Artificial Intelligence for Text Analytics

Question Answering and Dialogue Systems

Min-Yuh Day, Ph.D, Associate Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

1102AITA09
MBA, IM, NTPU (M5026) (Spring 2022)
Tue 2, 3, 4 (9:10-12:00) (B8F40)

https://meet.google.com/ paj-zhhj-myv
# Syllabus

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2022/02/22</td>
<td>Introduction to Artificial Intelligence for Text Analytics</td>
</tr>
<tr>
<td>2</td>
<td>2022/03/01</td>
<td>Foundations of Text Analytics: Natural Language Processing (NLP)</td>
</tr>
<tr>
<td>3</td>
<td>2022/03/08</td>
<td>Python for Natural Language Processing</td>
</tr>
<tr>
<td>4</td>
<td>2022/03/15</td>
<td>Natural Language Processing with Transformers</td>
</tr>
<tr>
<td>5</td>
<td>2022/03/22</td>
<td>Case Study on Artificial Intelligence for Text Analytics I</td>
</tr>
<tr>
<td>6</td>
<td>2022/03/29</td>
<td>Text Classification and Sentiment Analysis</td>
</tr>
</tbody>
</table>
# Syllabus

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>2022/04/05</td>
<td>Tomb-Sweeping Day (Holiday, No Classes)</td>
</tr>
<tr>
<td>8</td>
<td>2022/04/12</td>
<td>Midterm Project Report</td>
</tr>
<tr>
<td>9</td>
<td>2022/04/19</td>
<td>Multilingual Named Entity Recognition (NER), Text Similarity and Clustering</td>
</tr>
<tr>
<td>10</td>
<td>2022/04/26</td>
<td>Text Summarization and Topic Models</td>
</tr>
<tr>
<td>11</td>
<td>2022/05/03</td>
<td>Text Generation</td>
</tr>
<tr>
<td>12</td>
<td>2022/05/10</td>
<td>Case Study on Artificial Intelligence for Text Analytics II</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>2022/05/17</td>
<td>Question Answering and Dialogue Systems</td>
</tr>
<tr>
<td>14</td>
<td>2022/05/24</td>
<td>Deep Learning, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics</td>
</tr>
<tr>
<td>15</td>
<td>2022/05/31</td>
<td>Final Project Report I</td>
</tr>
<tr>
<td>16</td>
<td>2022/06/07</td>
<td>Final Project Report II</td>
</tr>
<tr>
<td>17</td>
<td>2022/06/14</td>
<td>Self-learning</td>
</tr>
<tr>
<td>18</td>
<td>2022/06/21</td>
<td>Self-learning</td>
</tr>
</tbody>
</table>
Question Answering and Dialogue Systems
Outline

• Question Answering
• Dialogue Systems
• Task Oriented Dialogue System
Question Answering

Inputs

Question
Which name is also used to describe the Amazon rainforest in English?

Context
The Amazon rainforest, also known in English as Amazonia or the Amazon Jungle

Output

Answer
Amazonia

https://huggingface.co/tasks/question-answering
Question Answering demo

Question Answering

Where do I live?

Compute

Context

My name is Michael and I live in Taipei.

Computation time on cpu: 0.0492 s

Taipei 0.920

https://huggingface.co/tasks/question-answering
Question Answering

!pip install transformers
from transformers import pipeline
qamodel = pipeline("question-answering")
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
qamodel(question = question, context = context)

{'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
from transformers import pipeline
qamodel = pipeline("question-answering", model = 'deepset/roberta-base-squad2')
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
output = qamodel(question = question, context = context)
print(output['answer'])

Taipei
IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-9 RITE

Department of Information Management
Tamkang University, Taiwan

Min-Yuh Day
myday@mail.tku.edu.tw

Chun Tu

NTCIR-9 Workshop, December 6-9, 2011, Tokyo, Japan
IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-10 RITE-2

Department of Information Management, Tamkang University, Taiwan

Min-Yuh Day
Chun Tu
Hou-Cheng Vong
Shih-Wei Wu
Shih-Jhen Huang
IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-11 RITE-VAL

Tamkang University

2014

Yu-Hsuan Tai
Cheng-Chia Tsai

Yu-An Lin
Shang-Yu Wu
Yu-Hsuan Tai
Cheng-Chia Tsai

Huai-Wen Hsu
Yu-An Lin
Shang-Yu Wu
Yu-Hsuan Tai
Cheng-Chia Tsai

NTCIR-11 Conference, December 8-12, 2014, Tokyo, Japan
IMTKU Question Answering System for World History Exams at NTCIR-12 QA Lab2

Department of Information Management
Tamkang University, Taiwan

myday@mail.tku.edu.tw

NTCIR-12 Conference, June 7-10, 2016, Tokyo, Japan
IMTKU Question Answering System for World History Exams at NTCIR-13 QALab-3

Department of Information Management
Tamkang University, Taiwan

Min-Yuh Day
Chao-Yu Chen

Wanchu Huang
Shi-Ya Zheng
I-Hsuan Huang
Tz-Rung Chen
Min-Chun Kuo
Yue-Da Lin
Yi-Jing Lin

myday@mail.tku.edu.tw
NTCIR-13 Conference, December 5-8, 2017, Tokyo, Japan
IMTKU Emotional Dialogue System for Short Text Conversation at **NTCIR-14 STC-3 (CECG) Task**

Department of Information Management
Tamkang University, Taiwan

Min-Yuh Day  Chi-Sheng Hung  Yi-Jun Xie  Jhih-Yi Chen  Yu-Ling Kuo  Jian-Ting Lin

**myday@mail.tku.edu.tw**

NTCIR-14 Conference, June 10-13, 2019, Tokyo, Japan
IMTKU Multi-Turn Dialogue System Evaluation at the NTCIR-15 DialEval-1
Dialogue Quality and Nugget Detection

1 Zeals Co., Ltd. Tokyo, Japan
2 Information Management, Tamkang University, Taiwan
3 Information Management, National Taipei University, Taiwan
### 2020 NTCIR-15 Dialogue Evaluation (DialEval-1) Task
Dialogue Quality (DQ) and Nugget Detection (ND)

