A BEGINNER'S GUIDE TO CHATBOTS

Саша Ершова aleks-ershova@ya.ru

July 31, 2018

С ЧЕГО НАЧАТЬ?

С ЧЕГО НАЧАТЬ?

1. Постановка задачи: зачем нам бот?
2. Разработка стратегии поведения бота
3. Какие методы лучше всего подходят выбранной стратегии?

Какие задачи решают чатботы?

- Потребность человека в общении («болталка» Алисы)
- Интерфейс приложения (любой чатбот для заказа пиццы)
- Голосовое управление (Siri, Alexa, Алиса)

РАЗРАБОТКА СТРАТЕГИИ БОТА

- О чём люди должны спрашивать чатбота?
- Какие интенты внутри поддерживаемых доменов бот должен обрабатывать?
- Какие сущности могут быть в каждом сценарии? Какие ограничения?
- Как будут обрабатываться ошибки?
- Как будут обрабатываться неподдерживаемые домены?
- Что будет, если на бота будут материться? Оскорблять? Приставать?
- Проработка персонажа.

РАЗРАБОТКА СТРАТЕГИИ БОТА

- Домен (топик) — e.g. ”заказ билета”
- Интент (намерение пользователя — ”поиск билета”)
- Сущность (”Лондон”, ”31 декабря”, ”эконом-класс”)

Open domain — чатбот умеет поддерживать разговор на любую тему.
Closed domain — чатбот умеет поддерживать беседу на ограниченное количество тем.

РАЗРАБОТКА СТРАТЕГИИ БОТА

Диалог vs. беседа:
Диалог — единичная реализация сценария.
Беседа = диалог + диалог + диалог

Поддержка состояний — когда бот запоминает предыдущие реплики человека. Это можно делать как в рамках одного диалога, так и в рамках всей беседы.

Зачем нужна поддержка состояний?

На уровне беседы:

- Запомнить информацию о юзере (имя, гендер, адрес, возраст, предпочтения, что угодно)
- Сохранить настройки бота

Зачем нужна поддержка состояний?

На уровне диалога:

- **Разрешение эллипсиса**
  - Какая сегодня погода?
  - N градусов
  - А завтра [какая погода]?
- **Разрешение анафоры**
  - Где находится Москва?
  - <ответ с локацией>
  - Какая в ней погода?
- **Сценарии длиннее одного шага**
  Например, заказ пиццы.

Откуда брать ответы?

- **Retrieval-based**
  Ответы ищутся в базе.
- **Generative-based**
  Ответы создаются (почти) с нуля.

RETRIEVAL-BASED

RETRIEVAL-BASED

- База готовых ответов
- Form-filling

КАК ВЫБРАТЬ ОТВЕТ?

- Правила (Юзер сказал А —> отвечаем Б)
- Ранжирование по похожести запроса на ответ
  - расстояние Левенштейна
  - векторная близость (tf-idf, дистрибутивная семантика, etc.)
- Машинное обучение (классификаторы, LSTM-нейросети)

РАССТОЯНИЕ ЛЕВЕНШТЕЙНА

Есть два слова. Нужно превратить первое во второе. За один шаг мы можем либо добавить букву в слово, либо убрать букву, либо заменить букву на другую.

Расстояние Левенштейна — минимальное количество шагов, которое требуется, чтобы превратить первое слово во второе.

РАССТОЯНИЕ ЛЕВЕНШТЕЙНА

Пример: найдём расстояние между словами ”мама” и ”лампа”.

мама $\rightarrow$ мампа $\rightarrow$ лампа

Нам понадобилось 2 шага, следовательно, расстояние Левенштейна между ”мамой” и ”лампой” — 2.

Можно (и эффективнее для чатботов) брать в качестве токенов не символы строки, а отдельные слова.

TF-IDF

Матрица term-document:
Есть корпус из N документов, во всём корпусе M уникальных слов. Составляем матрицу MxN, где в строках будут тексты, а в столбиках — слова. На пересечении указывается мера tf-idf для этого слова в этом документе.

TF-IDF

TF (term frequency) — отношение количества вхождений слова в текст к длине текста.

$$\mathrm{tf}(t, d) = \frac{n_t}{\sum_k n_k}$$

IDF (inverted document frequency) — обратная частота документа.

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{ d_i \in D \mid t \in d_i \}|}$$

TF-IDF — это TF * IDF.

TF-IDF

После заполнения матрицы каждый текст превращается в вектор длины M. Кроме того, мы можем взять любой текст и преобразовать его в вектор по этой матрице.
А уже между векторами можно посчитать косинусную близость.

Как применить к чатботу?

1. Превращаем базу ответов в матрицу term-document.
2. Берём запрос и раскладываем его по этой матрице.
3. Находим в матрице вектор, ближайший к вектору запроса.
4. ???
5. PROFIT!

FORM-FILLING

Для каждого интента есть готовый шаблон ответа, который заполняется данными из запроса и данными извне.

- Я хочу взять билет на самолёт из Москвы в Лиссабон на 1 ноября.
- Билет из Москвы в Лиссабон на 1 ноября будет стоить 13 000 рублей.

Сложность: как корректно извлечь сущности из запроса на естественном языке?

GENERATIVE-BASED

Зачем и когда использовать?

- Есть много данных.
- **Мы уверены в качестве используемых данных.**
- Нет ресурсов на разработку правил.
- Чатбот скорее развлекательный.
- Нам не очень жалко свою репутацию.

