Математические методы анализа текстов

Лекция

# Чат-боты и мобильные ассистенты. Вопросно-ответные системы.

Мурат Апишев (mel-lain@yandex.ru) [1]

3 декабря, 2019

[1] Подготовлена с использованием материалов курса «Анализ неструктурированных данных» ФКН НИУ ВШЭ (лекция Екатерины Артёмовой)

# Сразу уточним терминологию

- **Вопросно-ответные системе $\neq$ чат-боты**
- Несмотря на кажущуюся близость, эти решения имеют в основе существенно различные технологии
- Текстовые боты устроены гораздо проще, их могут создавать даже люди без данных и экспертизы в ML
- QA-система – это сложная ML-конструкция, которая умеет
  - понимать запросы на естественном языке, искать ответы
  - искать ответы на них в огромных массивах неструктурированных данных
  - генерировать ответ на понятном человеку языке
- В отличие от чат-ботов, создание QA-систем – это сложный процесс, требующий хорошего владения NLP и больших массивов данных
- Однако обе системы очень похожи с точки зрения пользовательского интерфейса («задал вопрос – получил ответ»)

# Неужели чат-боты ещё не всем надоели?

- Удивительно, но нет, тема на подъёме
- Чаще всего ботов используют в контакт-центрах
- Текущие решения, внедрённые в больших компаниях, в основном примитивны
- И всё равно это оказывается выгодным
- Голосовые ассистенты – это тоже чат-боты, но более сложные и «умные», умеющие работать с голосовыми технологиями
- Объём рынка голосовых устройств во всём мире вырос экспоненциально за последние годы и продолжает расти

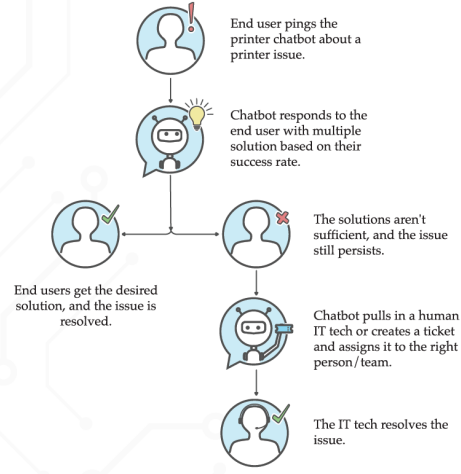
https://www.iconfinder.com/icons/3948214/bot_chatbot_live_chat_message_messenger_online_speech_bubble_icon

# Три условных уровня разговорного «AI»

1. **Проблемно-ориентированные системы**
   - Заточены под одну или несколько смежных тематик
   - Могут учитывать только локальный контекст диалога
   - Неспособный действовать за пределами своих сценариев
   - В простейшем случае вообще заточены под ответы на несколько вопросов
2. **Системы общего назначения (например, голосовые ассистенты)**
   - Подходят для работы со всеми темами, покрывающими основные пользовательские проблемы
   - Могут учитывать только локальный контекст диалога
   - Иногда могут вести почти осмысленную беседу в рамках диалога
3. **Системы-помощники (таких ещё нет)**
   - В разумных пределах покрывают все основные потребности человека при взаимодействии с информационными системами
   - Учитывают глобальный контекст и дообучаются
   - Возможно, смогут вести осмысленные беседы – кто знает...

# Что представляет собой чат-бот

- Современные чат-боты работают по сценариям и скриптам (конечный автомат)
- Никаких реальных ML-моделей (deep или не deep), способных стабильно вести диалог, не существует
- Почему так – обсудим чуть позже
- Есть две стандартные задачи, которые нужно решать при создании чат-бота:
  - определение интента (т.е. типа пользовательского запроса)
  - заполнение слотов – извлечение из текста пользователя информации, нужной для ответа на запрос по данному интенту