#### Chinese Dialogue Quality (S-score) Results (Zeng et al., 2020)

<table>
<thead>
<tr>
<th>Run</th>
<th>Mean RSNOD</th>
<th>Run</th>
<th>Mean NMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMTKU-run2</td>
<td>0.1918</td>
<td>IMTKU-run2</td>
<td>0.1254</td>
</tr>
<tr>
<td>IMTKU-run1</td>
<td>0.1964</td>
<td>IMTKU-run0</td>
<td>0.1284</td>
</tr>
<tr>
<td>IMTKU-run0</td>
<td>0.1977</td>
<td>IMTKU-run1</td>
<td>0.1290</td>
</tr>
<tr>
<td>TUA1-run2</td>
<td>0.2024</td>
<td>TUA1-run2</td>
<td>0.1310</td>
</tr>
<tr>
<td>TUA1-run0</td>
<td>0.2053</td>
<td>TUA1-run0</td>
<td>0.1322</td>
</tr>
<tr>
<td>NKUST-run1</td>
<td>0.2057</td>
<td>NKUST-run1</td>
<td>0.1363</td>
</tr>
<tr>
<td>BL-lstm</td>
<td>0.2088</td>
<td>TUA1-run1</td>
<td>0.1397</td>
</tr>
<tr>
<td>WUST-run0</td>
<td>0.2131</td>
<td>BL-popularity</td>
<td>0.1442</td>
</tr>
<tr>
<td>RSLNV-run0</td>
<td>0.2141</td>
<td>BL-lstm</td>
<td>0.1455</td>
</tr>
<tr>
<td>BL-popularity</td>
<td>0.2288</td>
<td>RSLNV-run0</td>
<td>0.1483</td>
</tr>
<tr>
<td>TUA1-run1</td>
<td>0.2302</td>
<td>WUST-run0</td>
<td>0.1540</td>
</tr>
<tr>
<td>NKUST-run0</td>
<td>0.2653</td>
<td>NKUST-run0</td>
<td>0.2289</td>
</tr>
<tr>
<td>BL-uniform</td>
<td>0.2811</td>
<td>BL-uniform</td>
<td>0.2497</td>
</tr>
</tbody>
</table>

Transformer-based Models Selection

- BERT
- RoBERTa
- XLM-RoBERTa

Pre-trained Models

Fine-tuning Techniques

- FinNum-2
- DialEval-1
  - Discriminative Fine-tuning
  - One-cycle Policy
  - Optimization
  - Tokenization Tricks

Transfer Learning

Short Text Conversation Task (STC-3) Chinese Emotional Conversation Generation (CECG) Subtask

Source: http://coai.cs.tsinghua.edu.cn/hml/challenge.html
### NTCIR Short Text Conversation
**STC-1, STC-2, STC-3**

<table>
<thead>
<tr>
<th></th>
<th>Japanese</th>
<th>Chinese</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NTCIR-12 STC-1</strong></td>
<td>Twitter, Retrieval</td>
<td>Weibo, Retrieval</td>
<td></td>
</tr>
<tr>
<td>22 active participants</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NTCIR-13 STC-2</strong></td>
<td>Yahoo! News, Retrieval+ Generation</td>
<td>Weibo, Retrieval+ Generation</td>
<td></td>
</tr>
<tr>
<td>27 active participants</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NTCIR-14 STC-3</strong></td>
<td></td>
<td></td>
<td>Multi-turn, task-oriented (helpdesk)</td>
</tr>
</tbody>
</table>

- **Chinese Emotional Conversation Generation (CECG) subtask**
- **Dialogue Quality (DQ) and Nugget Detection (ND) subtasks**
- **Weibo, Generation for given emotion categories**
- **Weibo+English translations, distribution estimation for subjective annotations**

Source: [https://waseda.app.box.com/v/STC3atNTCIR-14](https://waseda.app.box.com/v/STC3atNTCIR-14)
IMTKU System Architecture for NTCIR-13 QALab-3

Question (XML)

Question Analysis

Complex Essay
Simple Essay
True-or-False
Factoid
Slot-Filling
Unique

Document Retrieval

Stanford CoreNLP

JA&EN Translator

Wikipedia

Answer Extraction

Answer Generation

Word Embedding Wiki Word2Vec

Answer (XML)

NTCIR-13 Conference, December 5-8, 2017, Tokyo, Japan
System Architecture of Intelligent Dialogue and Question Answering System

- User Question Input
  - Dialogue Intention Detection
    - Dialogue Intention Detection
    - AIML Dialogue Engine
    - Real Time Dialogue API
    - System Response Generator
  - RNN
    - LSTM
    - GRU
  - AIML KB
    - Cloud Resource
  - System Response Generator

- Question Analysis
  - Question Analysis
  - Document Retrieval
  - Answer Extraction
  - Answer Generation
  - Answer Validation
  - Deep Learning
    - TensorFlow
  - Python
    - NLTK
  - Dialogue KB
    - IR
  - Deep Learning
IMTKU Emotional Dialogue
System Architecture

1. Retrieval-Based Model
2. Generation-Based Model
3. Emotion Classification Model
4. Response Ranking

NTCIR-14 Conference, June 10-13, 2019, Tokyo, Japan
The system architecture of IMTKU retrieval-based model for NTCIR-14 STC-3

Retrieval-Based Model

- Post
- Word Segmentation
- Keyword Boolean Query
- Corpus
- Building Index
- Solr Matching
- Distinct Result Data
- Emotion Matching
- Emotion Classification
- Word2Vec Similarity Ranking
- Retrieval-Based Response
The system architecture of IMTKU generation-based model for NTCIR-14 STC-3

**Generation-Based Model**

- Training Data
- Building Word Index
- Word Embedding
- Training Data Seq2seq model

**Post**

- Word Segmentation
- Short Text Emotion Classifier
- Trained Model

- Emotion Matching
- Word2Vec Similarity Ranking

**Generation-Based Response**
The system architecture of IMTKU emotion classification model for NTCIR-14 STC-3

Emotion Classification Model

Corpus → Emotion Classification → Training Dataset → MLP LSTM BiLSTM

_corpus

Training Dataset

Emotion Classification Model

Testing Dataset → Emotion Classification Model → Emotion Prediction

_ntcir-14 conference, june 10-13, 2019, tokyo, japan

27
The system architecture of IMTKU Response Ranking for NTCIR-14 STC-3

Response Ranking

1. STC3 Corpus
2. Chinese Segmentation using Jieba
3. Stop Words Removal
4. Word2Vec
5. 1.2 million data (300 dimensions)
6. Vector of Corpus
AI Humanoid
Robo-Advisor
AI Humanoid Robo-Advisor for Multi-channel Conversational Commerce

AI Portfolio Asset Allocation

AI Conversation Dialog System

Multichannel Platforms
- Web
- LINE
- Facebook
- Humanoid Robot
System Architecture of AI Humanoid Robo-Advisor
Conversational Model
(LINE, FB Messenger)
Conversational Robo-Advisor
Multichannel UI/UX
Robots

ALPHA 2

ZENBO
Question Answering
When did Marie Curie win her first Nobel Prize?

1903

With Henri Becquerel and her husband, Pierre Curie, Marie Curie was awarded the 1903 Nobel Prize for Physics. She was the sole winner of the 1911 Nobel Prize for Chemistry. She was the first woman to win a Nobel Prize and the only woman to win the award in two different fields.

https://www.britannica.com/Science/Physics/Physicists

Marie Curie | Biography & Facts | Britannica
The Retriever-Reader Architecture for Modern QA Systems

When did Marie Curie win her first Nobel Prize?

1903

Unstructured documents (Wikipedia, internet, ...)

