AliMe Bot: Knowledge-Based Question Answering, and NLP in Real-world Applications

YE Yuxiao

Alibaba Group

yeyuxiao@outlook.com

2018/12/26

Overview

1. AliMe Bot
   - Introduction
   - KBQA
2. Two Classic NLP Tasks
   - Sequence Labelling
   - Text Classification
3. Linguists' Role in Real-world NLP
   - Data Collection
   - Feature Engineering

CV

- 03/2018 - present, Algorithm Engineer, Alibaba Group
- 09/2017 - 01/2018, Research/Teaching Assistant, Tsinghua University
- 09/2016 - 09/2017, MSc in Speech and Language Processing, University of Edinburgh
- 09/2012 - 07/2016, BA in Foreign Languages and Literatures, Tsinghua University

Section 1

AliMe Bot

Chatbot

A **chatbot** is a computer program which conducts conversations with humans.
E.g., Siri by Apple, Alexa by Amazon.

To be more specific: Chit-chat bot, QA bot, Task bot, ...

AliMe Bot: Chatbot for Smart Service

AliMe is a chatbot specialized in the area of smart service. Most of you may have seen it on Taobao, as a customer service coordinator.

But now it is also used in many other areas (e.g., banking, government services, telecommunication).

Knowledge-Based Question Answering

AliMe Bot is actually composed of many tiny chatbots. Different components solve different tasks: some are responsible for casual chat, some for QA, some for task-oriented conversations.

My team builds QA bots. Particularly, we focus on Knowledge-Based Question Answering (KBQA).

To answer questions, machines first need to have related knowledge. They can obtain such knowledge by searching the Internet (Web-Based QA), unstructured documents (Document-Based QA), or structured knowledge bases (Knowledge-Based QA).

AliMe-KBQA in 11.11 Global Shopping Festival

Knowledge Base: Nodes and Edges

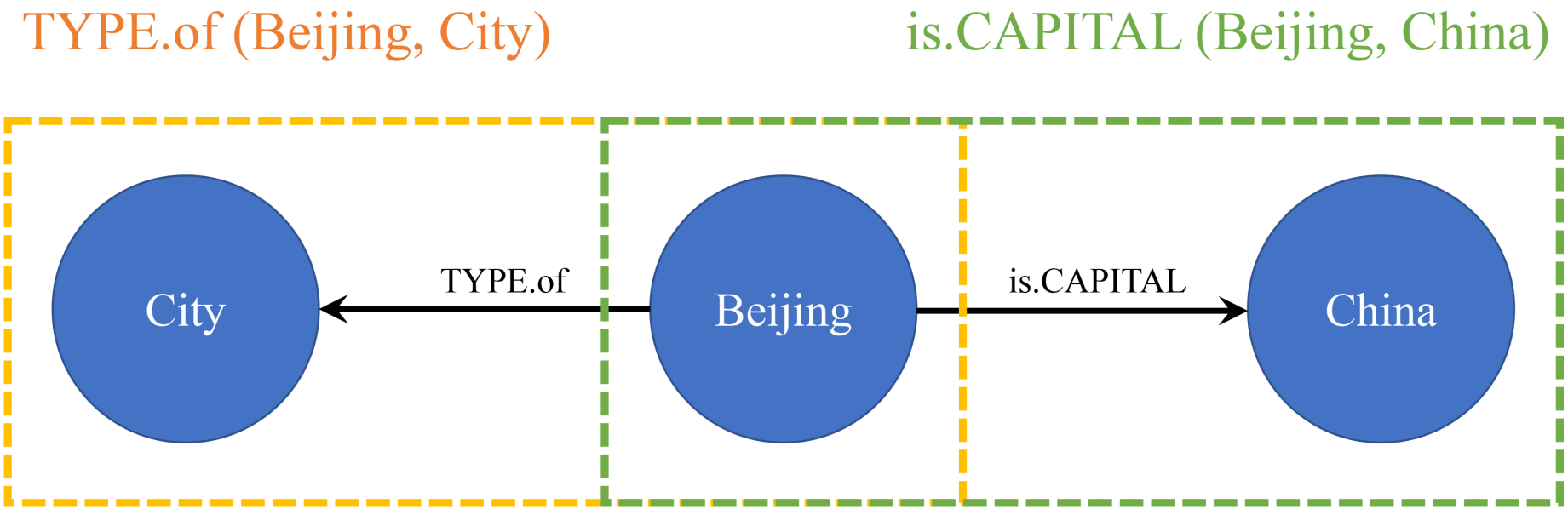
KB: curated datasets with well defined schema

