Artificial Intelligence for Text Analytics

Question Answering and Dialogue Systems

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

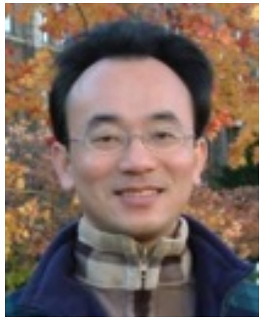
https://meet.google.com/paj-zhhj-mya

aws educate | Cloud Ambassador
2020 Cohort

aws academy Accredited Educator

aws certified Cloud Practitioner

aws certified Solutions Architect Associate



Min-Yuh Day, Ph.D,
Associate Professor
Institute of Information Management, National Taipei University
https://web.ntpu.edu.tw/~myday

2022-05-17

Syllabus

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

Syllabus

Week	Date	Subject/Topics
7	2022/04/05	Tomb-Sweeping Day (Holiday, No Classes)
8	2022/04/12	Midterm Project Report
9	2022/04/19	Multilingual Named Entity Recognition (NER), Text Similarity and Clustering
10	2022/04/26	Text Summarization and Topic Models
11	2022/05/03	Text Generation
12	2022/05/10	Case Study on Artificial Intelligence for Text Analytics II

Syllabus

Week	Date	Subject/Topics
13	2022/05/17	Question Answering and Dialogue Systems
14	2022/05/24	Deep Learning, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics
15	2022/05/31	Final Project Report I
16	2022/06/07	Final Project Report II
17	2022/06/14	Self-learning
18	2022/06/21	Self-learning

Question Answering and Dialogue Systems

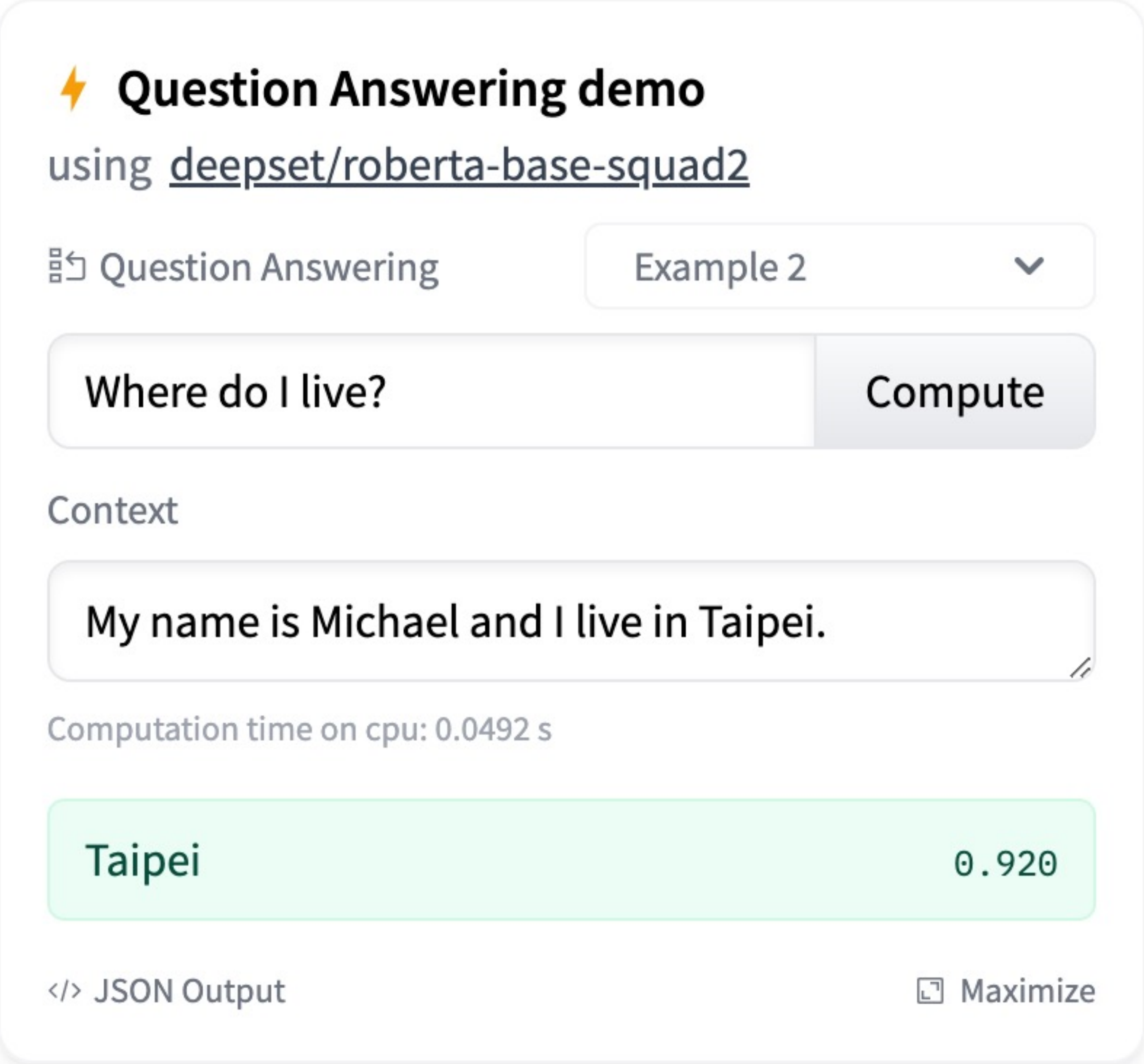
Outline

- Question Answering
- Dialogue Systems
- Task Oriented Dialogue System

Question Answering

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

Question Answering



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}
```

https://tinyurl.com/aintpupython101

Question Answering