Retriever

Document postprocessing

Q: When did Marie Curie win her first Nobel Prize?

Reader

Relevant documents

A: 1903

Answer postprocessing

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
Question Answering System (QAS)

User's Questions
  \--- Question Processing
     \-- Question Classification
  \--- Document Processing
     \-- Passage Retrieval
  \--- Answer Processing
     \-- Answer Presentation
  \--- Answer

Fine-tuning BERT on Question Answering (QA)

(c) Question Answering Tasks:
SQuAD v1.1

The span classification head for QA tasks

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
Multiple question-context pairs

Why is the camera of poor quality?  Item like the picture, fast deliver 3 days well packed, good quality for the price. The camera is decent (as phone cameras go). There is no flash though...

[CLS] [SEP] [SEP]

Stride

Why is the camera of poor quality?  Item like the picture, fast deliver 3 days well packed, good quality for the price. The camera is decent (as phone cameras go). There is no flash though...

[CLS] [SEP] [SEP]

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O’Reilly Media.
Dense Passage Retrieval (DPR)

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
Going Beyond Extractive QA Retrieval-Augmented Generation (RAG)

The RAG architecture for fine-tuning a retriever and generator end-to-end

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
The QA Hierarchy of Needs

- **Generation**
  - T5, RAG, GPT-3

- **Extraction**
  - Zero-shot NER, domain adaption

- **Search**
  - TF-IDF, BM25, DPR

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
Transformer (Attention is All You Need)
(Vaswani et al., 2017)

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT (Bidirectional Encoder Representations from Transformers)

Overall pre-training and fine-tuning procedures for BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT (Bidirectional Encoder Representations from Transformers)

BERT input representation

T5

Text-to-Text Transfer Transformer

"translate English to German: That is good."

"cola sentence: The course is jumping well."

"sts1 sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."

"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi..."

"Das ist gut."

"not acceptable"

"3.8"

"six people hospitalized after a storm in attala county."

Fine-tuning BERT on Different Tasks

(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Fine-tuning BERT on Question Answering (QA)

Fine-tuning BERT on Dialogue

Intent Detection (ID; Classification)

(b) Single Sentence Classification Tasks: SST-2, CoLA

Fine-tuning BERT on Dialogue Slot Filling (SF)

Question Answering (QA)

SQuAD

Stanford Question Answering Dataset
SQuAD 2.0
The Stanford Question Answering Dataset

What is SQuAD?

SQuAD (Stanford Question Answering Dataset) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.

SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Human Performance Stanford University (Rajpurkar &amp; Jia et al. ‘18)</td>
<td>86.831</td>
<td>89.452</td>
</tr>
<tr>
<td>1</td>
<td>SA-Net on Albert (ensemble) QIANXIN</td>
<td>90.724</td>
<td>93.011</td>
</tr>
<tr>
<td>2</td>
<td>SA-Net-V2 (ensemble) QIANXIN</td>
<td>90.679</td>
<td>92.948</td>
</tr>
<tr>
<td>?</td>
<td>Retro-Reader (ensemble)</td>
<td>90.578</td>
<td>92.978</td>
</tr>
</tbody>
</table>

https://rajpurkar.github.io/SQuAD-explorer/
SQuAD: 100,000+ Questions for Machine Comprehension of Text

Pranav Rajpurkar and Jian Zhang and Konstantin Lopyrev and Percy Liang
{pranavsr,zjian,klopyrev,pliang}@cs.stanford.edu
Computer Science Department
Stanford University

Abstract

We present the Stanford Question Answering Dataset (SQuAD), a new reading comprehension dataset consisting of 100,000+ questions posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage. We analyze the dataset to understand the types of reasoning required to answer the questions, learning heavily on dependency and constituency trees. We build a strong logistic regression model, which achieves an F1 score of 51.0%, a significant improvement over a simple baseline (20%). However, human performance (86.8%) is much higher, indicating that the dataset presents a good challenge problem for future research. The dataset is freely available at https://stanford-qa.com.

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall? gravity

What is another main form of precipitation besides drizzle, rain, sleet and hail? graupel

Where do water droplets collide with ice crystals to form precipitation? within a cloud

Figure 1: Question-answer pairs for a sample passage in the
SQuAD (Question Answering)

Q: What causes precipitation to fall?

Precipitation

From Wikipedia, the free encyclopedia

For other uses, see Precipitation (disambiguation).

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity from clouds. The main forms of precipitation include drizzle, rain, sleet, snow, ice pellets, graupel and hail. Precipitation occurs when a portion of the atmosphere becomes saturated with water vapor (reaching 100% relative humidity), so that the water condenses and "precipitates". Thus, fog and mist are not precipitation but suspensions, because the water vapor does not condense sufficiently to precipitate. Two processes, possibly acting together, can lead to air becoming saturated: cooling the air or adding water vapor to the air. Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers."
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called “showers”.

Q: What causes precipitation to fall?
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called “showers”.

Q: What causes precipitation to fall?
A: gravity
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called “showers”.

Q: What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

A: graupel
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called “showers”.

Q: Where do water droplets collide with ice crystals to form precipitation?

A: **within a cloud**
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called “showers”.

Q: What causes precipitation to fall?
A: gravity

Q: What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
A: graupel

Q: Where do water droplets collide with ice crystals to form precipitation?
A: within a cloud
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Question Answering

```python
# from transformers import pipeline
# qamodel = pipeline("question-answering", model='deepset/roberta-base-squad2')
# question = "Where do water droplets collide with ice crystals to form precipitation?"
# context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity within a cloud"
# output = qamodel(question = question, context = context)
# print(output['answer'])

gravity
```

```python
[12]
from transformers import pipeline
qamodel = pipeline("question-answering", model='deepset/roberta-base-squad2')
question = "What causes precipitation to fall?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity"
output = qamodel(question = question, context = context)
print(output['answer'])
```

```python
[13]
from transformers import pipeline
qamodel = pipeline("question-answering", model='deepset/roberta-base-squad2')
question = "What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity"
output = qamodel(question = question, context = context)
print(output['answer'])
```

https://tinyurl.com/aintpuppython101
from transformers import pipeline
qamodel = pipeline("question-answering", model = 'deepset/roberta-base-squad2')
question = "What causes precipitation to fall?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".""
output = qamodel(question = question, context = context)
print(output['answer'])
Question Answering on SQuAD 2.0

SQuAD 2.0 benchmark (Papers with Code)

https://paperswithcode.com/sota/question-answering-on-squad20

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
Neural Image Captioning (NIC)  
image-to-text description generation

A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Visual Question Answering

Neural caption generation is employed to aid answer prediction.

