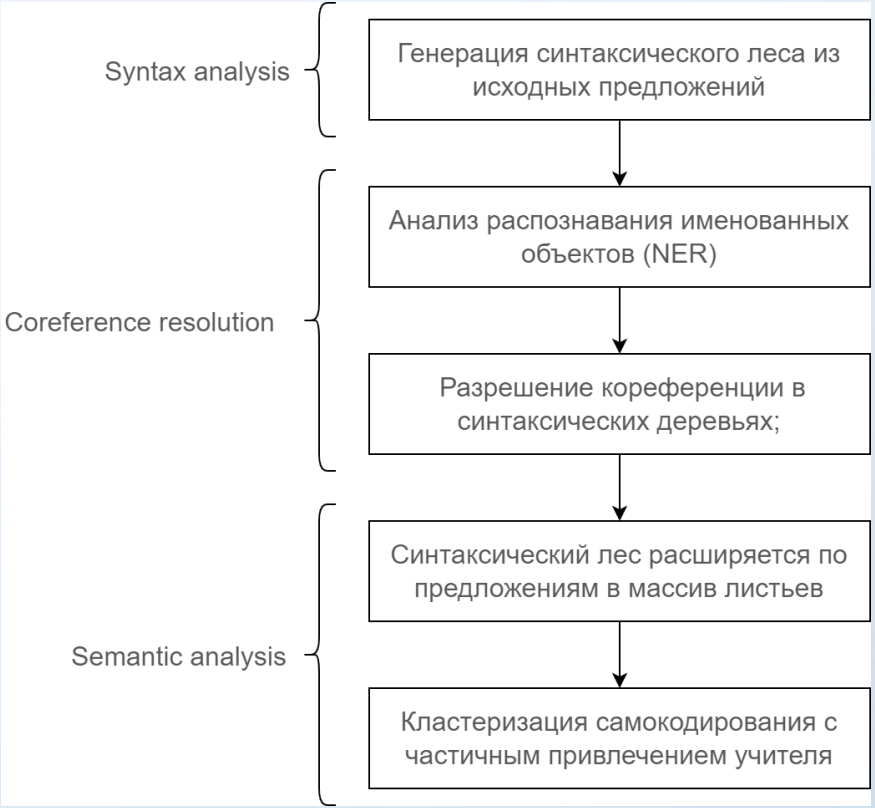
Популярные способы хранения графов знаний

- Neo4j - каноничное хранение знаний в виде графов;
- Mongodb - документоориентированное хранение;
- SQL - реализация через файлы с содержанием RDF-триплетов

Построение графов знаний

1. Синтаксический анализ
2. Морфологический анализ
3. Токенизация, Стемминг
4. NER, Coreference
5. А дальше как-то лепим граф :)

Наработки на кафедре по генерации графов знаний



Проблема поставленной задачи генерации текста

- Морфологические и синтаксические особенности каждого отдельного естественного языка
- Искоренение предвзятости к чувствительным темам в сгенерированных текстах [1]
- Малое количество датасетов для обучения моделей генерации текстов (к примеру, датасет по соревнованию WebNLG [2])
- Валидация результатов построения

[1]https://www.infoworld.com/article/3610403/battling-bias-and-other-toxicities-in-natural-language-generation.html
[2]https://github.com/WebNLG/GenerationEval

Оценка BLEU (bilingual evaluation understudy)

- Изначально использовался для задачи машинного перевода
- Считается с помощью соответствия сгенерированного текста “золотому стандарту”
- BLEU является одной из самых популярных оценок качества генерируемого текста

Рейтер в 2018 по поводу оценки BLEU:

“... результаты не подтверждают использование BLEU для оценки других типов систем NLP (помимо машинного перевода) и не поддерживают использование BLEU для оценки отдельных текстов, а не систем NLP…”

[1]https://direct.mit.edu/coli/article/44/3/393/1598/A-Structured-Review-of-the-Validity-of-BLEU

История вопроса генерации текстов на основе знаний

1. Середина XX века - машинный перевод как первая задача работы с текстовым представлением работы над знаниями;
2. 70-80 годы XX века - классический ИИ
3. Начало XXI века - “pipeline approach”
4. Настоящее время - “end-to-end approach”

Генерация текста на основе графов знаний

Прародителем современной основополагающей схемы систем генерации текстов можно назвать работу Рейтера и Дэля в 2000 году [1]. Основные этапы:

a. Определение содержания
b. Планирование дискурса
c. Агрегация предложений
d. Лексикализация
e. Генерация ссылочного выражения
f. Лингвистическая реализация

прим. Графы знаний могут рассматриваться или как источник данных для генерации текста, или как средство семантического обогащения генерируемого текста.

[1]https://www.researchgate.net/publication/2839784_Building_Applied_Natural_Language_Generation_Systems

Системы на основе правил, шаблонов, онтологий

- Основаны на теории риторических структур (RST - Rhetorical structure theory), которое описывает отношения между двумя частями текста [1]
- Примерами реализаций являются ILEX [2]
- В настоящее время применяется больше в задачах анализа текстов, нежели чем в генерации

[1]https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.2668&rep=rep1&type=pdf
[2]https://www.researchgate.net/publication/231787808_ILEX_An_architecture_for_a_dynamic_hypertext_generation_system

Системы на основе эволюционного подхода

- Надстройка над системой на основе правил, шаблонов и онтологий
- Над правилами построения текстов из графов дополнительно пробрасывается генетический алгоритм
- Фитнесс-функции считаются на основе сгенерированного текста. Могут использоваться и правила естественных языков
- Не получила широкой популярности и была забыта ввиду роста популярности подходов “pipeline approach” “end-to-end approach”