Как генерировать текст?

- Марковские цепи
- Sequence-to-sequence нейросети

В качестве токенов можно использовать как слова, так и отдельные символы. Реже — целые предложения.

МАРКОВСКИЕ ЦЕПИ

Вероятность появления каждого следующего слова вычисляется на основе предыдущего или предыдущих.

Как реализовать?

1. Берём обучающий корпус текстов.
2. Каким-то образом выбираем слово, с которого начнём цепь.
3. Какое слово с наибольшей вероятностью последует за ним, судя по материалам корпуса?
4. А за этими двумя?
5. И так далее.

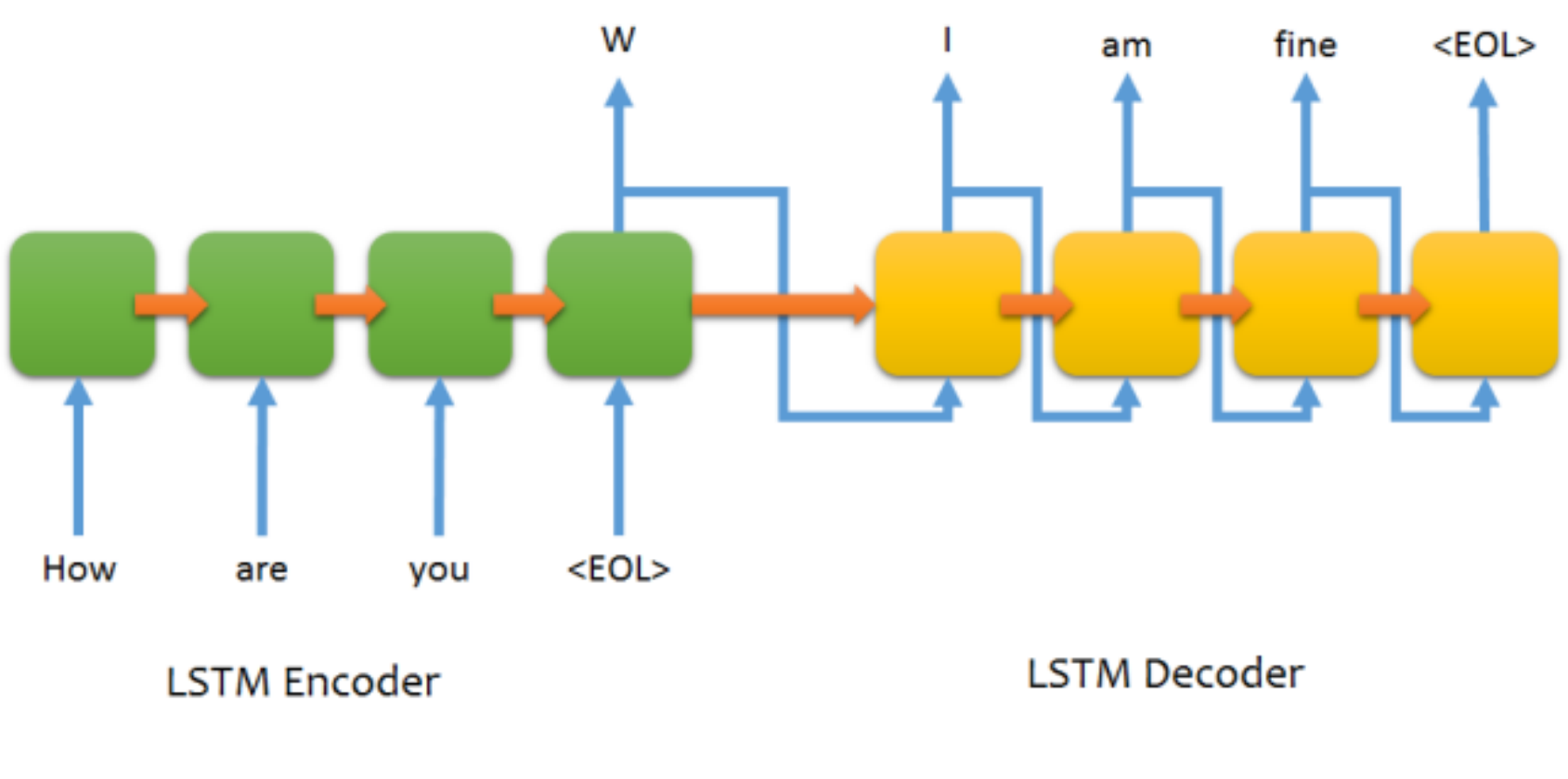
SEQ2SEQ-НЕЙРОСЕТИ

Нейросеть вида ”sequence to sequence” принимает на вход некоторую последовательность чего угодно (чисел, пикселей, символов) и порождает другую последовательность чего угодно, сооветствующую первой. Нас скорее интересуют последовательности букв или слов, хотя в отдельных случаях в качестве токенов используются целиком вопросы и ответы.

Как реализовать?

1. Берём обучающий корпус запросов и ответов.
2. Тренируем рекуррентную (LSTM или GRU) нейросеть, которая принимает на вход последовательность токенов запроса и выдаёт в качестве ответа некую другую последовательность токенов.

SEQ2SEQ-НЕЙРОСЕТИ



Спасибо за внимание!