A simplified Freebase-styled KB consists of:

- Entity (node): objects/values in the real world (e.g., Beijing, 2018)
- Predicate (edge): relation between entities (e.g., TYPE.of, is.CAPITAL)

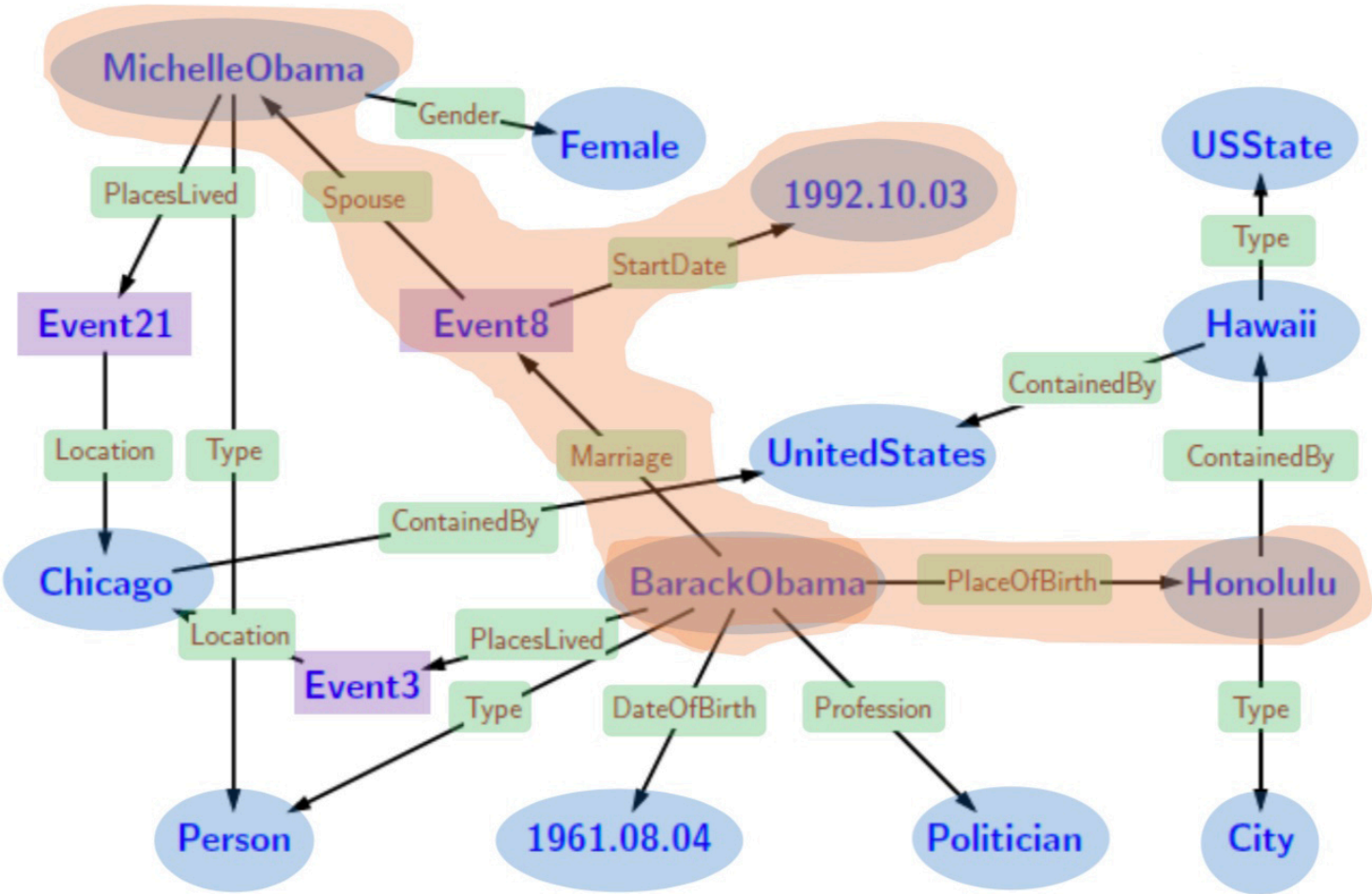
Knowledge Base: Triples

A fact in a KB can be represented by a Triple $<e_1, p, e_2>$.
E.g., <Beijing, TYPE.of, City>, <Beijing, is.CAPITAL, China>:



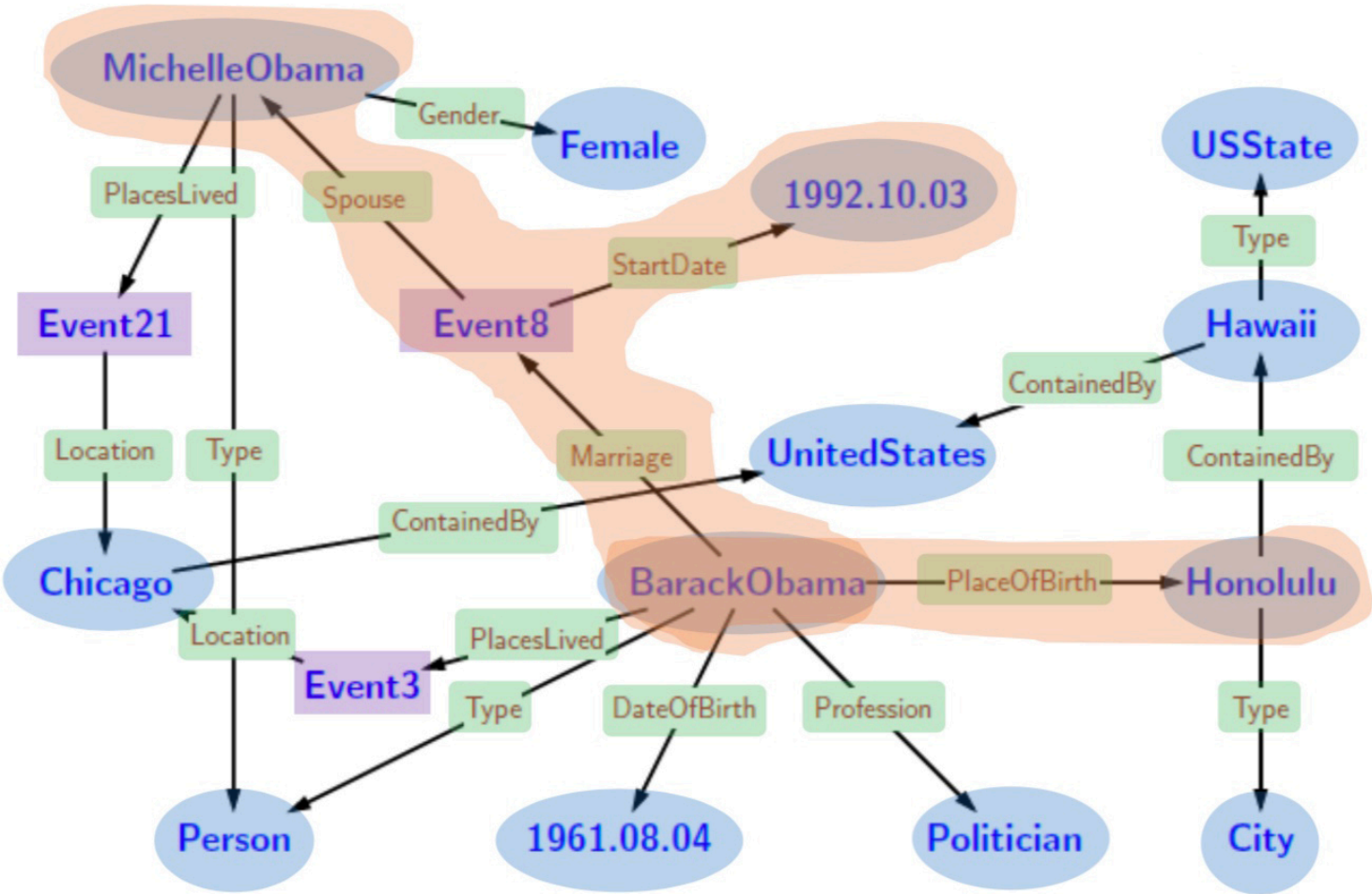
Knowledge Base as a Graph

The whole KB can be viewed as an interlinked graph:



Looking for Answers in a KB

Query: Where was Barack Obama born?