```
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
```

https://tinyurl.com/aintpupython101

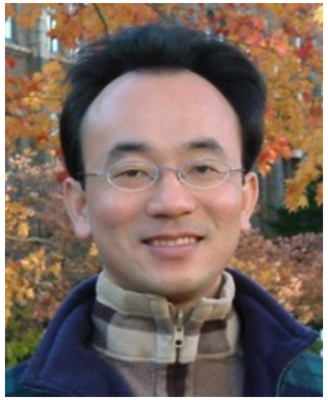
Tamkang University
淡江大學

2011

Tamkang University

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

Department of Information Management
Tamkang University, Taiwan




Min-Yuh Day　　Chun Tu

myday@mail.tku.edu.tw

NTCIR-9 Workshop, December 6-9, 2011, Tokyo, Japan

Tamkang University

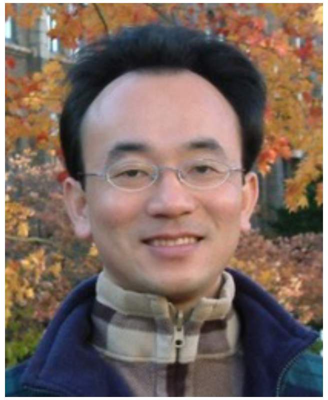
淡江大學

2013

Tamkang University

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

myday@mail.tku.edu.tw

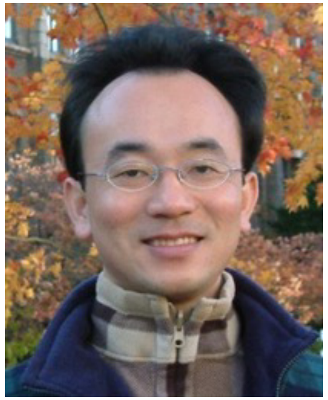
NTCIR-10 Conference, June 18-21, 2013, Tokyo, Japan