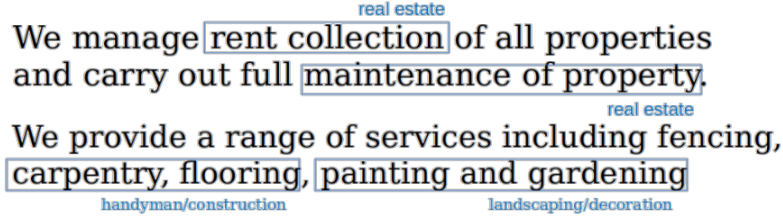

https://www.dataversity.net/ai-advantage-itsm-features-use-cases

# Определение интента

- В начале диалога нужно понять, с какой областью (и с каким сценарием) связана реплика пользователя, примеры:
  - запрос на определение погоды
  - запрос на бронирование столика в ресторане
  - поисковый запрос
- Внутри диалога требуется определять по очередной фразе, в какую ветку сценария нужно переходить
- В обоих случаях решается задача классификации
- Решать её можно по-разному:
  - поиск по словам
  - соответствие шаблонам или регулярным выражениям
  - одна или несколько моделей для классификации

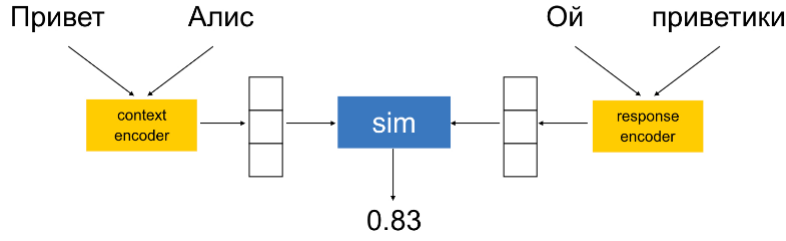
# Slot Filling

- Для выявленного интента нужно собрать необходимую информацию, примеры:
  - запрос на определение погоды: геолокация и метка времени
  - запрос на бронирование столика в ресторане: геолокация, название ресторана, метка времени, число персон
  - поисковый запрос: обычно вся реплика интепретируется как запрос
- Искать сущности тоже можно по-разному:
  - эвристические правила, морфологические шаблоны
  - регулярные выражения
  - модели для NER



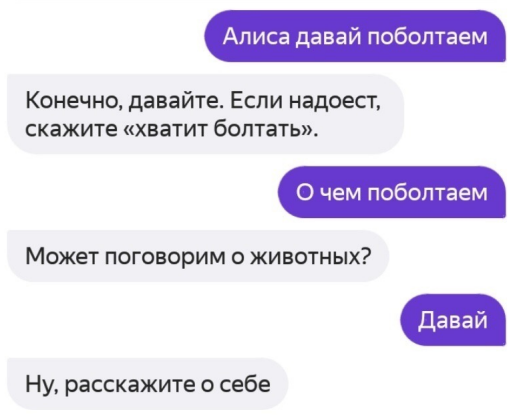
# Неужели всё только по сценарию?

- Современные чат-боты работают по сценариям и скриптам (конечный автомат)
- Но иногда чат-бот имеет функцию «общения» на свободные темы (без выявления специфического интента)
- Пример – «болталка» Алисы
- Ответы генерируются на основе контекста и ранжируются с помощью DSSM-подобной нейросети



# Почему всё-таки генеративный подход непопулярен

- Основное назначение чат-бота – помощь, в решение конкретных задач, а не развлечение репликами
- Отсюда следуют основные недостатки генеративного подхода по сравнению со сценарным:
  - непредсказуемость полезности результата
  - возможность несогласованности слов в ответе и отсутствия в нём смысла
  - очень сложные обучение и настройка, необходимость постоянного дообучения и валидации результатов
  - потребность в огромных массивах данных для обучения




