Автоматическая суммаризация текста

Математические методы анализа текстов
осень 2021

Попов Артём Сергеевич

1 декабря 2021 г.

Задача суммаризации текста (text summarization)

Суммаризация — построение по одному или нескольким документам краткого текста (summary) наиболее полно передающего их содержание.

Основные типы задач суммаризации:

- one-document — на входе один документ $d \in D$
- multi-document — на входе набор документов $D' \subseteq D$
- topic — на входе информация по теме t тематической модели

Два основных подхода к суммаризации:

- extractive — выборка из исходных предложений
- abstractive — генерация текста

Зачем нужна суммаризация?

Области применения суммаризации:

- выделение полезной информации из научных статей
- составление дайджестов научных статей
- генерация заголовков/сниппетов новостей

Какая может быть польза от суммаризации?

- не поглощаем избыточную информацию $\Rightarrow$ получаем больше свободного времени
- чтобы сжать текст без потери информации, необходимо полностью осознать его смысл $\Rightarrow$ суммаризация очень сложная и показательная для всей области задача

Метрика ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

Доля n-грамм из абстрактов, вошедших в суммаризацию s:

$$\text{ROUGE-}n(s) = \frac{\sum_{r \in R} \sum_w [w \in s][w \in r]}{\sum_{r \in R} \sum_w [w \in r]}$$

Максимальная доля n-грамм по возможным абстрактам, вошедших в s:

$$\text{ROUGE-}n_{\text{multi}}(s) = \max_{r \in R} \frac{\sum_w [w \in s][w \in r]}{\sum_w [w \in r]}$$

- $r \in R$ — множество абстрактов, написанных людьми
- s — абстракт, построенный системой
- чем больше, тем лучше — для всех метрик семейства ROUGE

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. 2004.

Оценивание суммаризации с точки зрения теории информации

Суммаризация — задача семантического сжатия информации.

Концепции из теории информации

- Энтропия — мера неопределённости распределения:

$$H(p) = -\sum_i p_i \log p_i$$

- Кросс-энтропия — мера неопределённости, возникающая при наблюдении p при ожидании q:

$$CE(p, q) = -\sum_i p_i \log(q_i)$$

- KL-дивергенция — потери информации при ипользовании p вместо q:

$$KL(p, q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right) = CE(p, q) - H(p)$$

Оценивание суммаризации с точки зрения теории информации

Пусть текст d, абстракт s, фоновая информация b определяются распределением над семантическими элементами:

- распределение токенов в документе $n_{wd} \,/\, \sum_{w \in d} n_{wd}$
- тематические представления $p(t|d)$

Качество абстракта можно оценить по формуле:

$$Q(s, d, b) = H(s) - \alpha CE(s, d) + \beta CE(s, b)$$

- $H(s)$ должно быть большим, так как абстракт должен содержать большое количество информации
- $CE(s, b)$ должно быть большим, абстракт не должен содержать общей информации

Peyrard (ACL 2019); A Simple Theoretical Model of Importance for Summarization

Пример суммаризации

Abstract

We present a method to produce abstractive summaries of long documents that exceed several thousand words via neural abstractive summarization. We perform a simple extractive step before generating a summary, which is then used to condition the transformer language model on relevant information before being tasked with generating a summary. We show that this extractive step significantly improves summarization results. We also show that this approach produces more abstractive summaries compared to prior work that employs a copy mechanism while still achieving higher rouge scores. *Note: The abstract above was not written by the authors, it was generated by one of the models presented in this paper.*

S.Subramanian et al; On Extractive and Abstractive Neural Document Summarization with Transformer Language Models. 2019.

Основные этапы extractive суммаризации

1. Построение представления текста
   - эмбеддинги предложений
   - граф / кластеризация

2. Ранжирование предложений

3. Отбор предложений для абстракта
   - оптимизация критериев информативности и различности
   - оптимизация последовательности предложений
   - учёт целей и особенностей прикладной задачи (новости/статьи/веб-страницы/посты/мэйлы)

TextRank — аналог ссылочного ранжирования PageRank

Текст — граф предложений. Рёбрами соединены «похожие» предложения.

Предложение $s \in S$ тем важнее:
— чем больше других предложений c, похожих на s
— чем важнее предложения c, похожие на s
— чем меньше других предложений, на которые c также похоже

Вероятность попасть в s, случайно блуждая по похожим:

$$TR(s) = (1-\delta)\frac{1}{|S|} + \delta \sum_{c \in S_s} \frac{1}{|S_c|} TR(c)$$

$S_s \subset S$ — множество предложений похожих на s
$\delta \in (0,1)$ — вероятность продолжать блуждания (damping factor)

Rada Mihalcea, Paul Tarau. TextRank: Bringing Order into Text. EMNLP-2004.