IMTKU Textual Entailment System for Recognizing Inference in Text at NTCIR-11 RITE-VAL

2014

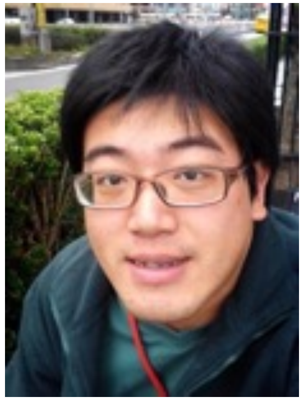
Tamkang University

淡江大學



Min-Yuh Day

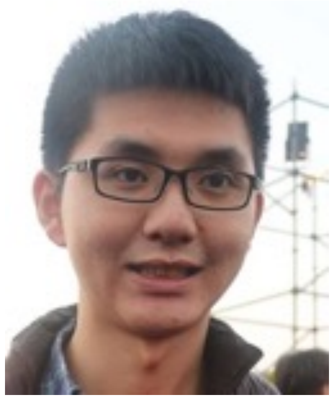
Ya-Jung Wang



Che-Wei Hsu

En-Chun Tu

Huai-Wen Hsu



Yu-An Lin

Shang-Yu Wu

Yu-Hsuan Tai

Cheng-Chia Tsai

Tamkang University
淡江大學

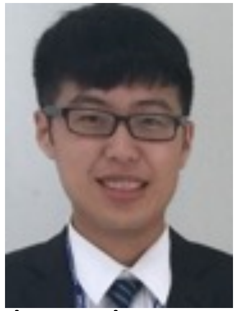
2016

IMTKU Question Answering System for World History Exams at NTCIR-12 QA Lab2

Department of Information Management
Tamkang University, Taiwan

Sagacity Technology


Min-Yuh Day


Cheng-Chia Tsai


Wei-Chun Chung


Hsiu-Yuan Chang


Tzu-Jui Sun


Yuan-Jie Tsai


