Математические методы анализа текстов

Лекция

Языковые модели и их приложения

Мурат Апишев (mel-lain@yandex.ru)

8 октября, 2019

Языковая модель

- Хотим присвоить вероятности последовательностям слов
- *Языковой моделью* называется модель, умеющая вычислять хоть одну из этих вероятностей:
  1. $P(W) = P(w_1, \ldots, w_n)$
  2. $P(w_n \mid w_1, \ldots, w_{n-1})$
- Цепное правило:

$$P(X_1, \ldots, X_n) = P(X_1)P(X_2 \mid X_1) \ldots P(X_n \mid X_1, \ldots, X_{n-1})$$

- Формула условной вероятности:

$$P(X_n \mid X_1, \ldots, X_{n-1}) = \frac{P(X_1, \ldots, X_n)}{P(X_1, \ldots, X_{n-1})}$$

Языковая модель

- Для оценивания $P(X_n \mid X_1, \ldots, X_{n-1})$ нужно посчитать $P(X_1, \ldots, X_n)$ и $P(X_1, \ldots, X_{n-1})$
- С ростом n число всевозможных последовательностей растёт экспоненциально $\Rightarrow$

  Проблема: для большого n эти вероятности близки к нулю
- Для упрощения применим *марковское предположение*:

$$P(X_n \mid X_1, \ldots, X_{n-1}) \approx P(X_n \mid X_{n-k+1}, \ldots, X_{n-1}), \quad k \ll n$$

- Окончательная формула ($w_{i-k}^{i} := w_{i-k}, \ldots, w_i$)

$$P(w_1, \ldots, w_n) = \prod_i P(w_i \mid w_{i-k+1}^{i-1})$$

Приложения языковых моделей

- Генерация текста
- Распознавание речи/текста
- Машинный перевод
- Исправление опечаток
- Определения языка
- Определение части речи (POS)
- ...

Задача генерации текста

- В узкой тематике бот может обмануть обычного человека, но не специалиста
- В широкой и простой тематике выявить качественного бота можно только по шаблонам в предложениях и явному комбинированию слов:

  Все ваши посты – типичное клише лживой инсинуации, которая стремится дискредитировать и осмеять всякого, кто начинает прозревать и открыто говорить о преступлениях преступного режима. Колет глаза держимордам кровавого кремлёвского упыря правда об их бесчеловечии и о фашистской сути кровавого кремлёвского режима! Интересной особенностью данного форума является то, что путинисты в основном занимаются флудом или обсуждением личностей, а топиков по существу проблем России, вроде этого, боятся как черт ладана.

- Разумеется, качество сильно падает в ситуации условной генерации, например, при ответе на конкретный вопрос
- Генерация текста без жёсткого шаблона может производится с помощью языковых моделей
- Вопрос в том, как их параметризовать и как обучать

Простейший подход

1. Параметры представляют собой в явном виде хранящиеся вероятности
2. Собираем вероятности статистически по корпусу:

$$\hat{P}_S(w_N \mid w_1^{N-1}) = \frac{c(w_1^N)}{c(w_1^{N-1})},$$

   $c(w_1^N)$ – число последовательностей $w_1, \ldots, w_N$ в корпусе
3. Сэмплируем из полученных эмпирических распределений
4. **Проблема:** для многих вероятностей даже при небольших n значения могут быть нулевыми
5. **Решения:** увеличение обучающего корпуса, сглаживание частот, откат

Борьба с нулями

- *Add-one smoothing* (сглаживание Лапласа):

$$\hat{P}_{AOS}(w_N \mid w_1^{N-1}) = \frac{c(w_1^N) + \delta}{c(w_1^{N-1}) + \delta V},$$

V — это размер словаря, а δ — некоторая фиксированная константа.

Чем плох такой подход?

- *Katz smoothing* (простой откат): если не получается применить модель высокого порядка, пробуем для данного слова модель меньшего порядка с понижающим множителем.

Получим не вероятностное распределение!

Борьба с нулями

- *Jelinek-Mercer smoothing* (интерполяционное сглаживание): заведем вектор $\bar{\lambda} = (\lambda_1, \ldots, \lambda_N)$, такой, что $\sum_i \lambda_i = 1$ и $\lambda_i \geqslant 0$. Тогда

$$\hat{P}_{IS}(w_N \mid w_1^{N-1}) = \sum_{i=1}^{N} \lambda_i \hat{P}_S(w_N \mid w_{N-i+1}^{N-1}).$$

- Другие виды сглаживаний:
  - *Good-Turing estimate*
  - *Witten-Bell smoothing*
  - *Absolute discounting*
  - *Kneser-Ney smoothing*

Пример: генерация текстов Шекспира

Попробуем генерировать тексты с помощью HMM:

- **Униграммная модель:**

  To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have. Every enter now severally so, let. Hill he late speaks; or! a more to leg less first you enter.