TextRank с использованием близости

Пусть граф предложений — полный, но каждое ребро между предложениями s и c имеет вес $sim(s, c)$. Будем интерпретировать нормированный вес ребра как вероятность перехода из одной вершины в другую:

$$TR(s) = (1 - \delta)\frac{1}{|S|} + \delta \sum_{c \in S} \frac{sim(s, c)}{Z(c)} TR(c)$$

$$Z(c) = \sum_{s ß S} sim(s, c)$$

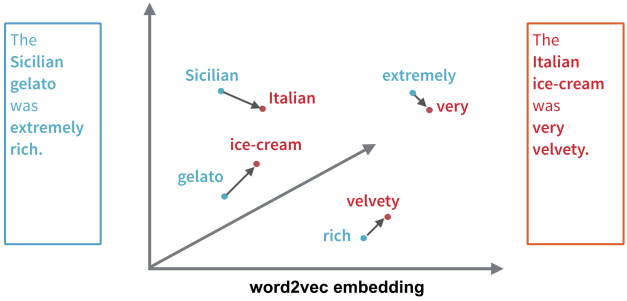
В качестве близости можно использовать:

- Косинус/скалярное произведение между эмбеддингами предложений
- Выход [CLS] токена по двум поданным предложениям
- Близости основанные на Word Mover's Distance / BERT score

Word Mover's Distance (WMD)

Проблема: если эмбеддинг предложения строится как взвешенная сумма его слов, его смысл размывается.

Идея: введём расстояние между предложениями как расстояние между двумя матрицами эмбеддингов.



Word Mover's Distance (WMD)

Данные два предложения,заданные BOW вектором: $\{n_w\}_{w \in W}$ и $\{q_u\}_{u \in W}$.

WMD между ними задаётся как решения оптимизационной задачи:

$$\sum_{w \in W} \sum_{u \in W} T_{wu} distance(u, w) \to \min_{T_{wu} \geqslant 0}$$
$$\sum_{w \in W} T_{wu} = q_u \quad \forall u \in W$$
$$\sum_{u \in W} T_{wu} = n_w \quad \forall w \in W$$

- перед подсчётом из предложений исключают стоп-слова
- при большом количестве подсчётов разумно кэшировать $distance(u, w)$

Нейросетевые модели extractive суммаризации

Extractive суммаризация — задача разметки предложений с классами «включать в абстракт» и «не включать».

Можем использовать нейросетевые модели для работы с последовательностями предложений:

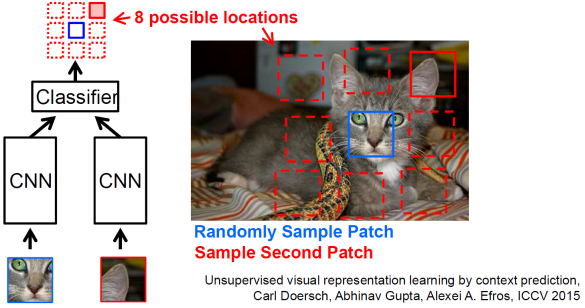
- рекуррентные нейронные сети
- трансформеры
- свёрточные сети

На вход модели подаётся последовательность эмбеддингов предложений.

Проблема: эмбеддинги предложений составляются без учёта контекста (других предложений документа).

Концепция самообучения (self-supervision)

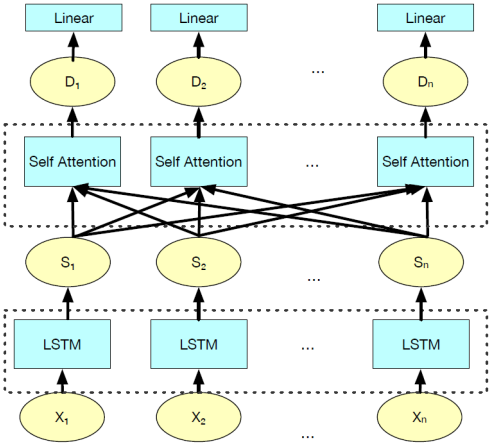
В компьютерном зрении сеть учится предсказывать взаимное расположение двух фрагментов на одном изображении.



Unsupervised visual representation learning by context prediction, Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

Преимущество: не нужна размеченная обучающая выборка, при этом сеть способна выучить векторные представления.

Self-supervision для предобучения сети



- Каждое предложение задаётся последовательностью слов
- LSTM применяется к словам предложения, задаёт эмбеддинг предложения
- Несколько self attention слоёв отвечают за учёт контекста
- На выходе сети решаем одну из синтетических задач
- После предобучения архитектура используется для решения первоначальной задачи

Wang et al. Self-Supervised Learning for Contextualized Extractive Summarization. 2019.