Dialogue Systems
Conversational Commerce
Chatbot
Dialogue System
Intelligent Agent
Dialogue Subtasks

Source: https://paperswithcode.com/area/natural-language-processing/dialogue
Chatbots: Evolution of UI/UX

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>80s</th>
<th>90s</th>
<th>00s</th>
<th>10s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>PC</td>
<td>Web</td>
<td>Smartphone</td>
<td>Messaging</td>
</tr>
<tr>
<td>Examples</td>
<td>Desktop (DOS, Windows, Mac OS)</td>
<td>Browser (Mosaic, Explorer, Chrome)</td>
<td>Mobile OS (iOS, Android)</td>
<td>Messaging Apps (WhatsApp, Messenger, Slack)</td>
</tr>
<tr>
<td>Applications</td>
<td>Clients (Excel, PPT, Lotus)</td>
<td>Website (Yahoo, Amazon)</td>
<td>Apps (Angry Birds, Instagram)</td>
<td>Bots (Weather, Travel)</td>
</tr>
<tr>
<td>UI/UX</td>
<td>Native Screens</td>
<td>Web Pages</td>
<td>Native Mobile Screens</td>
<td>Message</td>
</tr>
<tr>
<td>S/w Dev</td>
<td>Client-side</td>
<td>Server-side</td>
<td>Client-side</td>
<td>Server-side</td>
</tr>
</tbody>
</table>

Source: https://bbvaopen4u.com/en/actualidad/want-know-how-build-conversational-chatbot-here-are-some-tools
From E-Commerce to Conversational Commerce: Chatbots and Virtual Assistants
Conversational Commerce: eBay AI Chatbots

Hotel Chatbot

**Intent Detection**

**Intents**
An intent performs an action in response to natural language user input.

**Utterances**
Spoken or typed phrases that invoke your intent.

**Slots**
Slots are input data required to fulfill the intent.

**Slot Filling**

**Fulfillment**
Fulfillment mechanism for your intent.

Source: https://sdtimes.com/amazon/guest-view-capitalize-amazon-lex-available-general-public/
Fine-tuning BERT on Dialogue

Intent Detection (ID; Classification)

(b) Single Sentence Classification Tasks: SST-2, CoLA

Fine-tuning BERT on Dialogue Slot Filling (SF)

H&M’s Chatbot on Kik

Source: http://www.guided-selling.org/from-e-commerce-to-conversational-commerce/
Uber’s Chatbot on Facebook’s Messenger

Uber’s chatbot on Facebook’s messenger
- one main benefit: it loads much faster than the Uber app

Source: http://www.guided-selling.org/from-e-commerce-to-conversational-commerce/
Chatbot

Source: https://www.mdsdecoded.com/blog/the-rise-of-chatbots/
Overall Architecture of Intelligent Chatbot

Can machines think? (Alan Turing, 1950)

Chatbot

“online human-computer dialog system with natural language.”

Chatbot Conversation Framework

<table>
<thead>
<tr>
<th>Conversations</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Domain</td>
<td>Impossible</td>
</tr>
<tr>
<td>Closed Domain</td>
<td>Rules-Based [Easiest]</td>
</tr>
<tr>
<td></td>
<td>Smart Machine [Hard]</td>
</tr>
<tr>
<td></td>
<td>Generative-Based</td>
</tr>
</tbody>
</table>

Source: https://chatbotslife.com/ultimate-guide-to-leveraging-nlp-machine-learning-for-you-chatbot-531ff2dd870c
Chatbots

Bot Maturity Model

Customers want to have simpler means to interact with businesses and get faster response to a question or complaint.

Bot Life Cycle and Platform Ecosystem
The Bot Lifecycle

Source: https://chatbotsmagazine.com/the-bot-lifecycle-1ff357430db7
The bot platform ecosystem
and the emerging giants

Nearly every large software company has announced some sort of bot strategy in the last year. Here’s a look at a handful of leading platforms that developers might use to send messages, interpret natural language, and deploy bots, with the emerging bot-ecosystem giants highlighted.

General AI agents with platforms
Developer access available now or announced

Source: https://www.oreilly.com/ideas/infographic-the-bot-platform-ecosystem
Bot frameworks and deployment platforms

- Wit.ai
  - Facebook
- BotKit
  - Howdy
- Chatfuel
- Automat
  - Bot Framework
  - Microsoft
- Api.ai
  - Google
- Pandorabots
- MindMeld
- Gupshup
- Sequel

Source: https://www.oreilly.com/ideas/infographic-the-bot-platform-ecosystem
How to Build Chatbots

Chatbot Frameworks and AI Services

• Bot Frameworks
  • Botkit
  • Microsoft Bot Framework
  • Rasa NLU

• AI Services
  • Wit.ai
  • api.ai
  • LUIS.ai
  • IBM Watson

# Chatbot Frameworks

## Comparison Table of Most Prominent Bot Frameworks

<table>
<thead>
<tr>
<th>Feature</th>
<th>Botkit</th>
<th>Microsoft Bot Framework</th>
<th>RASA NLU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Built-in Integration with messaging platforms</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td><strong>NLP support</strong></td>
<td>✗ (but possible to integrate with middlewares)</td>
<td>✗ (but have close bonds with LUIS.ai)</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Out-of-box bots ready to be deployed</strong></td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td><strong>Programming Language</strong></td>
<td>JavaScript (Node)</td>
<td>JavaScript (Node), C#</td>
<td>Python</td>
</tr>
</tbody>
</table>

# Comparison of Most Prominent AI Services

<table>
<thead>
<tr>
<th></th>
<th>wit.ai</th>
<th>api.ai</th>
<th>LUIS.ai</th>
<th>IBM Watson</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Free of charge</strong></td>
<td>✔️</td>
<td>✔️ but has paid enterprise version</td>
<td>✔️</td>
<td>30 days trial then priced for enterprise use</td>
</tr>
<tr>
<td><strong>Text and Speech processing</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️ with use of Cortana</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Machine Learning Modeling</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Support for Intents, Entities, Actions</strong></td>
<td>✔️ Intents used as trait entities, actions are combined operations</td>
<td>✔️ Intents is the main prediction mechanism, Domains of entities, intents and actions</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Pre-build entities for easy parsing of numbers, temperature, date, etc.</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Integration to messaging platforms</strong></td>
<td>✔️ web service API</td>
<td>✔️ also has facility for deploying to heroku. Paid environment</td>
<td>✔️ integrated to Azure</td>
<td>✔️ possible via API</td>
</tr>
<tr>
<td><strong>Support of SDKs</strong></td>
<td>✔️ includes SDKs for Python, Node.js, Rust, C, Ruby, iOS, Android, Windows Phone</td>
<td>✔️ C#, Xamarin, Python, Node.js, iOS, Android, Windows Phone</td>
<td>✔️ enables building with Web Service API, Microsoft Bot Framework integration</td>
<td>Proprietary language “AlchemyLanguage”</td>
</tr>
</tbody>
</table>

Task-Oriented Dialogue System
Task-Oriented Dialogue System
(Deriu et al., 2021)

Task-Oriented Dialogue Systems
(Zhang et al., 2020)