- **Биграммная модель:**

  What means, sir. I confess she? then all sorts, he is trim, captain. Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

Пример: генерация текстов Шекспира

- **3-граммная модель:**

  Sweet prince, Falstaff shall die. Harry of Monmouth's grave. This shall forbid it should be branded, if renown made it empty. What is't that cried? Indeed the duke; and had a very good friend. Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

- **4-граммная модель:**

  King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; Will you not tell me who I am? It cannot be but so. Indeed the short and the long. Marry, 'tis a noble Lepidus. They say all lovers swear more performance than they are wont to keep obliged faith unforfeited.

Другой пример: исправление опечаток

- Задача большая и разнообразная:
  - ошибки пропуска/вставки символов
  - ошибки неграмотного написания
  - ошибки склейки/расклейки слов
  - неверная раскладки
  - написание транслитерацией
- В классической постановке рассматриваем первые два случая
- Формально можно записать модель «шумного канала»:
  - исходное слово w подаётся в канал и искажается в слово $\hat{w}$
  - хотим восстановить исходное слово w (предполагаем в уме наличие контекста):

$$w = \operatorname*{argmax}_{v \in W} p(v \mid \hat{w}) = \operatorname*{argmax}_{v \in W} \frac{p(v, \hat{w})}{p(\hat{w})} = \operatorname*{argmax}_{v \in W} p(v)\, p(\hat{w} \mid v)$$

- $p(v)$ – языковая модель, $p(\hat{w} \mid v)$ – модель канала

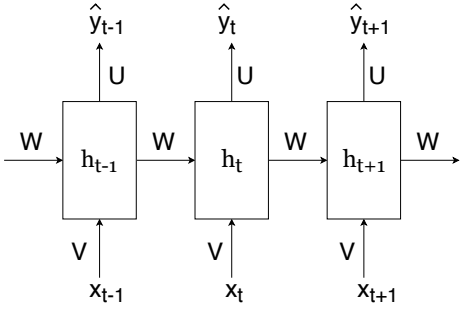
Рекуррентные нейронные сети

- Обычные нейронные сети плохо подходят для обработки последовательностей, поскольку наблюдают только текущий элемент
- Для учёта контекста используются *рекуррентные нейронные сети* (RNN/LSTM/GRU)

Примеры задач:

- Распознавание речи/музыки
- POS-теггинг, NER
- Распознавание рукописного текста
- Распознавание/генерация печатного текста
- Анализ временных рядов
- Машинный перевод

Модель рекуррентной нейронной сети (RNN)



h_t — скрытое состояние в момент t

$$h_t = f(Vx_t + Wh_{t-1} + b)$$

$$\hat{y}_t = g(Uh_t + \hat{b})$$

Обучение сети — минимизация суммарных потерь:

$$\sum_{t=1}^{n} \mathcal{L}_t(y_t, \hat{y}_t) \to \min_{V,U,W,b,\hat{b}}$$

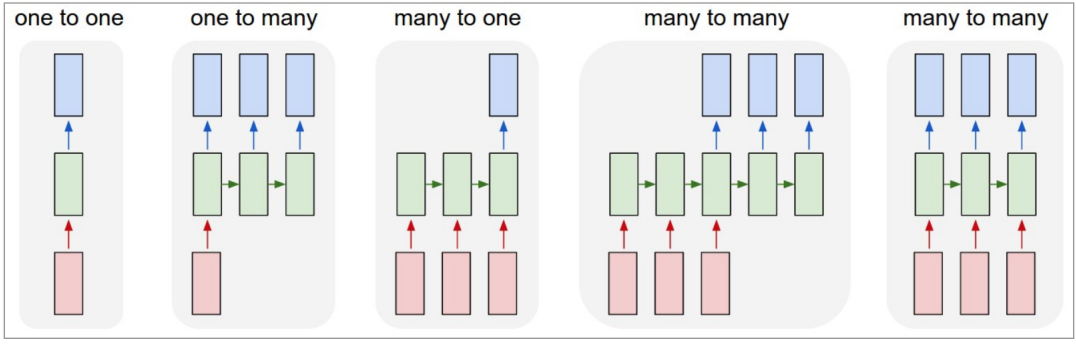
Сеть обучается с помощью алгоритма Backpropagation[1]

[1] Часто, вариацию алгоритма Backpropagation для обучения RNN называют Backpropagation through time

Архитектура LSTM

- В реальности обычная RNN хранит информацию только о коротком контексте (затухание градиентов)
- Такого недостатка лишена LSTM – нейросетевой рекуррентный блок, состоящий из пяти элементов:
  - Основной слой (как и в обычной RNN)
  - Три сигмоидальных слоя-фильтра
  - Ячейка памяти (вектор) – **дополнительное состояние**
- Каждый слой имеет свои обучаемые веса
- Каждый фрейм LSTM передаёт не только свои выходы, но и состояние ячейки памяти