http://joyreactor.cc/post/3994138

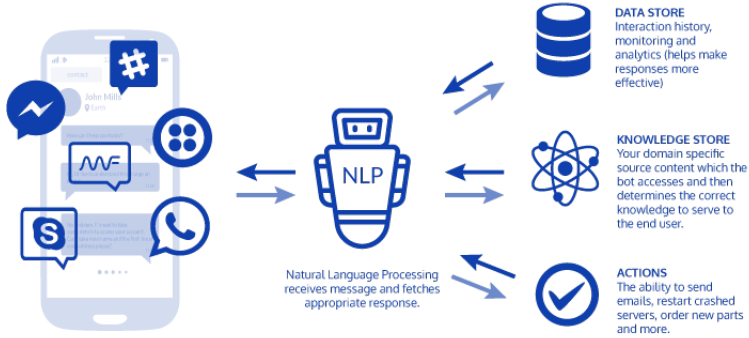
# Всё на самом деле серьёзно: пример

- Чатботы с голосовыми интерфейсами часто встраивают в колонки и умные экраны
- В США более половины домохозяйств уже имеют умную колонку
- Китай активно догоняет, весь мир следом – тоже
- Всё чаще пользователями устройств становятся дети
- Они задают вопросы, просят поставить мультики, музыку или рассказать сказки
- Представьте, что будет, если скрипты в Alexa заменить на аналог «болталки»...

https://www.securitylab.ru/news/498751.php

# Платформы для создания чат-ботов

- Facebook WIT
- Amazon Lex
- Google Dialogflow
- Microsoft LUIS
- Just AI CP
- IBM Watson
- ManyChat
- Chatfuel




https://enterprisebotmanager.com/chatbot-architecture-and-how-they-work/

# Типы вопросов для QA-систем

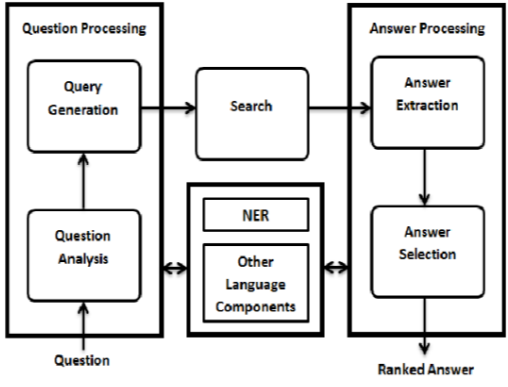
- **Фактологические вопросы:**
  - Какого цвета спелый помидор?
  - Какие традиции есть на Алтае?
- **Вопросы на понимание здравого смысла:**
  - Что можно положить в холодильник из списка: слон, яблоко, катер?
- **Оценочные и сравнительные вопросы:**
  - Какая обжарка кофе вкуснее, светлая или темная?
  - В какой московской кофейне делают самый вкусный кофе?
- **Вопросы, задаваемые по тексту** (machine reading comprehension)
- **Открытые вопросы** (open domain QA, ODQA)

***Типы ответов на вопросы:*** бинарный, несколько вариантов, текстовый фрагмент

# Основные парадигмы для фактологических QA-систем

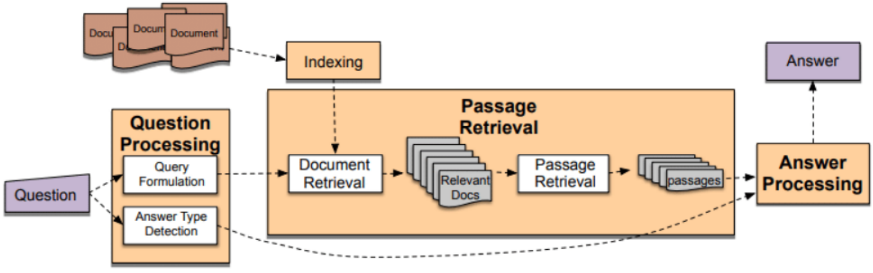
1. **Алгоритмы, похожие на алгоритмы информационного поиска (IR-based QA)**: ищем текстовый фрагмент, отвечающий на вопрос
2. **Алгоритмы поиска по базам знаний (KB-based QA)**: строим запрос на основании текста вопроса и обращаемся с ним в базу

   *Когда умер Владимир Ленин?* $\rightarrow$ *дата-смерти(Владимир Ленин, ?x)*




https://www.researchgate.net/figure/Common-Architecture-of-QA-systems_fig1_269931090