Source: Zhang, Zheng, Ryuichi Takanobu, Qi Zhu, Minlie Huang, and Xiaoyan Zhu (2020).
“Recent advances and challenges in task-oriented dialog systems.” Science China Technological Sciences (2020): 1-17.
Dialog State Tracker (DST)

## Dialogue Acts

(Youn et al., 2010)

<table>
<thead>
<tr>
<th>Dialogue act</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hello (a = x, b = y, \ldots)</td>
<td>Open a dialogue and give info (a = x, b = y, \ldots)</td>
</tr>
<tr>
<td>inform (a = x, b = y, \ldots)</td>
<td>Give information (a = x, b = y, \ldots)</td>
</tr>
<tr>
<td>request ((a, b = x, \ldots))</td>
<td>Request value for (a) given (b = x, \ldots)</td>
</tr>
<tr>
<td>reqalts (a = x, \ldots)</td>
<td>Request alternative with (a = x, \ldots)</td>
</tr>
<tr>
<td>confirm ((a = x, b = y, \ldots))</td>
<td>Explicitly confirm (a = x, b = y, \ldots)</td>
</tr>
<tr>
<td>confreq ((a = x, \ldots, d))</td>
<td>Implicitly confirm (a = x, \ldots) and request value of (d)</td>
</tr>
<tr>
<td>select ((a = x, a = y))</td>
<td>Select either (a = x) or (a = y)</td>
</tr>
<tr>
<td>affirm ((a = x, b = y))</td>
<td>Affirm and give further info (a = x, b = y, \ldots)</td>
</tr>
<tr>
<td>negate ((a = x))</td>
<td>Negate and give corrected value (a = x)</td>
</tr>
<tr>
<td>deny ((a = x))</td>
<td>Deny that (a = x)</td>
</tr>
<tr>
<td>bye()</td>
<td>Close a dialogue</td>
</tr>
</tbody>
</table>

# Sample Dialogue Acts

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Dialogue Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>U: Hi, I am looking for somewhere to eat</td>
<td>hello(task = find, type = restaurant)</td>
</tr>
<tr>
<td>S: You are looking for a restaurant. What type of food?</td>
<td>confreq(type = restaurant, food)</td>
</tr>
<tr>
<td>U: I’d like an Italian somewhere near the museum.</td>
<td>inform(food = Italian, near = museum)</td>
</tr>
<tr>
<td>S: Roma is a nice Italian restaurant near the museum.</td>
<td>inform(name = “Roma”, type = restaurant, food = Italian, near = museum)</td>
</tr>
<tr>
<td>U: Is it reasonably priced?</td>
<td>confirm(pricerange = moderate)</td>
</tr>
<tr>
<td>S: Yes, Roma is in the moderate price range.</td>
<td>affirm(name = “Roma”, pricerange = moderate)</td>
</tr>
<tr>
<td>U: What is the phone number?</td>
<td>request(phone)</td>
</tr>
<tr>
<td>S: The number of Roma is 385456.</td>
<td>inform(name = “Roma”, phone = “385456”)</td>
</tr>
<tr>
<td>U: Ok, thank you goodbye.</td>
<td>bye()</td>
</tr>
</tbody>
</table>

Dialogue on Airline Travel Information System (ATIS)
The ATIS (Airline Travel Information System) Dataset

https://www.kaggle.com/siddhadev/atis-dataset-from-ms-cntk

<table>
<thead>
<tr>
<th>Sentence</th>
<th>what</th>
<th>flights</th>
<th>leave</th>
<th>from</th>
<th>phoenix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slots</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>B-fromloc</td>
</tr>
<tr>
<td>Intent</td>
<td>atis_flight</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Training samples: 4978  
Testing samples: 893  
Vocab size: 943  
Slot count: 129  
Intent count: 26

Intent Detection on ATIS
State-of-the-art

<table>
<thead>
<tr>
<th>RANK</th>
<th>METHOD</th>
<th>ACCURACY</th>
<th>PAPER TITLE</th>
<th>YEAR</th>
<th>PAPER</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SF-ID</td>
<td>0.9776</td>
<td>A Novel Bi-directional Interlaced Model for Joint Intent Detection and Slot Filling</td>
<td>2019</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Capsule-NLU</td>
<td>0.950</td>
<td>Joint Slot Filling and Intent Detection via Capsule Neural Networks</td>
<td>2018</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: [paperswithcode.com/sota/intent-detection-on-atis](https://paperswithcode.com/sota/intent-detection-on-atis)
Slot Filling on ATIS

State-of-the-art

Slot Filling on ATIS

<table>
<thead>
<tr>
<th>RANK</th>
<th>METHOD</th>
<th>F1</th>
<th>PAPER TITLE</th>
<th>YEAR</th>
<th>PAPER</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SF-ID</td>
<td>0.958</td>
<td>A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling</td>
<td>2019</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Capsule-NLU</td>
<td>0.952</td>
<td>Joint Slot Filling and Intent Detection via Capsule Neural Networks</td>
<td>2018</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: [paperswithcode.com/sota/slot-filling-on-atis](https://paperswithcode.com/sota/slot-filling-on-atis)
A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling

PARAdigm for Dialog System Evaluation
PARADISE Framework (Walker et al. 1997)

Interaction Quality procedure
(Schmitt and Ultes, 2015)

Datasets for task-oriented dialogue systems

<table>
<thead>
<tr>
<th>Name</th>
<th>Topics</th>
<th># dialogues</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTC1</td>
<td>Bus schedules</td>
<td>15,000</td>
<td>(Williams et al. 2013)</td>
</tr>
<tr>
<td>DSTC2</td>
<td>Restaurants</td>
<td>3000</td>
<td>(Henderson et al. 2014)</td>
</tr>
<tr>
<td>DSTC3</td>
<td>Tourist information</td>
<td>2265</td>
<td>(Henderson et al. 2013a)</td>
</tr>
<tr>
<td>DSTC4 &amp; DSTC5</td>
<td>Tourist information</td>
<td>35</td>
<td>(Kim et al. 2016)</td>
</tr>
<tr>
<td>DSTC6</td>
<td>Restaurant reservation</td>
<td>–</td>
<td>(Perez et al. 2017)</td>
</tr>
<tr>
<td>DSTC7 (Flex Data)</td>
<td>Student guiding</td>
<td>500</td>
<td>(Gunasekara et al. 2019)</td>
</tr>
<tr>
<td>DSTC8 (MetaLWOz)</td>
<td>47 domains</td>
<td>37,884</td>
<td>(Lee et al. 2019)</td>
</tr>
<tr>
<td>DSTC8 (Schema-Guided)</td>
<td>20 domains</td>
<td>22,825</td>
<td>(Rastogi et al. 2019)</td>
</tr>
<tr>
<td>MultiWOZ</td>
<td>Tourist information</td>
<td>10,438</td>
<td>(Budzianowski et al. 2018)</td>
</tr>
<tr>
<td>Taskmaster-1</td>
<td>6 domains</td>
<td>13,215</td>
<td>(Byrne et al. 2019)</td>
</tr>
<tr>
<td>MultiDoGo</td>
<td>6 domains</td>
<td>86,698</td>
<td>(Peskov et al. 2019)</td>
</tr>
</tbody>
</table>