Виды RNN



Ссылка на источник картинки

Пример с Хабра: генерация стихов

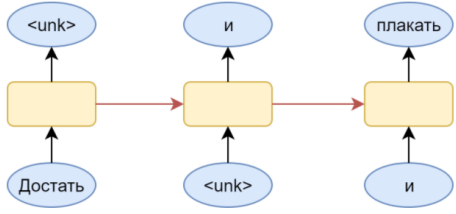
- **Задача:** написать генератор стихов на русском языке
- В основе лежит рекурентная сеть **с кучей обвязок**
- Обучаем на данных `http://stihi.ru/` (+ морфологическая разметка)
- Объект выборки – строка стихотворения

Умоляю перестань мне сниться
Я люблю тебя моя невеста
Белый иней на твоих ресницах
Поцелуй на теле бессловесном

Ссылка на статью

Обработка отсутствующих слов

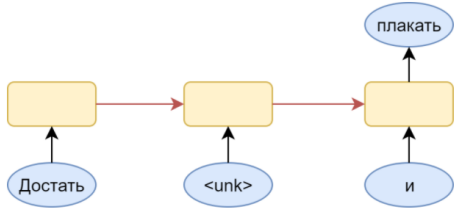
- Словарь может состоять из миллионов слов, но часто его приходится сильно фильтровать
- Вместо отсутствующего слова берём <unk>:



- В модели предсказания слова по предыдущему выдаваемое распределение на словах сместится в пользу <unk>
- **Выход:** можно сэмплировать без него, но получается плохо

Обработка отсутствующих слов

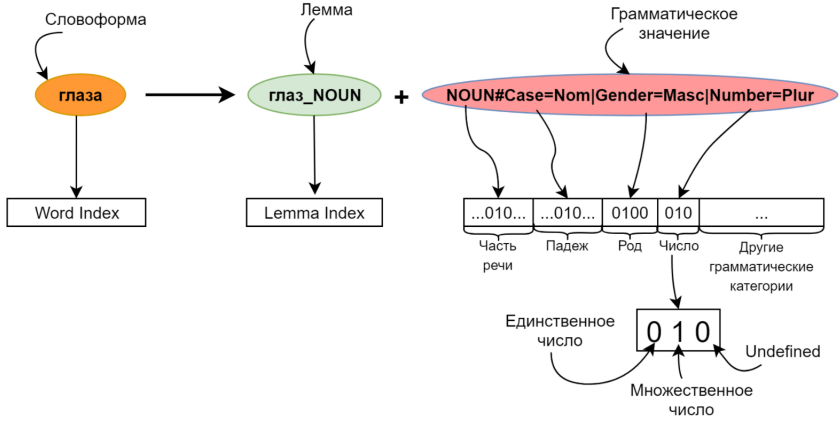
- Альтернатива – предсказывать слова по цепочке предыдущих:



- Из обучающей выборки придётся нарезать всевозможные цепочки, что приведёт к её существенному увеличению
- Зато можно выкинуть все цепочки, заканчивающиеся неизвестным словом

Доработка входного слоя

- Необходимо сократить размерность выходного слоя

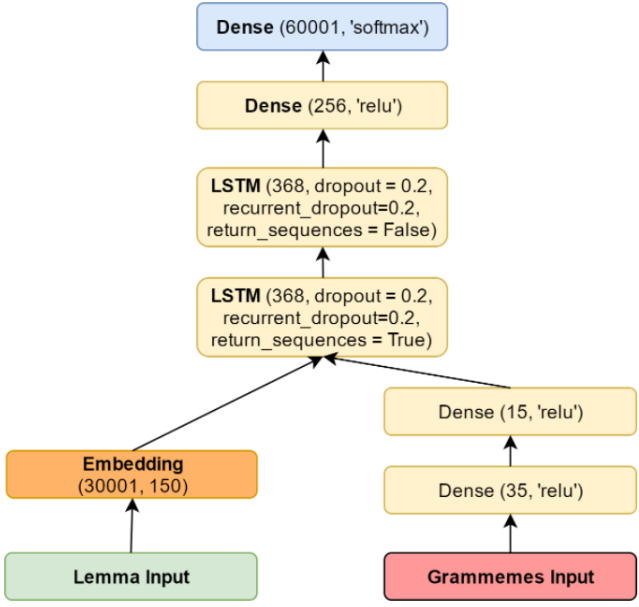


- Можно использовать уже предобученные эмбеддинги для лемм (например, от RusVectores)