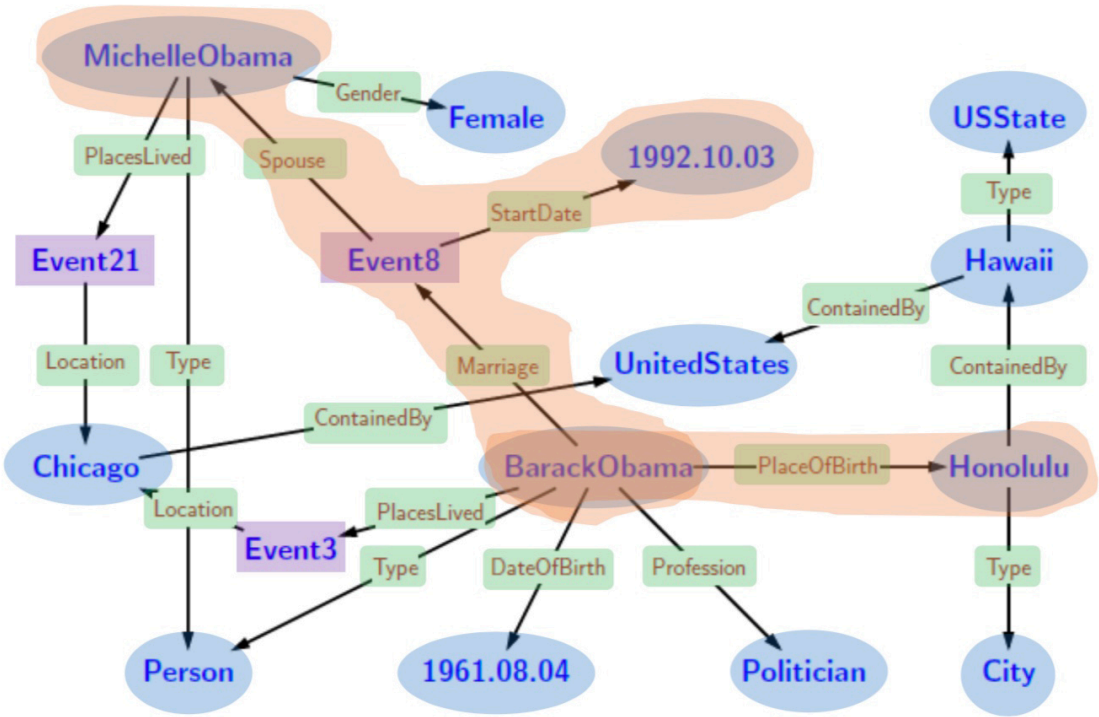


Looking for Answers in a KB

Query: Where was Barack Obama born?

Parsing:

- Entity: Barack Obama
- Predicate: PlaceOfBirth



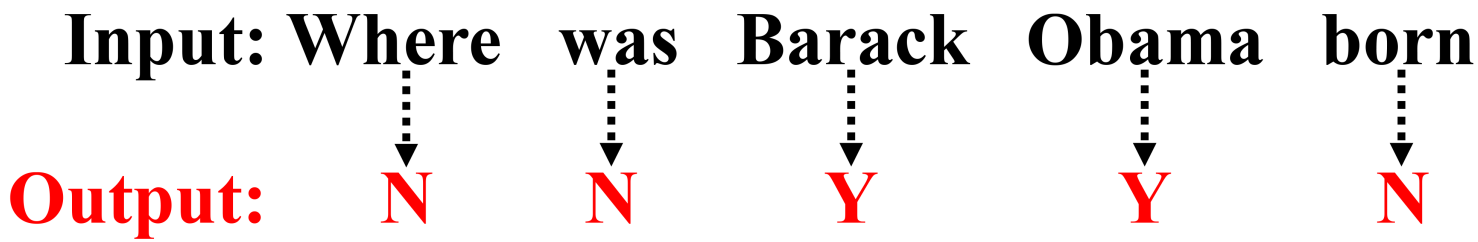
Section 2

Two Classic NLP Tasks

Sequence Labelling

Input: a sequence of tokens (e.g., words, characters)
Output: a sequence of labels (e.g., POS tags, semantic role labels)

How is finding the entity in a query a sequence labelling problem?