Self-supervision для предобучения сети

- **Mask**
  — с вероятностью $P_m = 0.25$ пропускать предложение
  — предсказывать предложение из пула пропущенных T_m
- **Replace**
  — с вероятностью $P_r = 0.25$ заменять предложение случайным предложением из *другого документа*
  — предсказывать, было ли предложение заменено
- **Switch**
  — с вероятностью $P_s = 0.25$ заменять предложение случайным предложением из *данного документа*
  — предсказывать, было ли предложение заменено

Резюме по extractive суммаризации

- Можно делать как supervised, так и unsupervised
- На текущий момент используется в коммерческих проектах чаще чем abstractive, особенно для длинных текстов
- Одного алгоритма ранжирования недостаточно, нужно также представить информацию в удобном для пользователя виде:
  - разбить предложения по категориям (например, results, methods, background)
  - расположить предложения в удобном для просмотра порядке (например, согласно их появлению в тексте)

Подход abstractive суммаризации

Хотим решать задачу при помощи архитектуры encoder-decoder:

- На входе исходный текст
- На выходе абстракт

Мы можем использовать всё, что мы проходили до этого для задач машинного перевода и генерации текста:

- архитектура encoder-decoder с LSTM и механизмом внимания
- архитектура encoder-decoder с трансформером
- языковое моделирование
- BPE токенизация
- гиперпараметры генерации предложений: beam-search, температура, topK, topP

Какие возникают проблемы при abstractive суммаризации?

- Нужна большая обучающая выборка текст-абстракт
- Модель будет специфична для конкретного домена
- Исходный текст может иметь очень большую длину
- В задаче суммаризации очень важно точно воспроизвести в абстракте «конкретную информацию»: именнованные сущности, даты и т.д.

Идея: хотим совместить abstractive и extractive подходы. Модель должна генерировать текст, но при этом некоторые куски она может целиком копировать из исходного текста.

Pointer-Generator Networks

- Пусть вероятность следующего токена в абстракте для модели encoder-decoder задаётся $p_{model}(w_i|d, a_{<i})$
- Хотим подмешивать в вероятности следующего токена веса внимания:

$$p_{attn}(w_i|d, a_{<i}) = \sum_{u \in d, u = w_i} \alpha_{ui}$$

$$p(w_i|d, a_{<i}) = p_{gen} p_{model}(w_i|d, a_{<i}) + (1 - p_{gen}) p_{attn}(w_i|d, a_{<i})$$

α_{ui} — вес внимания от u-го токена входа к i-ому токену выхода

- Коэффициент взвешивания $p_{gen} \in [0, 1]$ можно положить константным, но эффективнее вычислять его значения в единой end-to-end архитектуре аналогично механизму внимания

See et al; Get To The Point: Summarization with Pointer-Generator Networks;

Регуляризация механизма внимания

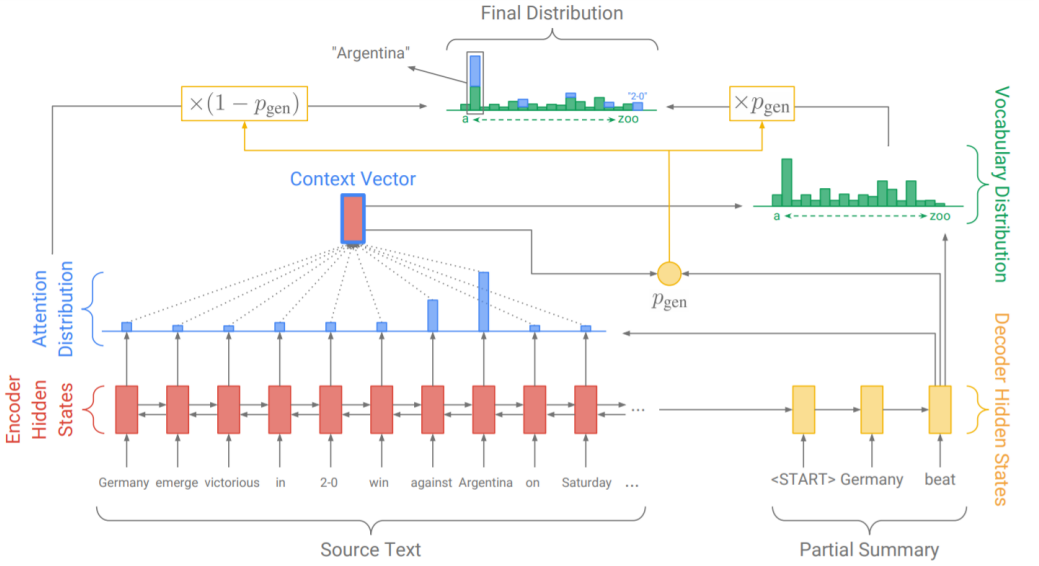
Проблема модели: так как модель явно подмешивает выходы механизма внимания в распределение выхода, крайне нежелательно, чтобы внимание концентрировалось вокруг конкретных токенов.