Jin-Kun Lin


Cheng-Hung Lee


Yu-Ming Guo


Yue-Da Lin


Wei-Ming Chen


Yun-Da Tsai


Cheng-Jhih Han


Yi-Jing Lin


Yi-Heng Chiang


Ching-Yuan Chien

myday@mail.tku.edu.tw

NTCIR

2017

Tamkang University

淡江大學

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

NTCIR

2019

Tamkang University 淡江大學

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

NTCIR

2020

Tamkang University 淡江大學

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

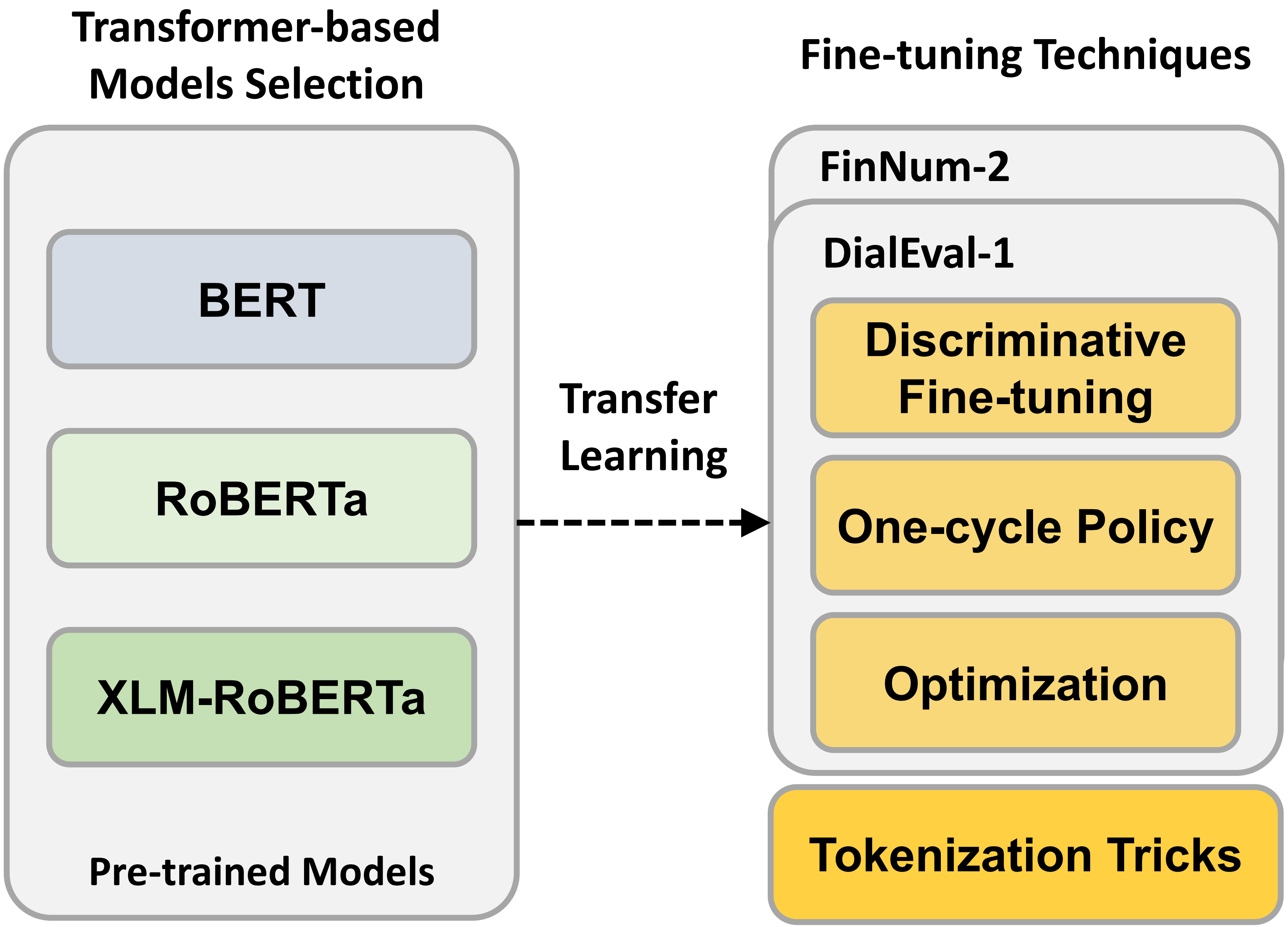
Mike Tian-Jian Jiang[1] Zhao-Xian Gu[2] Cheng-Jhe Chiang[2] Yueh-Chia Wu[2] Yu-Chen Huang[2] Cheng-Han Chiu[2] Sheng-Ru Shaw[2] Min-Yuh Day[3]

NTCIR-15 Conference, December 8-11, 2020, Tokyo, Japan