[1]https://aclanthology.org/W02-2112/

Системы “pipeline approach”

- Планировщик становится моделью машинного обучения -> уход от жестко структурированных правил
- Появление нейронных сетей для подзадач генерации текстов
- Примеры:
  - Neural Data-to-Text Generation with LM-based Text Augmentation [1]
  - Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation [2]
  - Data-to-Text Generation with Iterative Text Editing [3]
- В среднем хуже систем “end-to-end”, но тем не менее находят своё применение [4]

[1]https://arxiv.org/abs/2102.03556
[2]https://aclanthology.org/N19-1236/
[3]https://arxiv.org/abs/2011.01694
[4]https://aclanthology.org/D19-1052/

Системы “end-to-end approach”

- Использование нейронных сетей для преобразования Data-to-Text
- Ярким представителем будет T5 (text-to-text transfer transformer) [1], но и используют также и другие виды нейронных сетей (RNN [2], LSTM [3])
- Датасетом является связка “входящие вершины - выходящий текст”
- Доминирующее решение в настоящее время, однако имеет ту же проблему “черного ящика”

[1]https://github.com/google-research/text-to-text-transfer-transformer
[2]https://arxiv.org/abs/1803.07133
[3]https://aclanthology.org/P19-1195/

Системы на основе обучения с подкреплением

- Развитие эволюционного подхода
- Модель “генератор” (агент) - создает текст
  Модель “критик” (среда) - проверяет текст на качество
- Обучение с учителем (Supervised learning) без подготовки данных
- Проблема оценки качества текста приводит к сложности определения модели “критика”
- Примеры:
  - RDF-to-Text Generation with Reinforcement Learning Based Graph-augmented Structural Neural Encoders [1]
  - Generation with Inverse Reinforcement Learning [2]
  - How Helpful is Inverse Reinforcement Learning for Table-to-Text Generation? [3]

[1]https://arxiv.org/abs/2111.10545
[2]https://www.researchgate.net/publication/324859832_Towards_Diverse_Text_Generation_with_Inverse_Reinforcement_Learning
[3]https://aclanthology.org/2021.acl-short.11/

Системы макропланирования и микропланирования

- Планирование == “Распределение последовательности слов в части текста”
- Макропланирование - на уровне текста,
  микропланирование - на уровне предложения;
- Оба аспекта в той или иной степени отражаются во всех системах генерации текстов, однако работы [1] и [2] акцентируют внимание полноценно только на одной из составляющих.

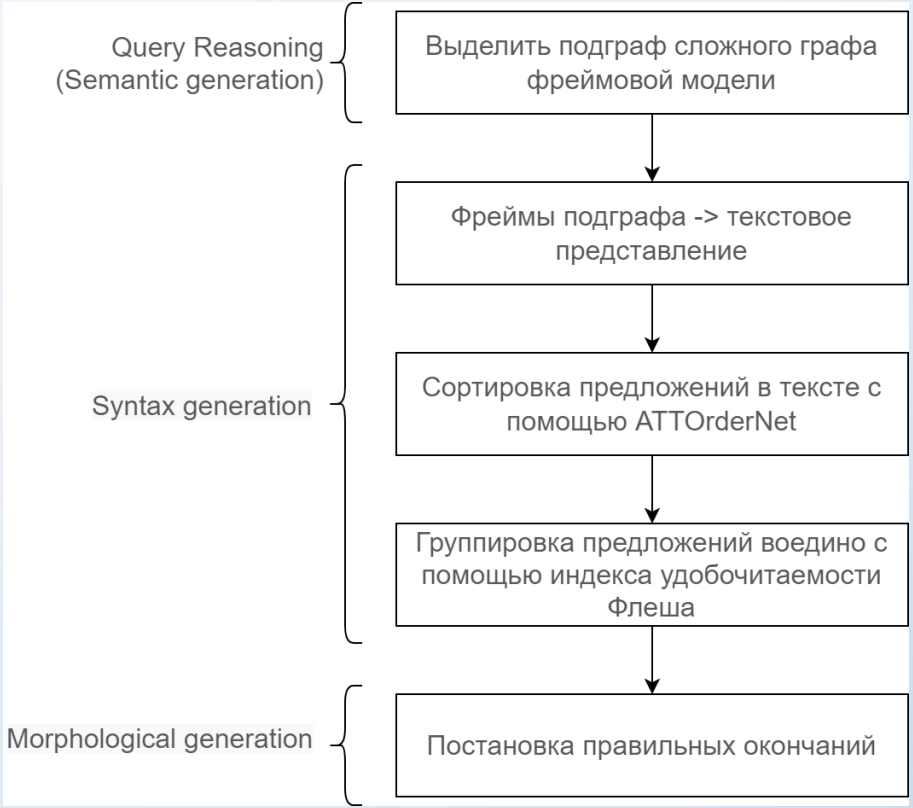
[1]https://arxiv.org/pdf/2102.02723
[2]https://direct.mit.edu/coli/article/45/4/737/93360/Scalable-Micro-planned-Generation-of-Discourse

Дополнительные задачи по генерации текста

- Генерация текстов на основе коротких текстов (расширение текстов) [1]
- Перенос (имитация) стиля при генерации текста [2, 3]
- Заслуживают отдельного внимания, но не являются задачами генерации текстов как таковые

[1]https://arxiv.org/abs/2012.04332
[2]https://arxiv.org/abs/2010.12742v2
[3]https://arxiv.org/abs/1901.09501

Наработки на кафедре по генерации текстов из графов



Дополнительные источники

Обзорные статьи по генерации текстов на основе нейронных сетей:
https://arxiv.org/abs/1803.07133
https://www.sciencedirect.com/science/article/pii/S1319157820303360