Можно изменить механизм внимания, чтобы явно учитывать прошлые шаги:

$$c_{ui} = \sum_{j=0}^{i-1} \alpha_{uj}$$

$$\alpha_{ui} = \mathrm{softmax}(v^t \tanh(W[h_u, h_i] + \langle \mathrm{w}, \mathrm{c_u} \rangle + b)), \quad W, v, w, b \text{ — параметры}$$

Pointer-Generator Networks



Bottom-Up Attention

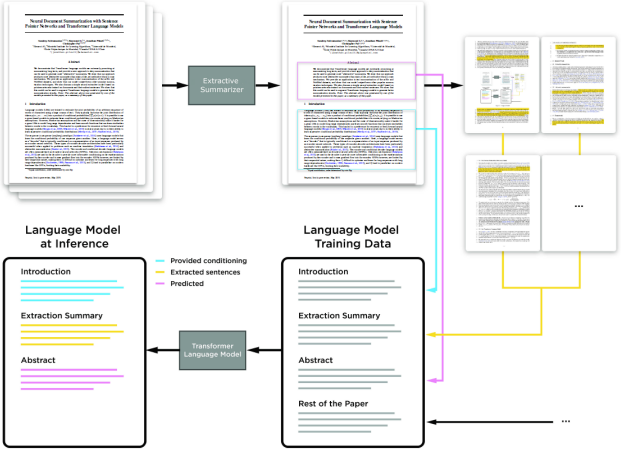
Проблема: pointer-generator сети могут копировать слишком много информации. Одно из решений это проблемы — явный запрет копирования некоторых токенов.

1. Решить задачу разметки на отдельных токенах (входит/не входит в абстракт).
2. В распределение p_{attn}, отвечающем за копирование информации, оставляем только токены, для которых предсказано вхождение.

Gehrmann et al (2018); Bottom-Up Abstractive Summarization

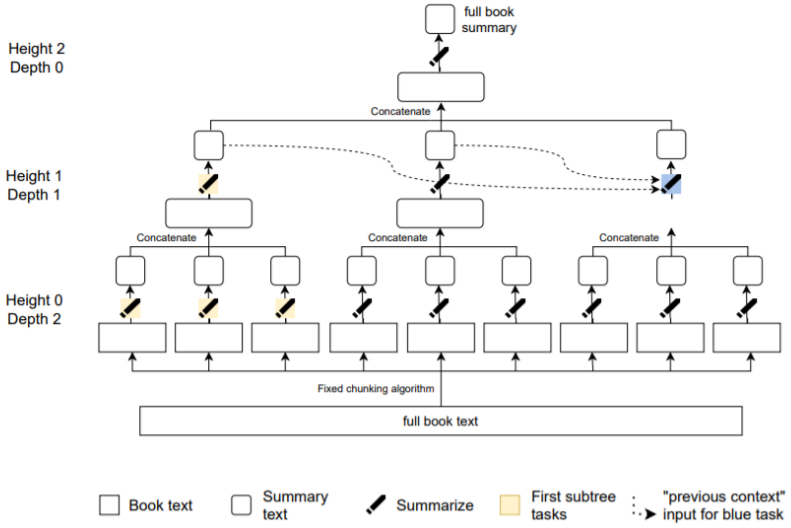
Суммаризация как задача языкового моделирования

Вместо encoder-decoder архитетуры можно использовать языковую модель.



Subramanian et al (2019); On Extractive and Abstractive Neural Document Summarization with Transformer Language Models

Суммаризация больших фрагментов текстов



Wu et al (2021); Recursively Summarizing Books with Human Feedback

Резюме по abstractive суммаризации

- Всё ещё открытая задача, не так много успешных коммерческих проектов.
- Одна из самых сложных задач NLP.
- Часто хорошо себя показывают методы, которые пытаются комбинировать extractive и abstractive подходы.

Материалы по суммаризации

- *D.Das, A.Martins.* A survey on automatic text summarization. 2007.
- *A.Nenkova, K.McKeown.* A survey of text summarization techniques. 2012.
- *Yogita Desai, Prakash Rokade.* Multi Document Summarization: Approaches and Future Scope. 2015.
- *Mahak Gambhir, Vishal Gupta.* Recent automatic text summarization techniques: a survey. 2016.