Restaurants Dialogue Datasets

• MIT Restaurant Corpus
  • https://groups.csail.mit.edu/sls/downloads/restaurant/

• CamRest676
  (Cambridge restaurant dialogue domain dataset)
  • https://www.repository.cam.ac.uk/handle/1810/260970

• DSTC2 (Dialog State Tracking Challenge 2 & 3)
  • http://camdial.org/~mh521/dstc/
## CrossWOZ:
A Large-Scale Chinese Cross-Domain Task-Oriented Dialogue Dataset

<table>
<thead>
<tr>
<th>MultiWOZ</th>
<th>CrossWOZ</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>usr:</strong> I’m looking for a college type attraction. ...</td>
<td><strong>usr:</strong> 你好，可以帮我推荐一个评分是4.5分以上的景点吗？</td>
</tr>
<tr>
<td><strong>usr:</strong> I would like to visit in town centre please. ...</td>
<td><strong>Hello, could you recommend an attraction with a rating of 4.5 or higher?</strong></td>
</tr>
<tr>
<td><strong>usr:</strong> Can you find an Indian restaurant for me that is also in the town centre?</td>
<td><strong>sys:</strong> 天安门城楼，簋街小吃和北京欢乐谷都是很不错的地方呢。**</td>
</tr>
<tr>
<td><strong>usr:</strong> I want a hotel in San Diego and I want to check out on Thursday next week. ...</td>
<td><strong>usr:</strong> I like Beijing Happy Valley. What hotels are around this attraction?</td>
</tr>
<tr>
<td><strong>usr:</strong> I need a one way flight to go there.</td>
<td><strong>sys:</strong> 那可多了，有A酒店, B酒店, C酒店。**</td>
</tr>
<tr>
<td><strong>sys:</strong> There are many, such as hotel A, hotel B, and hotel C.</td>
<td><strong>usr:</strong> 太好了，我正打算在景点附近找个酒店住宿呢，知道哪家评分是4分以上，提供叫醒服务的不？**</td>
</tr>
<tr>
<td><strong>usr:</strong> Great! I am planning to find a hotel to stay near the attraction. Which one has a rating of 4 or higher and offers wake-up call service?</td>
<td><strong>sys:</strong> Great! I am planning to find a hotel to stay near the attraction. Which one has a rating of 4 or higher and offers wake-up call service?</td>
</tr>
</tbody>
</table>

CrossWOZ: A Large-Scale Chinese Cross-Domain Task-Oriented Dialogue Dataset

<table>
<thead>
<tr>
<th>Type</th>
<th>Single-domain goal</th>
<th>Multi-domain goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>DSTC2</td>
<td>WOZ 2.0</td>
</tr>
<tr>
<td>Language</td>
<td>EN</td>
<td>EN</td>
</tr>
<tr>
<td>Speakers</td>
<td>H2M</td>
<td>H2H</td>
</tr>
<tr>
<td># Domains</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td># Dialogues</td>
<td>1,612</td>
<td>600</td>
</tr>
<tr>
<td># Turns</td>
<td>23,354</td>
<td>4,472</td>
</tr>
<tr>
<td>Avg. domains</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Avg. turns</td>
<td>14.5</td>
<td>7.5</td>
</tr>
<tr>
<td># Slots</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td># Values</td>
<td>212</td>
<td>99</td>
</tr>
</tbody>
</table>

Task-Oriented Dialogue

Initial user state (=user goal)

id=1 (Attraction): fee=free,
name=?, nearby hotels=?

id=2 (Hotel): name=near (id=1),
wake-up call=yes, rating=?

id=3 (Taxi): from=(id=1), to=(id=2),
car type=? plate number=?

Final user state

id=1 (Attraction): name=Tiananmen Square,
fee=free, nearby hotels=[Beijing Capital Hotel, Guidu Hotel Beijing]
id=2 (Hotel): name=Beijing Capital Hotel,
wake-up call=yes, rating=4.6

id=3 (Taxi): from=Tiananmen Square,
to=Beijing Capital Hotel,
car type=##CX, plate number=##CP

## An example dialog from the test set for MultiWOZ (en→zh) sub-task

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Utterance</th>
<th>Dialog State Update</th>
</tr>
</thead>
</table>
| User    | Hello! I am looking for a local guesthouse in the centre. | hotel: {area: centre, type: guesthouse}  
          | 你好！我在市中心找一家本地宾馆。 | 旅馆: {区域: 中心, 类型: 宾馆} |
| System  | OK. I am glad to recommend Alexander Bed and Breakfast to you. | 旅馆: {名称: 亚历山大住宿加早餐旅馆}  
          | 好的 - 这边很高兴向您推荐亚历山大住宿加早餐旅馆。 |  |
| User    | Where is it? | hotel: {name: alexander bed and breakfast}  
          | 它在哪里？ | 餐厅: {食物: 欧洲的, 区域: 中心} |
| System  | It is at 56 Saint Barnabas Road. | 旅馆: {名称: 亚历山大住宿加早餐旅馆}  
          | 圣巴纳巴斯路56号。 | |
| User    | I also wish to have a meal in a local European restaurant in the centre. | restaurant: {food: european, area: centre}  
          | 我还想在市中心的一家本地欧洲餐厅吃饭。 |  |
| System  | You can choose Eraina. |  | 您可以选择伊莱娜。 |
| User    | Please give me its address. | restaurant: {name: eraina}  
          | 那请给我它的地址。 | 餐厅: {名称: 伊莱娜} |
| System  | It is in Free School Lane City Centre. |  |  |
|         | 市中心自由校园巷。 | |
| User    | Ok. I’ll go there. I need to book a taxi from Alexander Bed and Breakfast to Eraina after 07:00. | taxi: {leaveAt: 07:00, destination: eraina,  
          | 好，那我去那里，我还要预订一辆7:00时从亚历山大住 | departure: alexander bed and breakfast}  
          |          | 出租车: {出发时间: 07:00, 目的地: 伊莱娜,  
          | 进住宿加早餐旅馆到伊莱娜的出租车。 | 出发地: 亚历山大住宿加早餐旅馆} |
| System  | Well. I find a yellow Skoda. |  |  |
|         | 好的。我找到了一辆黄色的斯柯达。 | |
| User    | How about its phone number? |  | No update |
| System  | It is 7851967525, 7851967525. |  |  |
| User    | Thank you for your help. Bye! |  | No update |
| System  | A pleasure. Bye bye! |  |  |
|         | 谢谢你帮忙，再见！ |  |
|         | 我很乐意，再见！ |  |
# Hugging Face Tasks