# IR-based QA-системы




1 Обработка вопроса

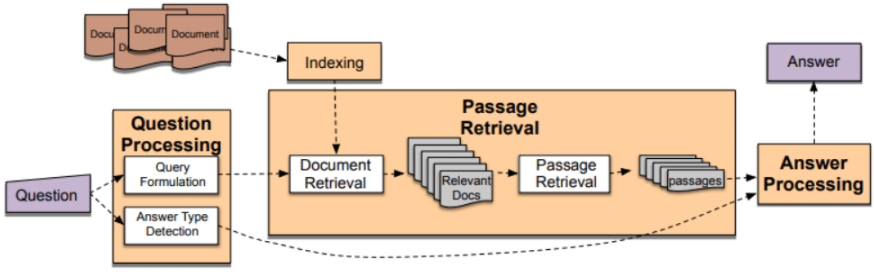
- определение типа ответа (персона, геолокация, метка времени)
- определение ключевых сущностей
- определение типа вопроса

2 Формирование запроса

- переформулировка запроса: удаление wh-слов, изменение порядка слов
- расширение запроса (+ синонимы, исправление опечаток и т.п.)

https://www.researchgate.net/figure/The-architecture-of-IR-based-question-answering-system_fig1_328861207

# IR-based QA-системы




3 Извлечение документов и фрагментов из них

4 Извлечение и обработка ответа:

*Что означает апостроф в английской транскрипции?*

*На самом же деле, в* ***транскрипции английского*** *ничего сложного нет. Если вы . . . это* ***ударение****, которое в* ***транскрипции*** *отмечается* ***апострофом***

# Датасеты для IR-based QA

**Passage**: Tesla later approached Morgan to ask for more funds to build a more powerful transmitter. **When asked where all the money had gone, Tesla responded by saying that he was affected by the Panic of 1901**, which he (Morgan) had caused. Morgan was shocked by the reminder of his part in the stock market crash and by Tesla's breach of contract by asking for more funds. Tesla wrote another plea to Morgan, but it was also fruitless. Morgan still owed Tesla money on the original agreement, and Tesla had been facing foreclosure even before construction of the tower began.

**Question**: On what did Tesla blame for the loss of the initial money?

**Answer**: Panic of 1901

Рис.: Пример сэмпла из датасета SQuAD

- Stanford Question Answering Dataset (SQuAD)
- NewsQA
- WikiQA
- MedQuAD
- CuratedTREC
- WebQuestions
- WikiMovies
- SberQuAD (на русском)

# Датасет SQuAD2.0

- **Состав:**
  - 100k вопросов в SQuAD1.1
  - + более 50k вопросов без ответов в SQuAD2.0
- **Формирование:**
  - Отобраны топ-10k статей английской Википедии по PageRank, из них случайно выбрано 536
  - Каждая статья была разбита на параграфы
  - Задача для краудсорсинга: сформулировать до 5 запросов к каждому из параграфов
  - Вопрос формулировался своими словами, без копирования текста из параграфа
  - Анализ: разнообразие ответов, сложность вопросов, степень синтаксического различия между вопросом и ответом
- **Ссылки:**
  - `https://arxiv.org/abs/1606.05250`
  - `https://arxiv.org/abs/1806.03822`
  - `https://rajpurkar.github.io/SQuAD-explorer`

# Датасет MS Marco

| Field | Description |
|---|---|
| Query | A question query issued to Bing. |
| Passages | Top 10 passages from Web documents as retrieved by Bing. The passages are presented in ranked order to human editors. The passage that the editor uses to compose the answer is annotated as is_selected: 1. |
| Document URLs | URLs of the top ranked documents for the question from Bing. The passages are extracted from these documents. |
| Answer(s) | Answers composed by human editors for the question, automatically extracted passages and their corresponding documents. |
| Well Formed Answer(s) | Well-formed answer rewritten by human editors, and the original answer. |
| Segment | QA classification. E.g., tallest mountain in south america belongs to the ENTITY segment because the answer is an entity (Aconcagua). |

Рис.: Формат датасета

Три задачи:

- предсказать, можно ли на вопрос дать ответ, и если да, то сгенерировать правильный ответ
- генерация ответа в подходящей и понятной форме
- ранжирование готовых ответов (фрагментов) по релевантности запросу