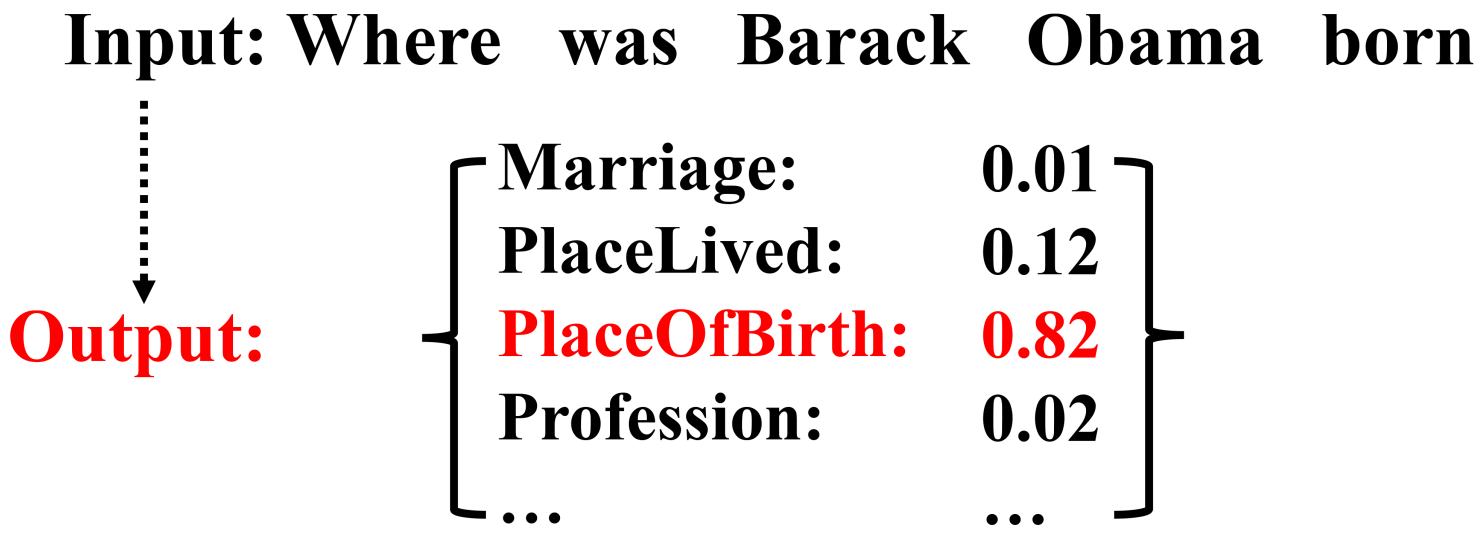


Y: entity
N: non-entity

Algorithms: HMM, ME, CRF, Seq2Seq DNN, etc.

Text Classification

Input: a sequence of tokens (e.g., words, characters)
Output: a label (or multi-label) for that sequence (e.g., sentiment categories, news domains)



Algorithms: LR, SVM, GBDT, CNN, RNN, etc.

Section 3

Linguists' Role in Real-world NLP

Data Collection

In real-world scenarios, which is often more important: model vs. data?

Data Collection

In real-world scenarios, which is often more important: model vs. data?
Improvements by sophisticated models: 1% - 3%
Improvements by well-collected training data: 5% - 10%

Data Collection

In real-world scenarios, which is often more important: model vs. data?
Improvements by sophisticated models: 1% - 3%
Improvements by well-collected training data: 5% - 10%

A real case: the KBQA component needs to detect Chinese comparative sentences (e.g., '地球和火星哪个大？'). Given a huge number of Chinese queries on the Internet (e.g., queries in '百度知道'), how can you automatically collect comparative sentences from them?