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)

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

Source: Zeng, Zhaohao, Sosuke Kato, Tetsuya Sakai, and Inho Kang (2020), "Overview of the NTCIR-15 Dialogue Evaluation (DialEval-1) Task", Proceedings of NTCIR-15, 2020



Source: Jiang, Mike Tian-Jian, Shih-Hung Wu, Yi-Kun Chen, Zhao-Xian Gu, Cheng-Jhe Chiang, Yueh-Chia Wu, Yu-Chen Huang, Cheng-Han Chiu, Sheng-Ru Shaw, and Min-Yuh Day (2020). "Fine-tuning techniques and data augmentation on transformer-based models for conversational texts and noisy user-generated content." In 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 919-925. IEEE, 2020.

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

	Japanese	Chinese	English
NTCIR-12 STC-1 22 active participants	Twitter, Retrieval	Weibo, Retrieval	
NTCIR-13 STC-2 27 active participants	Yahoo! News, Retrieval+ Generation	Weibo, Retrieval+ Generation	
NTCIR-14 STC-3		Weibo, Generation for given emotion categories	
		Weibo+English translations, distribution estimation for subjective annotations	

Chinese Emotional Conversation Generation (CECG) subtask

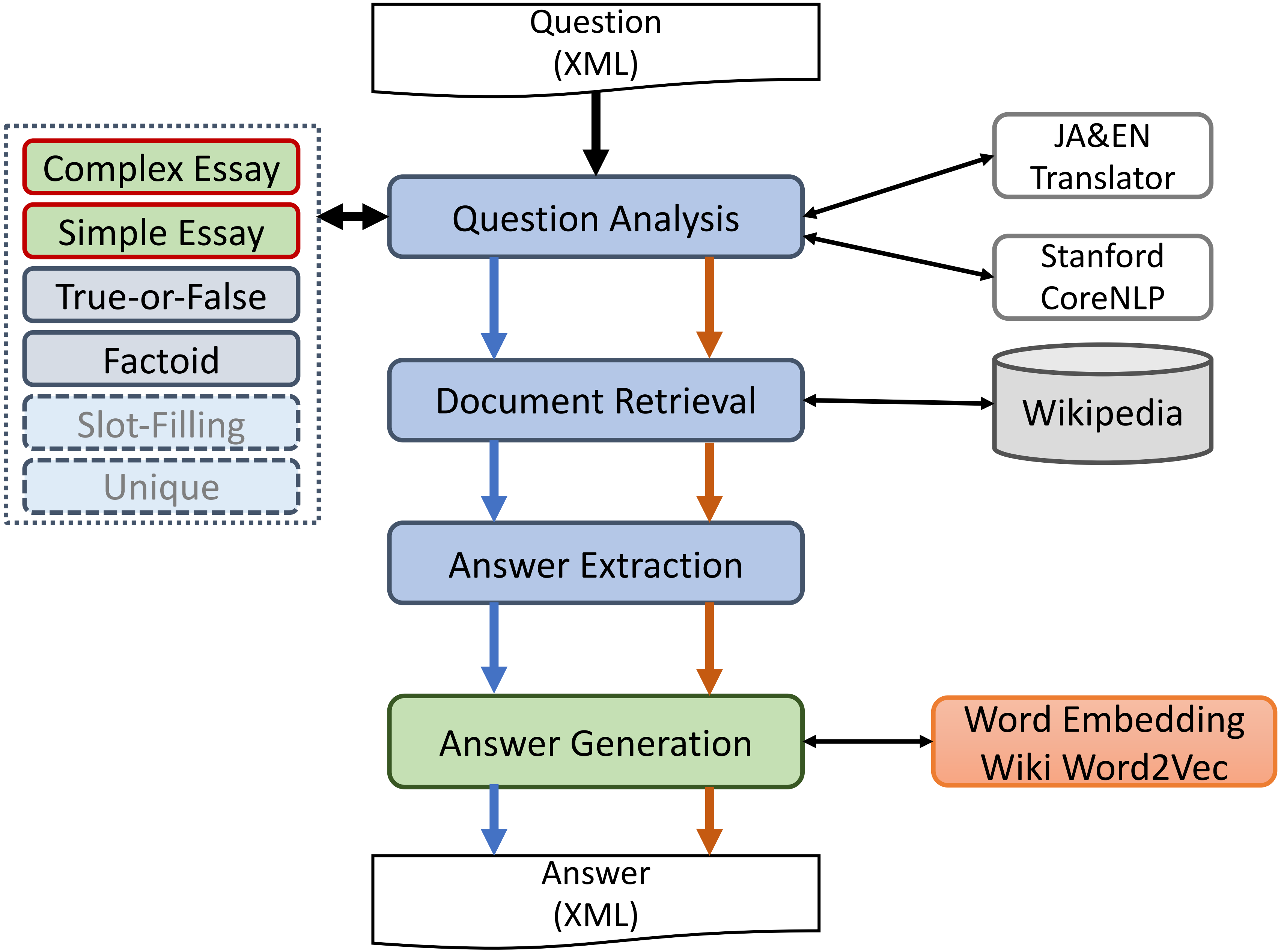
Dialogue Quality (DQ) and Nugget Detection (ND) subtasks

Single-turn, Non task-oriented

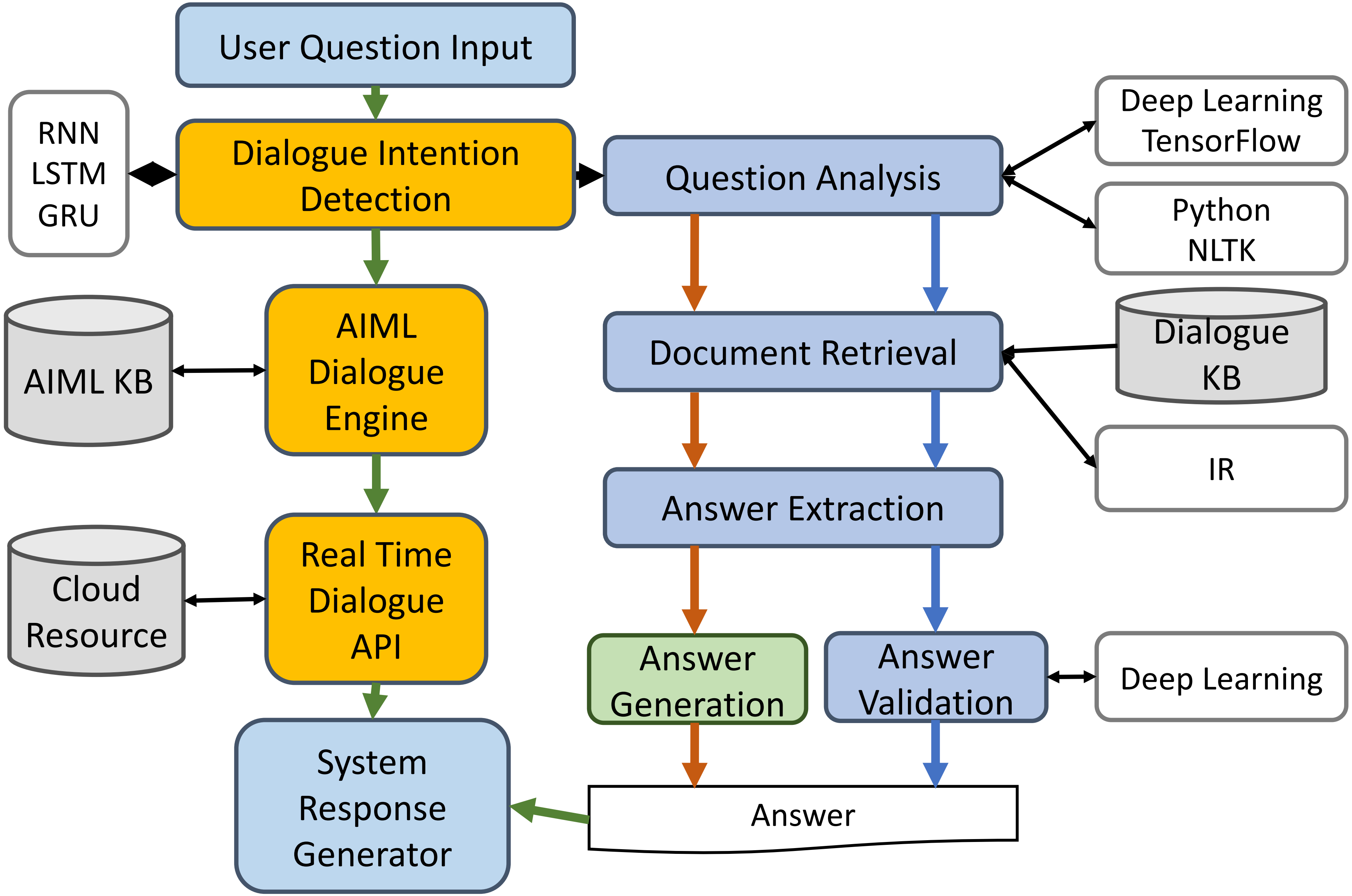