`http://www.msmarco.org`

# Решение задачи SQuAD

- Задача решается в два этапа:
  1. Отбор кандидатов для ранжирования
  2. Классификация слов в кандидатах: для каждого слова модель определяет, является ли оно частью ответа, и если да, то какой
- Первый этап обычно решается простыми и легковесными методами
- Пример: косинусное расстояние между векторами TF-IDF или FastText вопроса и абзацев-кандидатов
- На втором этапе обычно применяются дорогие и тежеловесные нейросетевые модели – рассмотрим ниже несколько примеров
- Дополнительно второй этап требует метод оценки ответов для выбора из нескольких возможных

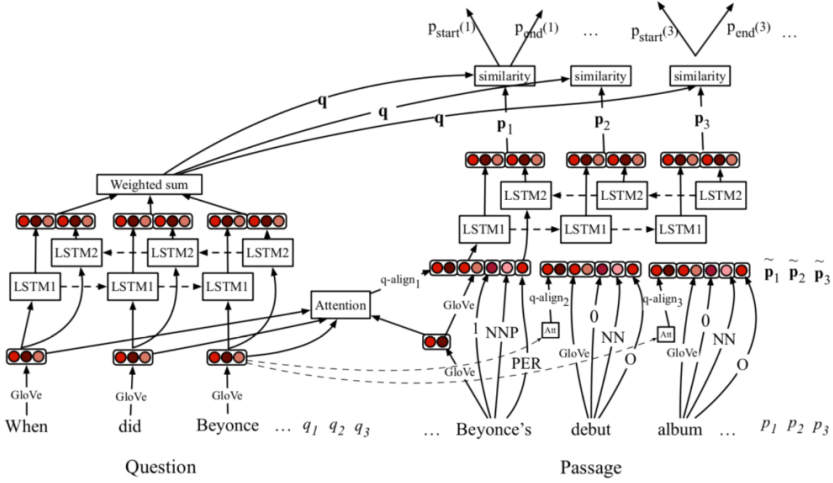
# Модель DrQA

- **Извлечение документов:**
  ищем пять ближайших статей Википедии, используя близость по векторам TF-IDF
- **Чтение документа:**
  получаем запрос $q = q_1, \ldots, q_\ell$ и $m$ параграфов $p_1, \ldots, p_m$
- **Кодирование вопроса:**
  взвешенная сумма выходов BiLSTM на словах запроса
- **Кодирование параграфа:**
  - Для каждого слова берём его эмбеддинг (GloVe)
  - Добавляем флаг точного совпадения с одним из слов запроса
  - Добавляем морфологические признаки
  - Пропускаем через BiLSTM, получаем выходной вектор

`https://arxiv.org/abs/1704.00051`
`https://github.com/facebookresearch/DrQA`

# Модель DrQA

- **Предсказание:**
  - используем $P_{start} \propto \exp(p_i W_s q), P_{end} \propto \exp(p_i W_e q)$
  - выбираем наилучший фрагмент от $i-$го токена до $i'-$го токена, такие что $i \geqslant i' \geqslant i + 15$ и $P_{start}(i) \times P_{end}(i')$ максимально



# Модель BiDAF

1. **Слой символьных эмбеддингов:**
   строим эмбеддинги слов с помощью char-CNN

2. **Слой словарных эмбеддингов:**
   берёт для слов предобученные эмбеддинги

3. **Слой контекстных эмбеддингов:**
   пропускаем первые два вида эмбеддингов через BiLSTM и выходы используем в качестве новых представлений