Data Collection

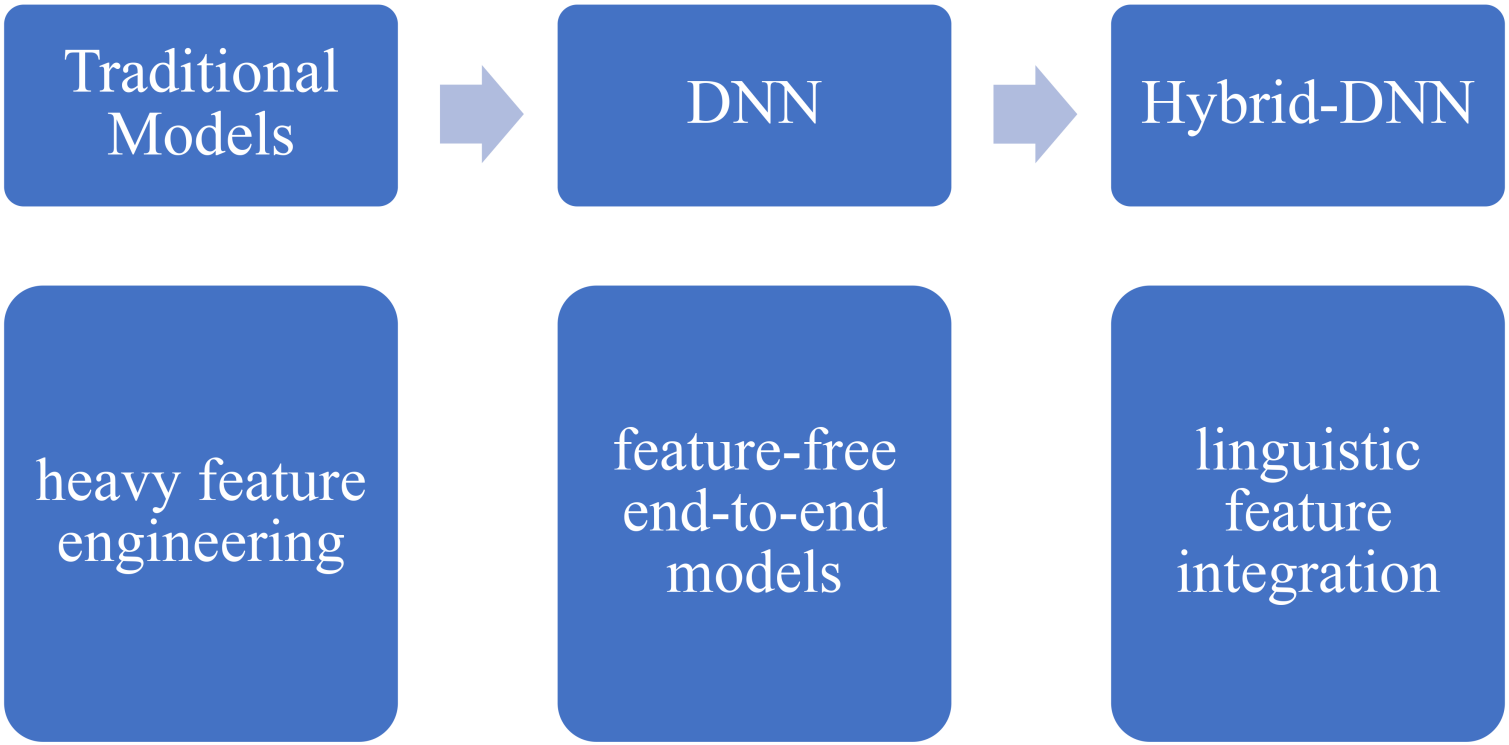
In real-world scenarios, which is often more important: model vs. data?
Improvements by sophisticated models: 1% - 3%
Improvements by well-collected training data: 5% - 10%

A real case: the KBQA component needs to detect Chinese comparative sentences (e.g., '地球和火星哪个大？'). Given a huge number of Chinese queries on the Internet (e.g., queries in '百度知道'), how can you automatically collect comparative sentences from them?

pat1 = r'(和 _con| 跟 _con).*_adj'
pat2 = r'(更 _adv| 相比 _v| 比 _prep).*_adj'
pat3 = r'_adj.* 还是 _con| 还是 _con.*_adj'

Feature Engineering

Trend of NLP models:



Feature engineering is specially important when data is scarce.

Feature Engineering

A real case: the sentiment analysis (i.e., given a sentence, decide its sentiment category) shared-task in CCL 2018. What features can you think of to help with this task?

Feature Engineering

A real case: the sentiment analysis (i.e., given a sentence, decide its sentiment category) shared-task in CCL 2018. What features can you think of to help with this task?

Some examples: sentiment words, N-grams, negation, figurative expressions, POS tags, WordNet synsets and hypernyms, dependency parsing structures.

We Need Linguists...A Lot of!

- We need linguists who have solid linguistics background.
- If you happen to know coding and some basic NLP knowledge, that's a plus.
- You can even write machine learning models? You are the popular kids.