## Natural Language Processing

<table>
<thead>
<tr>
<th>Task</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Classification</td>
<td>3345 models</td>
</tr>
<tr>
<td>Token Classification</td>
<td>1492 models</td>
</tr>
<tr>
<td>Question Answering</td>
<td>1140 models</td>
</tr>
<tr>
<td>Translation</td>
<td>1467 models</td>
</tr>
<tr>
<td>Summarization</td>
<td>323 models</td>
</tr>
<tr>
<td>Text Generation</td>
<td>3959 models</td>
</tr>
<tr>
<td>Fill-Mask</td>
<td>2453 models</td>
</tr>
<tr>
<td>Sentence Similarity</td>
<td>352 models</td>
</tr>
</tbody>
</table>

[https://huggingface.co/tasks](https://huggingface.co/tasks)
NLP with Transformers Github

https://github.com/nlp-with-transformers/notebooks
NLP with Transformers Github Notebooks

Running on a cloud platform

To run these notebooks on a cloud platform, just click on one of the badges in the table below:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Colab</th>
<th>Kaggle</th>
<th>Gradient</th>
<th>Studio Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Text Classification</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Transformer Anatomy</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Multilingual Named Entity Recognition</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Text Generation</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Summarization</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Question Answering</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Making Transformers Efficient in Production</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Dealing with Few to No Labels</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Training Transformers from Scratch</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
<tr>
<td>Future Directions</td>
<td><img src="#" alt="Open in Colab" /></td>
<td><img src="#" alt="Open in Kaggle" /></td>
<td><img src="#" alt="Run on Gradient" /></td>
<td><img src="#" alt="Open" /></td>
</tr>
</tbody>
</table>

Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using Kaggle, Gradient, or SageMaker Studio Lab. These platforms tend to provide more performant GPUs like P100s, all for free!

https://github.com/nlp-with-transformers/notebooks
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

NLP with Transformers

Natural Language Processing with Transformers

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

```python
  2 cd notebooks
  3 from install import *
  4 install_requirements()

[3] 1 from util import *
  2 setup_chapter()

[12] 1 text = """Dear Amazon, last week I ordered on Optimus Prime action figure \n  2 from your online store in Germany. Unfortunately, when I opened the package, \n  3 I discovered to my horror that I had been sent an action figure of Megatron \n  4 instead! As a lifelong enemy of the Decepticons, I hope you can understand my \n  5 dilemma. To resolve the issue, I demand an exchange of Megatron for the \n  6 Optimus Prime figure I ordered. Enclosed are copies of my records concerning \n  7 this purchase. I expect to hear from you soon. Sincerely, Rumblebee.""

Text Classification

[13] 1 from transformers import pipeline
  2 classifier = pipeline("text-classification")

[14] 1 import pandas as pd
  2 outputs = classifier(text)
  3 pd.DataFrame(outputs)
```

https://tinyurl.com/aintpuppython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4z1zTunjMqf2RkCrT

Text Classification

Text Classification with Transformers

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

```python
import os
os.environ['CUDA_VISIBLE_DEVICES'] = ''
```

Uncomment and run this cell if you're on Colab or Kaggle

```python
!git clone https://github.com/nlp-with-transformers/notebooks.git
to run notebooks
```

```
from install import *
install_requirements()
```

```
from utils import *
```

The Dataset

```python
from datasets import list_datasets
all_datasets = list_datasets()
print(f"There are {len(all_datasets)} datasets currently available on the Hub")
print(f"The first 10 are: {all_datasets[:10]}")
```

There are 3783 datasets currently available on the Hub
The first 10 are: ['acronym_identification', 'ade_corpus_v2', 'adversarial_qa',

https://tinyurl.com/aintpupython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Named Entity Recognition (NER)

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Github: https://github.com/nlp-with-transformers/notebooks

```
from transformers import pipeline
classifier = pipeline("ner")
classifier("Hello I'm Omar and I live in Zürich.")
```

https://tinyurl.com/aintpuppython101
**Text Summarization**

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O’Reilly Media.

Github: [https://github.com/nlpaudio/transformers](https://github.com/nlpaudio/transformers)

1. `pip install transformers`
2. `from transformers import pipeline`
3. `classifier = pipeline("summarization")`
4. `text = "Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents as of 2018, in an area of more than..."`
5. `classifier(text, max_length=30)`

No model was supplied, default to ashleifer/distilbart-cnn-12-6 ([https://huggingface.co/ashleifer/distilbart-cnn-12-6](https://huggingface.co/ashleifer/distilbart-cnn-12-6))

Your min_length=56 must be inferior than your max_length=30.

[{'summary_text': 'Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents. The City of Paris'}]

1. `pip install transformers`
2. `text = "Dear Amazon, last week I ordered an Optimus Prime action figure..."`
3. `from your online store in Germany. Unfortunately, when I opened the package,..."`
4. `I discovered to my horror that I had been sent an action figure of Megatron instead! As a lifelong enemy of the Decepticons, I hope you can understand my..."`
5. `dilemma. To resolve the issue, I demand an exchange of Megatron for the..."`
6. `Optimus Prime figure I ordered. Enclosed are copies of my records concerning..."`
7. `this purchase. I expect to hear from you soon. Sincerely, Bumblebee.""``
8. `from transformers import pipeline`
9. `summarizer = pipeline("summarization")`
10. `outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)`
11. `print(outputs[0]["summary_text")]

[https://tinyurl.com/aintpupy101](https://tinyurl.com/aintpupy101)
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvfwUbeo4zJ1zTunjMqf2RkCrT

Text Generation

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Github: https://github.com/nlp-with-transformers/notebooks

```python
[9] 1 #Source: https://huggingface.co/tasks/text-generation
2 2 pip install transformers
3 from transformers import pipeline
4 generator = pipeline('text-generation', model = 'gpt2')
5 generator('Hello, I’m a language model', max_length = 30, num_return_sequences=3)

Setting 'pad_token_id' to 'eos_token_id':50256 for open-end generation.

{'generated_text': 'Hello, I’m a language model. But then, one day, I’m not trying to teach the language in my head."

'generated_text': 'Hello, I’m a language model, not a programmer. As you know, languages are not a linear model. The thing that jumps out at')

[1] 1 from transformers import pipeline
2 generator = pipeline('text-generation', model = 'gpt2')
3 outputs = generator('Once upon a time', max_length = 30, num_return_sequences=3)
4 print(outputs[0]['generated_text'])

Setting 'pad_token_id' to 'eos_token_id':50256 for open-end generation.

Once upon a time, every person who ever saw Jesus, knew that He was Christ. And even though he might not have known Him, He was

https://tinyurl.com/aintpupython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Question Answering and Dialogue Systems

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

```python
[3] !pip install transformers
  from transformers import pipeline
  qmodel = pipeline("question-answering")
  question = "Where do I live?"
  context = "My name is Michael and I live in Taipei."
  qmodel(question = question, context = context)

1 from transformers import pipeline
2 qmodel = pipeline("question-answering")
3 question = "Where do I live?"
4 context = "My name is Michael and I live in Taipei."
5 qmodel(question = question, context = context)
6 {'answer': 'Taipei', 'end': 39, 'score': 0.9370741381645203, 'start': 33}