Multi-turn, task-oriented (helpdesk)

Source: https://waseda.app.box.com/v/STC3atNTCIR-14

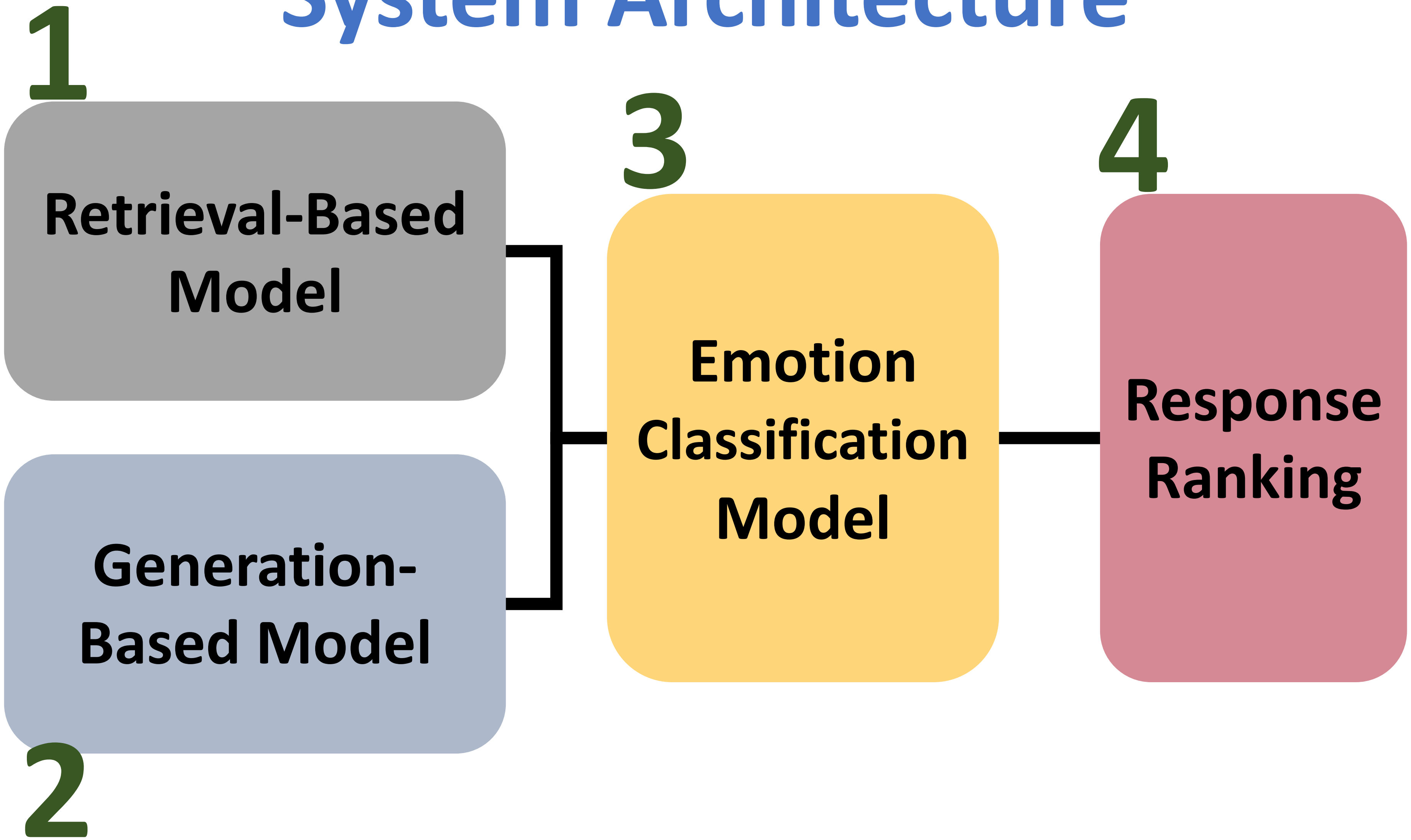
IMTKU System Architecture for NTCIR-13 QALab-3



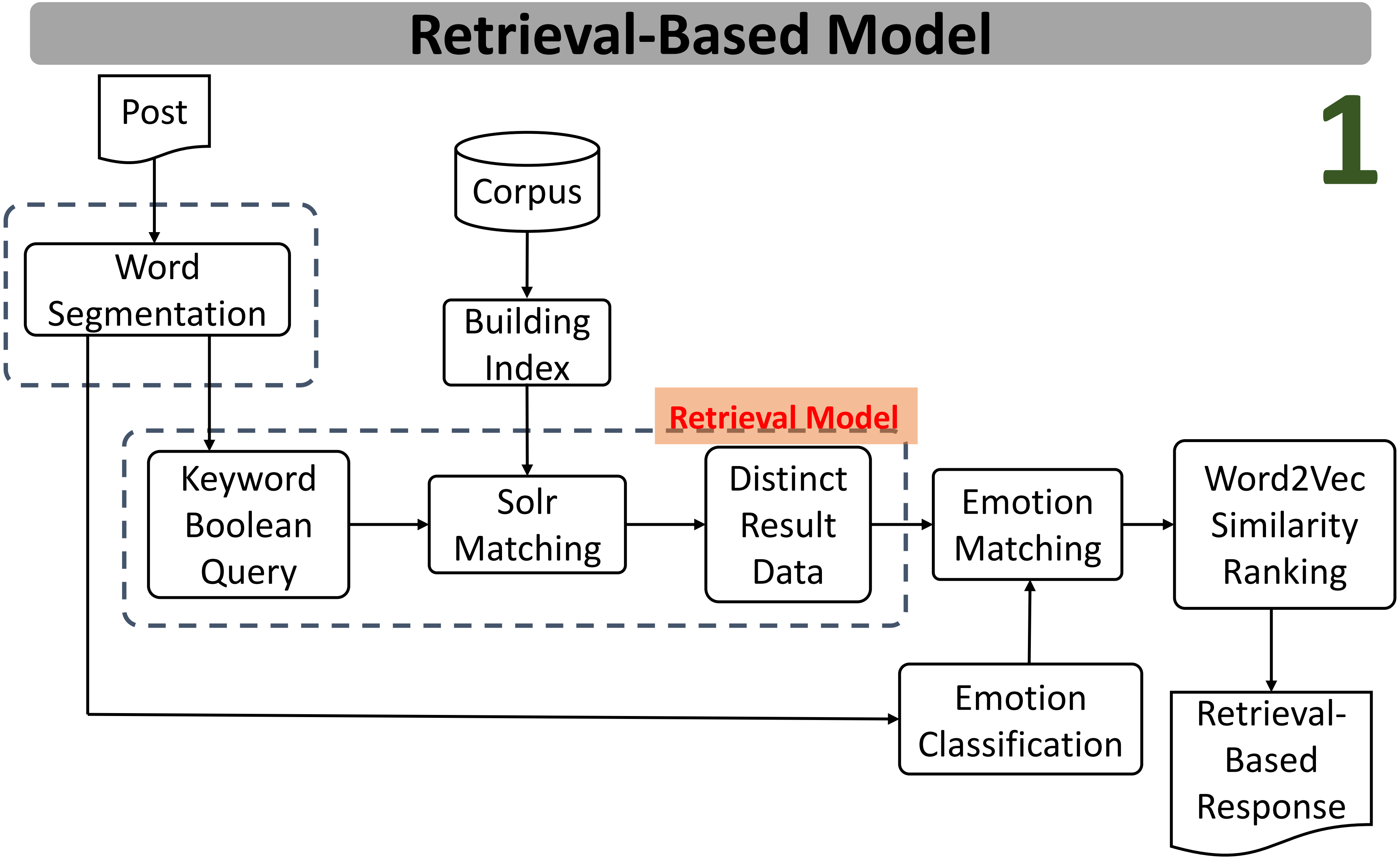
System Architecture of Intelligent Dialogue and Question Answering System



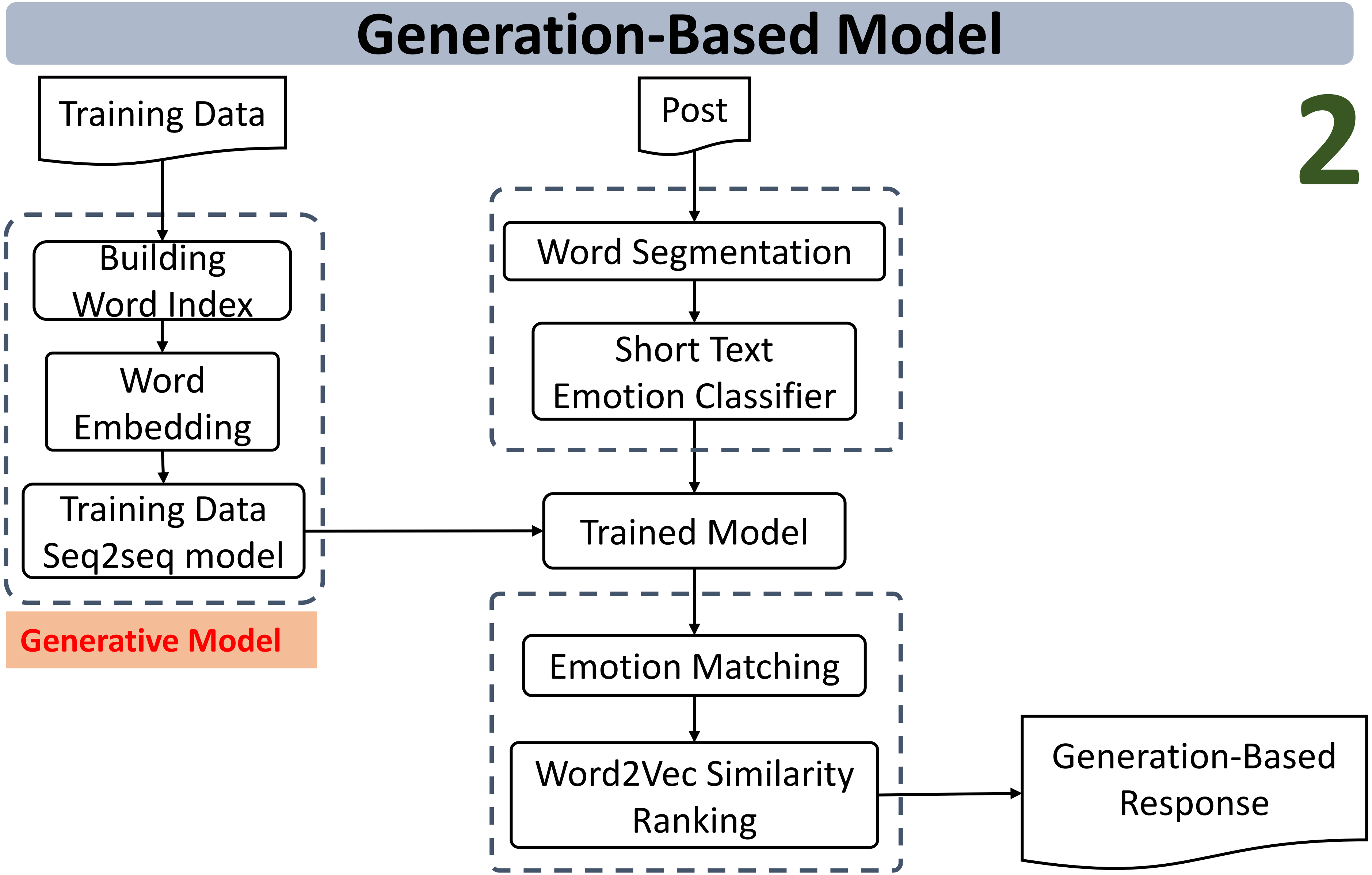
IMTKU Emotional Dialogue System Architecture



The system architecture of IMTKU retrieval-based model for NTCIR-14 STC-3



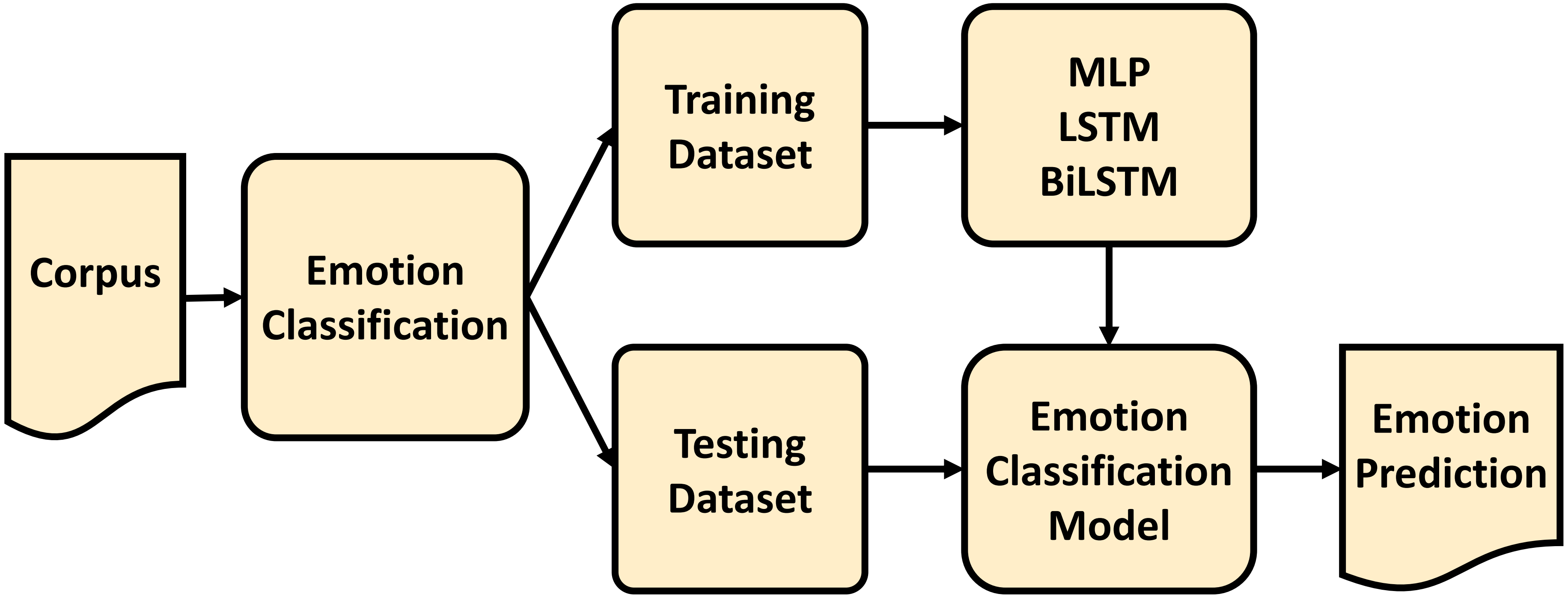
The system architecture of IMTKU generation-based model for NTCIR-14 STC-3



The system architecture of IMTKU emotion classification model for NTCIR-14 STC-3