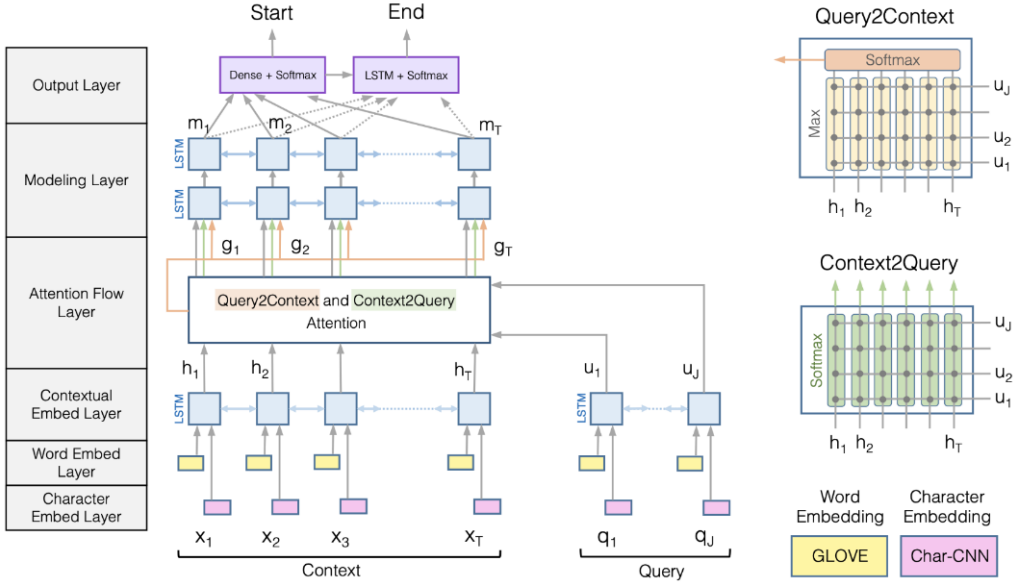
   Первые три слоя применяются и к запросу, и к параграфу (context)

4. **Слой внимания:**
   получаем из векторов слов запроса и параграфа набор векторных признаков для каждого слова параграфа и один вектор, определяющий важность слов параграфа

5. **Слой моделирования:**
   объединяем все векторы со слоёв 3 и 4 и подаём в рекурентную сеть

6. **Выходной слой:**
   формируем ответ как индексы старта и конца фрагмента в параграфе

# Модель BiDAF




https://arxiv.org/abs/1611.01603

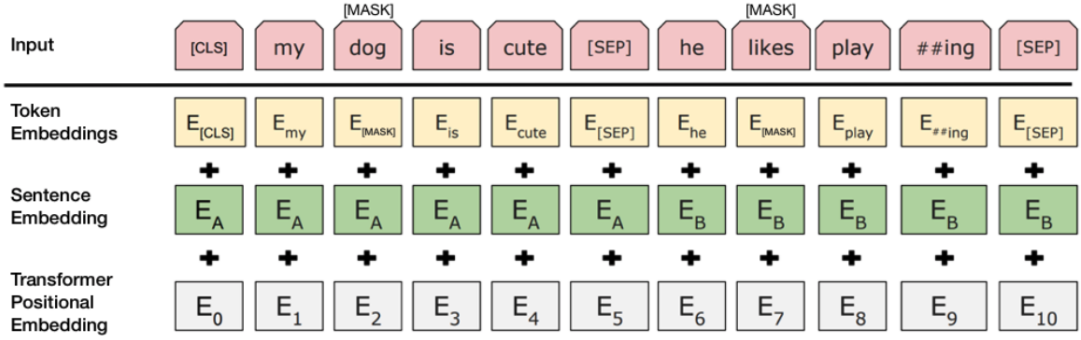
# Какие ещё есть модели

- **R-NET** (https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf)
- **S-NET** (https://arxiv.org/abs/1706.04815)
- **QANet** (https://arxiv.org/pdf/1804.09541.pdf)
- **V-NET** (https://arxiv.org/pdf/1805.02220.pdf)
- **Deep Cascade QA** (https://arxiv.org/abs/1811.11374)

Идеологически все эти модели похожи на DrQA, отличие только в деталях архитектуры и эвристиках

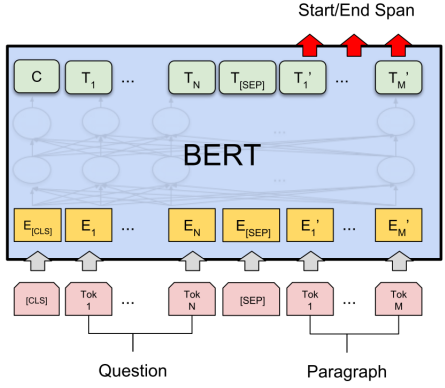
# Напоминание: BERT

- Архитектурно – кодировщик трансформера
- Обучается одновременно на
  - предсказание маскированных слов в последовательности
  - определение последовательной связности пар предложений