No model was supplied, defaulted to distilbert-base-cased-distilled-squad (https://huggingface.co/distilbert-base-cased-distilled-squad) {'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
```

https://tinyurl.com/aintpupython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Question Answering

```python
from transformers import pipeline
qamodel = pipeline("question-answering", model='deepset/roberta-base-squad2')

# Question: What causes precipitation to fall?
question = "What causes precipitation to fall?"
context = "In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity"
output = qamodel(question=question, context=context)
print(output['answer'])

# Question: What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
question = "What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?"
context = "In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity"
output = qamodel(question=question, context=context)
print(output['answer'])

# Question: Where do water droplets collide with ice crystals to form precipitation?
question = "Where do water droplets collide with ice crystals to form precipitation?"
context = "In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity"
output = qamodel(question=question, context=context)
print(output['answer'])
```

https://tinyurl.com/aintpuppython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Question Answering and Dialogue Systems

Question Answering (QA)

BERT for Question Answering

Source: Apoorv Nandan (2020), BERT (from HuggingFace Transformers) for Text Extraction, https://keras.io/examples/nlp/text_extraction_with_bert/

Description: Fine tune pretrained BERT from HuggingFace Transformers on SQuAD.

Introduction

This demonstration uses SQuAD (Stanford Question-Answering Dataset). In SQuAD, an input consists of a question, and a paragraph for context. The goal is to find the span of text in the paragraph that answers the question. We evaluate our performance on this data with the "Exact Match" metric, which measures the percentage of predictions that exactly match any one of the ground-truth answers.

We fine-tune a BERT model to perform this task as follows:

1. Feed the context and the question as inputs to BERT.
2. Take two vectors S and T with dimensions equal to that of hidden states in BERT.
3. Compute the probability of each token being the start and end of the answer span. The probability of a token being the start of the answer is given by a dot product between S and the representation of the token in the last layer of BERT, followed by a softmax over all tokens. The probability of a token being the end of the answer is compute similarly with the vector T.
4. Fine-tune BERT and learn S and T along the way.

References:
- BERT
- SQuAD

https://tinyurl.com/aintpupython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvfwUbeo4zJ1zTunjMqf2RkCrT

---

**Table of contents**

- RandomizedSearchCV
- Sentiment Analysis
  - Unsupervised
  - Lexical
- Sentiment Analysis - Supervised
  - Machine Learning
- Sentiment Analysis - Supervised
  - Deep Learning Models
- Sentiment Analysis - Advanced Deep Learning
- Deep Learning and Universal Sentence-Embedding Models
  - Universal Sentence Encoder (USE)
  - Universal Sentence Encoder Multilingual (USEM)
- Question Answering and Dialogue Systems
  - Question Answering (QA)
    - BERT for Question Answering
- Dialogue Systems
  - Joint Intent Classification and Slot Filling with Transformers
- Data Visualization

---

```python
# Python101.ipynb

Table of contents
```

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
<th>Connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>input_1 (InputLayer)</td>
<td>(None, 384)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>input_3 (InputLayer)</td>
<td>(None, 384)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>input_2 (InputLayer)</td>
<td>(None, 384)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>tf_bert_model (TFBertModel)</td>
<td>(None, 384, 768), (109482240)</td>
<td>input_1[0][0]</td>
<td></td>
</tr>
<tr>
<td>start_logit (Dense)</td>
<td>(None, 384, 1)</td>
<td>768</td>
<td>tf_bert_model[0][0]</td>
</tr>
<tr>
<td>end_logit (Dense)</td>
<td>(None, 384, 1)</td>
<td>768</td>
<td>tf_bert_model[0][0]</td>
</tr>
<tr>
<td>flatten (Flatten)</td>
<td>(None, 384)</td>
<td>0</td>
<td>start_logit[0][0]</td>
</tr>
<tr>
<td>flatten_1 (Flatten)</td>
<td>(None, 384)</td>
<td>0</td>
<td>end_logit[0][0]</td>
</tr>
<tr>
<td>activation_7 (Activation)</td>
<td>(None, 384)</td>
<td>0</td>
<td>flatten[0][0]</td>
</tr>
<tr>
<td>activation_8 (Activation)</td>
<td>(None, 384)</td>
<td>0</td>
<td>flatten_1[0][0]</td>
</tr>
</tbody>
</table>

Total params: 109,483,776
Trainable params: 109,483,776
Non-trainable params: 0

CPU times: user 20.8 s, sys: 7.75 s, total: 28.5 s
Wall time: 42 s

---

https://tinyurl.com/aintpupython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Dialogue Systems

Joint Intent Classification and Slot Filling with Transformers

The goal of this notebook is to fine-tune a pretrained transformer-based neural network model to convert a user query expressed in English into a representation that is structured enough to be processed by an automated service.

Here is an example of interpretation computed by such a Natural Language Understanding system:

```python
>>> msg="Book a table for two at Le Ritz for Friday night",
    tokenizer, joint_model, intent_names, slot_names)

{  'intent': 'BookRestaurant',
   'slots': {
      'party_size_number': 'two',
      'restaurant_name': 'Le Ritz',
      'timeRange': 'Friday night'
   }
}
```

Intent classification is a simple sequence classification problem. The trick is to treat the structured knowledge extraction part ("Slot Filling") as token-level classification problem using BIO-annotations:

https://tinyurl.com/aintpupython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

https://tinyurl.com/aintpupython101
Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

https://tinyurl.com/aintpupython101
Question Answering

```python
!pip install transformers
def from transformers import pipeline
def qamodel = pipeline("question-answering")
def question = "Where do I live?"
def context = "My name is Michael and I live in Taipei."
def qamodel(question = question, context = context)

{'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
```

https://tinyurl.com/aintpupython101
from transformers import pipeline
qamodel = pipeline("question-answering", model = 'deepset/roberta-base-squad2')
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
output = qamodel(question = question, context = context)
print(output['answer'])

Taipei
from transformers import pipeline
qamodel = pipeline("question-answering", model = 'deepset/roberta-base-squad2')
question = "What causes precipitation to fall?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers"."""
output = qamodel(question = question, context = context)
print(output["answer"])
Summary

• Question Answering
• Dialogue Systems
• Task Oriented Dialogue System
References

- Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Denis Rothman (2021), Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing.
- Savay Vajjala and Meyyam Asgari-Chenaghi (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.
- Charu C. Aggarwal (2018), Machine Learning for Text, Springer.
- Gabe Ignatow and Rada F. Mihalcea (2017), An Introduction to Text Mining: Research Design, Data Collection, and Analysis, SAGE Publications.
- Rajesh Arumugam (2018), Hands-On Natural Language Processing with Python: A practical guide to applying deep learning architectures to your NLP applications, Packt.
- NLP with Transformers, https://github.com/nlp-with-transformers