Доработка выходного слоя

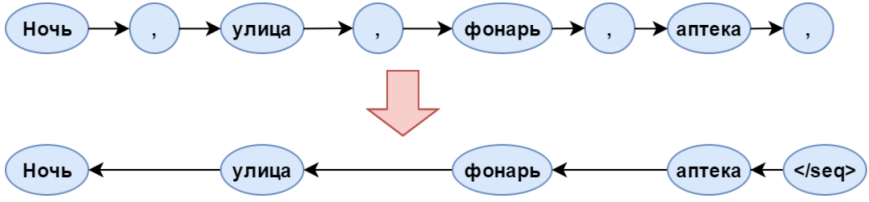
- Вместо индекса слова можно предсказывать по-отдельности лемму и грамматическое значение
- **Проблема:** у сэмплированной леммы может не оказаться нужного грамматического значения
- **Варианты решения:**
  1. выбирать наиболее вероятную пару «лемма + грамматическое значение» из существующих
  2. выбирать наиболее вероятное грамматическое значение среди возможных для сэмплированной леммы

Итоговая архитектура сети (keras)



Обработка данных

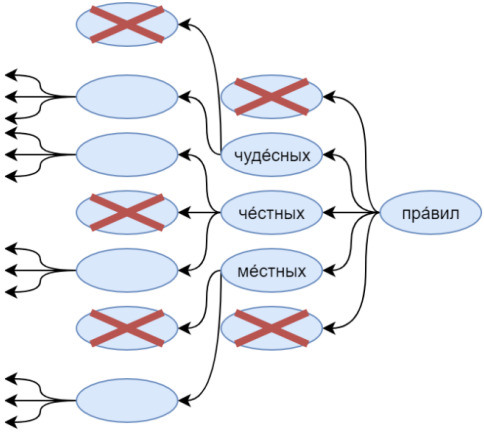
- В конец каждой строки добавляется завершающий символ
- Строки инвертируются для упрощения рифмовки при генерации
- Из выборки удалены знаки препинания (сеть сильно обучается на запятых и многоточиях)



Правила фильтрации

- Есть модель-генератор, нужно фильтровать слова так, чтобы получались именно стихотворения
- Метрические правила определяют последовательность ударных и безударных слогов в строке
- Правила рифмы допускают только словоформы, которые корректно рифмуются (слова с одной леммой рифмовать запрещено)
- Ударения получаются путём обучения классификатора на словаре, рифмы – эвристическими правилами

Лучевой поиск (beam search)



- В результате работы фильтров могло не остаться ни одного слова
- Для борьбы с этим применим beam search:
  - для первого слова генерируем N наиболее вероятных продолжений
  - для каждой из этих N последовательностей получаем ещё N наиболее вероятных продолжений
  - из всех полученных последовательностей выбираем N наиболее вероятных
  - и т.д.

Пример результата

Так толку мне теперь грустить

Что будет это прожито

Не суждено кружить в пути

Почувствовав боль бомжика

Забегая вперёд: 'болталка' Алисы

- Подробно про Алису поговорим позже
- Если кратко, пайплайн обработки запроса выглядит так:
  - Пробуем распознать интент запроса, если распознали – выделяем слоты и дальше отвечаем по сценарию
  - Если интент не распознался – проверяем, не является ли запрос поисковым, если является - ищем и показываем результат
  - Если не является – включаем 'болталку'
- 'Болталка' – система на основе сиамской сети, которая получает на вход запрос и контекст и пытается подобрать наилучший ответ
- Разумеется, на её ответы накладываются значительные ограничения
- Это не генеративная модель, поскольку ответы всё равно подбираются из уже существующих
- Но база огромна, так что пользователь скорее всего получит нетривиальный ответ

Ссылка на статью

Забегая вперёд: GPT-2

- GPT-2 относится к классу огромных глубоких нейросетевых моделей, совершивших качественный скачок в развитии NLP
- Это языковая модель, то есть сеть учится по контексту предсказывать следующее слово
- Архитектура похожа на трансформер, в отличие от BERT она не требует тюнинга
- Сеть с огромным числом парамтеров (1.5 млрд.) была обучена на 8 млн. текстах, накрауленных из различных источников в Интернете
- Результаты получились реально крутые
- Полную модель так и не выложили

Ссылка на статью

Что может GPT-2

- Генерация связного текста на несколько страниц с перекрёстными отсылками, сохранением хода мысли и персонажей
- Генерация ответов на правильным образом сформулированные вопросы (например, продолжение определений)
- Генерация суммаризации входного текста путём добавления в конец 'TL;DR'
- Перевод текста (при подходящей структуре входного текста)

Резюме лекции

- Языковые модели – одна из центральных конструкций в анализе текстов
- Простые модели для решения прикладных задач можно строить статистически
- Большие нейросетевые языковые модели – один из самых сильных инструментов современного NLP
- К сожалению, они не всегда есть в нужном домене и их очень сложно и дорого обучать