# BERT и его модификации в QA

- Предобученные нейросетевые языковые модели (BERT, XLNet etc.) эффективны в определении связей между парами предложений (задачи парафраза и логического следования)
- Задача SQuAD (поиск ответа в абзаце) отлично ложится на архитектуру BERT без дополнительных модификаций
- В лидерборде SQuAD на текущий момент лидируют модели на основе ALBERT и XLNet



# Knowledge Based QA-системы

- Строим запрос на основании текста вопроса и обращаемся с ним в базу
- Имеют малую популярность, поскольку
  - требуются огромных массивов информации
  - текущие алгоритмы построения запросов далеки от совершенства
- Общий пайплайн обработки запроса:

  **Пример:** *Когда умер Владимир Ленин?*

  1. Выделение именованных сущностей (*Владимир Ленин*)
  2. Классификация вопроса и определение предиката (, *когда*, *умер*)
  3. Поиск сущности в базе (**проблема**: разрешение неоднозначности)
  4. Формирование итогового ответа
- Существующие системы плохо справляются с вопросами, в которых
  - более одной сущности
  - требуется сделать несколько запросов к базе

  **Пример:** *Когда умерла мать Владимира Ленина?*

# Представление данных

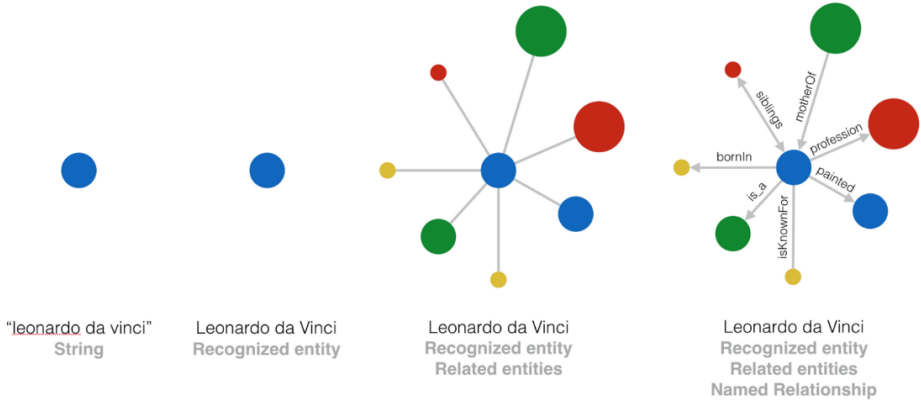



Рис.: Поиск сущностей и предикатов по шаблонам требует удобного представления информации о сущностях

# Датасеты

| | |
|---|---|
| What American cartoonist is the creator of Andy Lippincott? | (andy_lippincott, character_created_by, garry_trudeau) |
| Which forest is Fires Creek in? | (fires_creek, containedby, nantahala_national_forest) |
| What is an active ingredient in childrens earache relief ? | (childrens_earache_relief, active_ingredients, capsicum) |
| What does Jimmy Neutron do? | (jimmy_neutron, fictional_character_occupation, inventor) |
| What dietary restriction is incompatible with kimchi? | (kimchi, incompatible_with_dietary_restrictions, veganism) |

Рис.: Примеры простых вопросов и ответов на них, сделанных по набору данных SimpleQuestions, ответы подчёркнуты

- **SimpleQuestions** – 100k вопросов, написанных и проверенных людьми (`https://arxiv.org/abs/1506.02075`)
- **WebQuestions** – 6k вопросов, собранных автоматически с помощью Google suggest API

# Модель KBQA

- KBQA – модель, использующая для ответов на вопросы данные Wikidata
- Пайплайн ответа на вопрос:
  1. Модель NER выявляет сущности в виде подстрок текста вопроса
  2. Классификатор относит вопрос к одному из предопределённых видов отношений в Wikidata
  3. Найденная с помощью NER подстрока ищется в базе сущностей (по редакторскому расстоянию) – получается список кандидатов
  4. Далее на основании этого списка, сущности и отношения подбирается наиболее вероятный ответ
- `http://docs.deeppavlov.ai/en/master/features/models/kbqa.html`