Emotion Classification Model

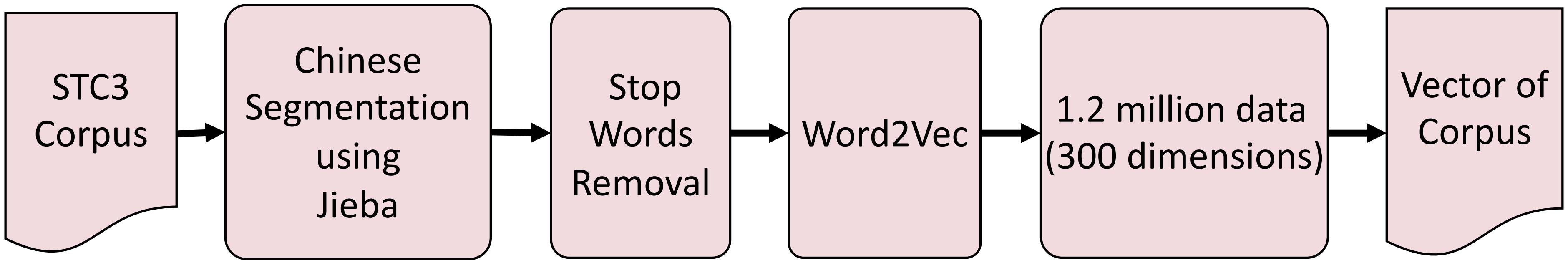
3



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

Response Ranking

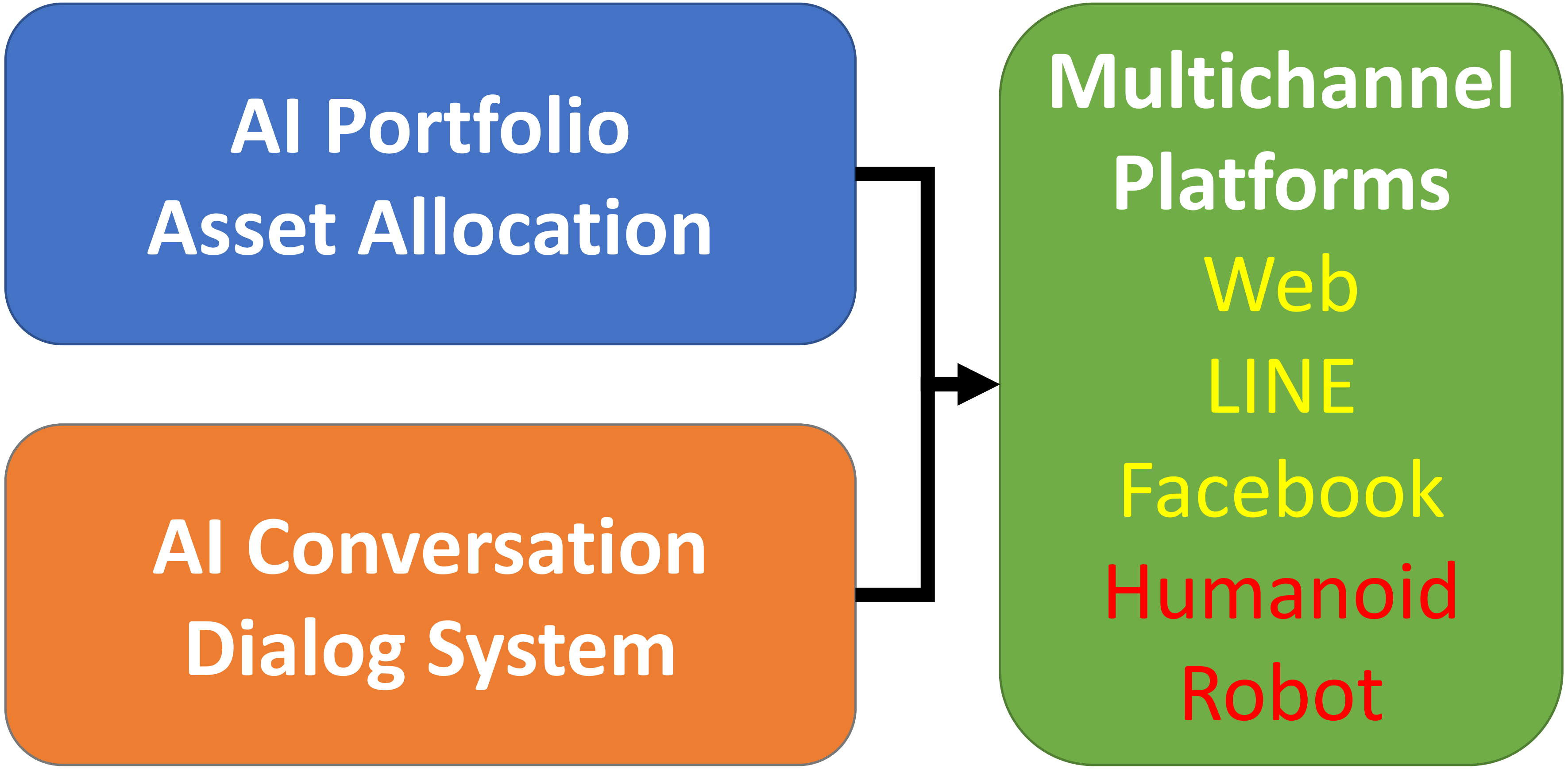
4



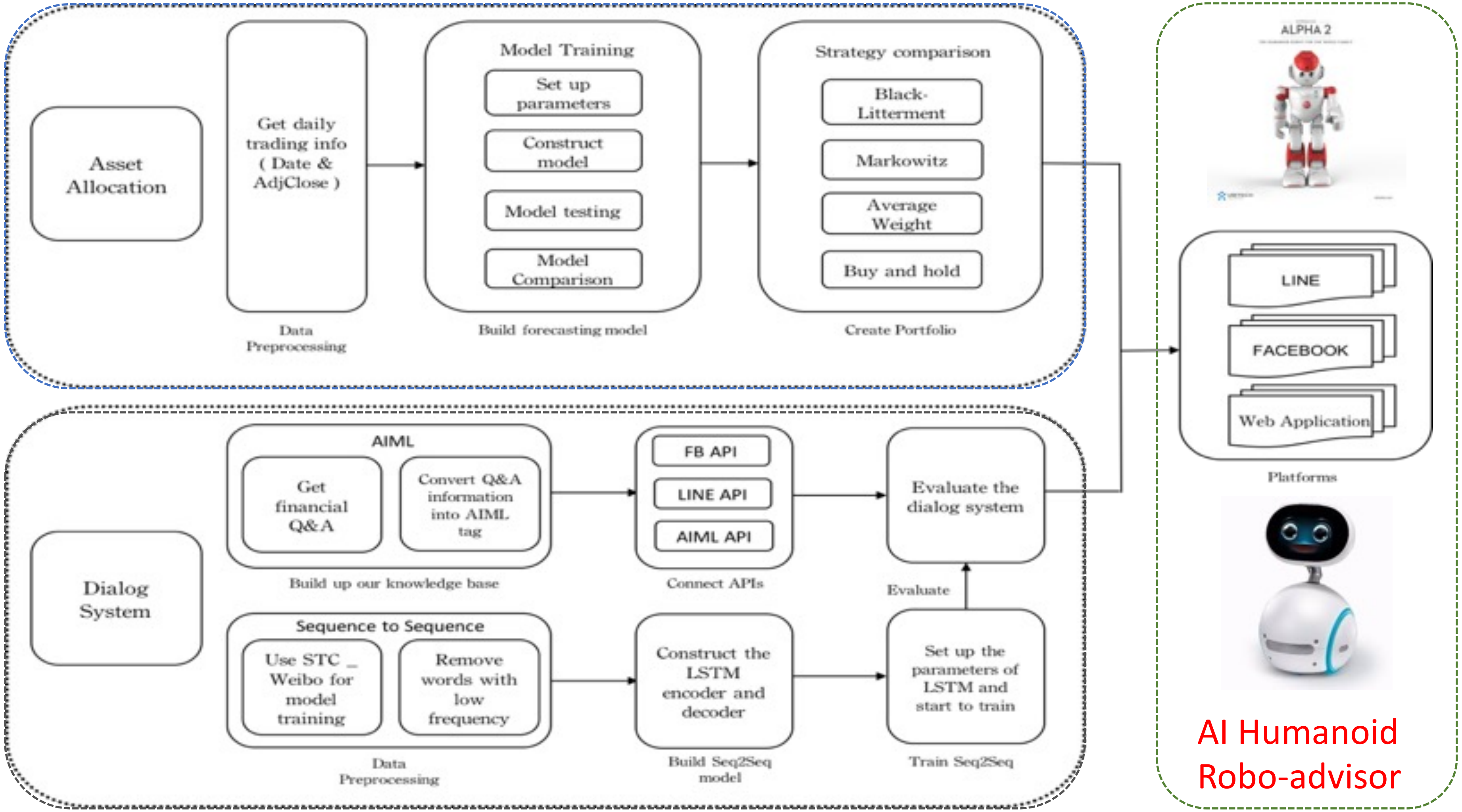
AI Humanoid
Robo-Advisor

AI Humanoid Robo-Advisor

for Multi-channel Conversational Commerce

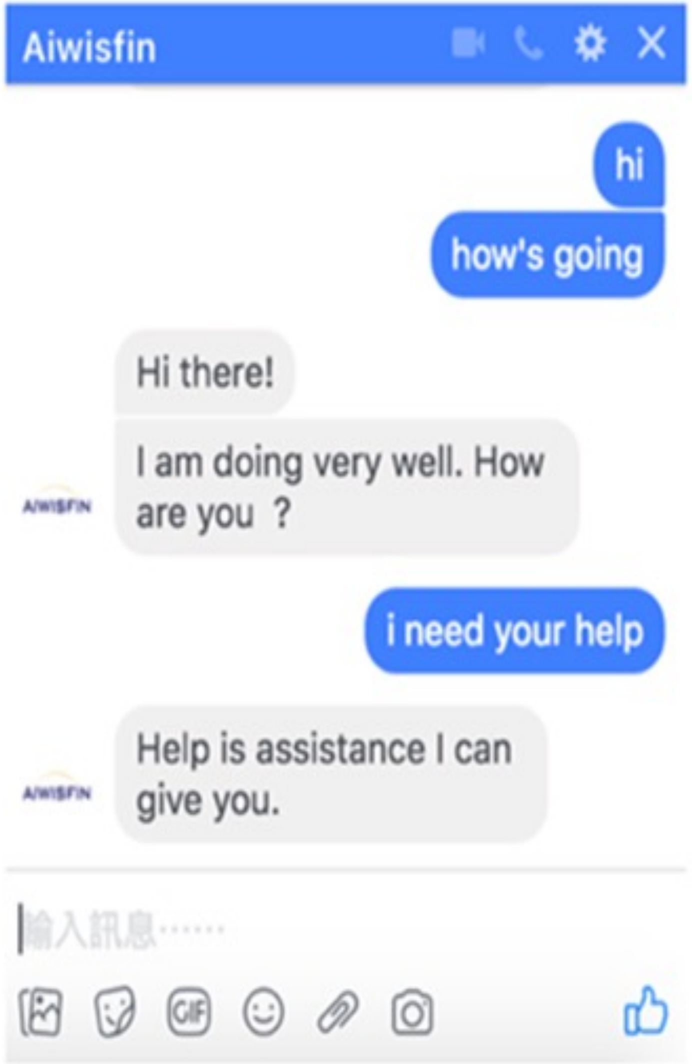


System Architecture of AI Humanoid Robo-Advisor



Conversational Model

(LINE, FB Messenger)



Conversational Robo-Advisor Multichannel UI/UX Robots

ALPHA 2

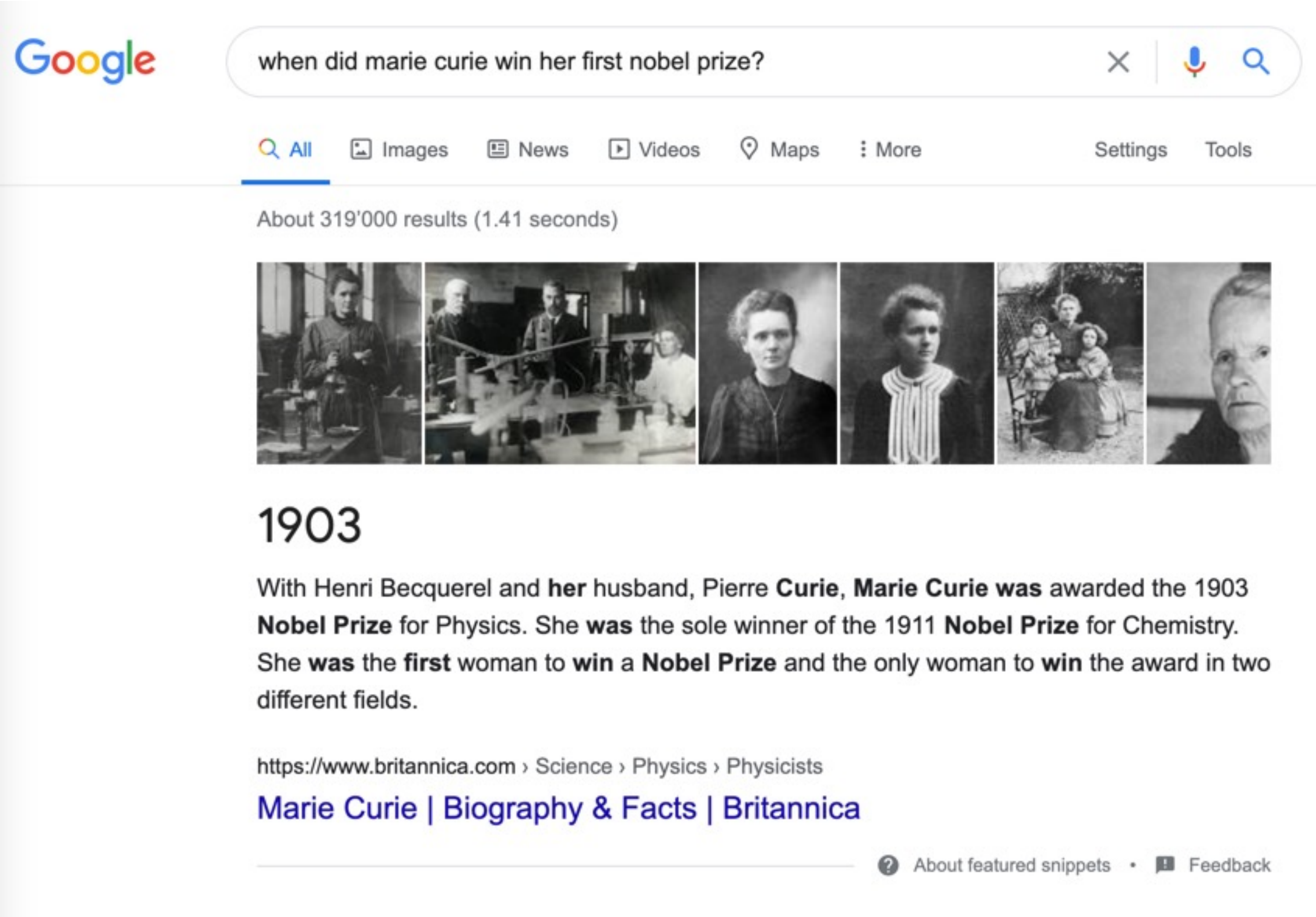
ZENBO

Question Answering

Question Answering

When did Marie Curie win her first Nobel Prize?

1903

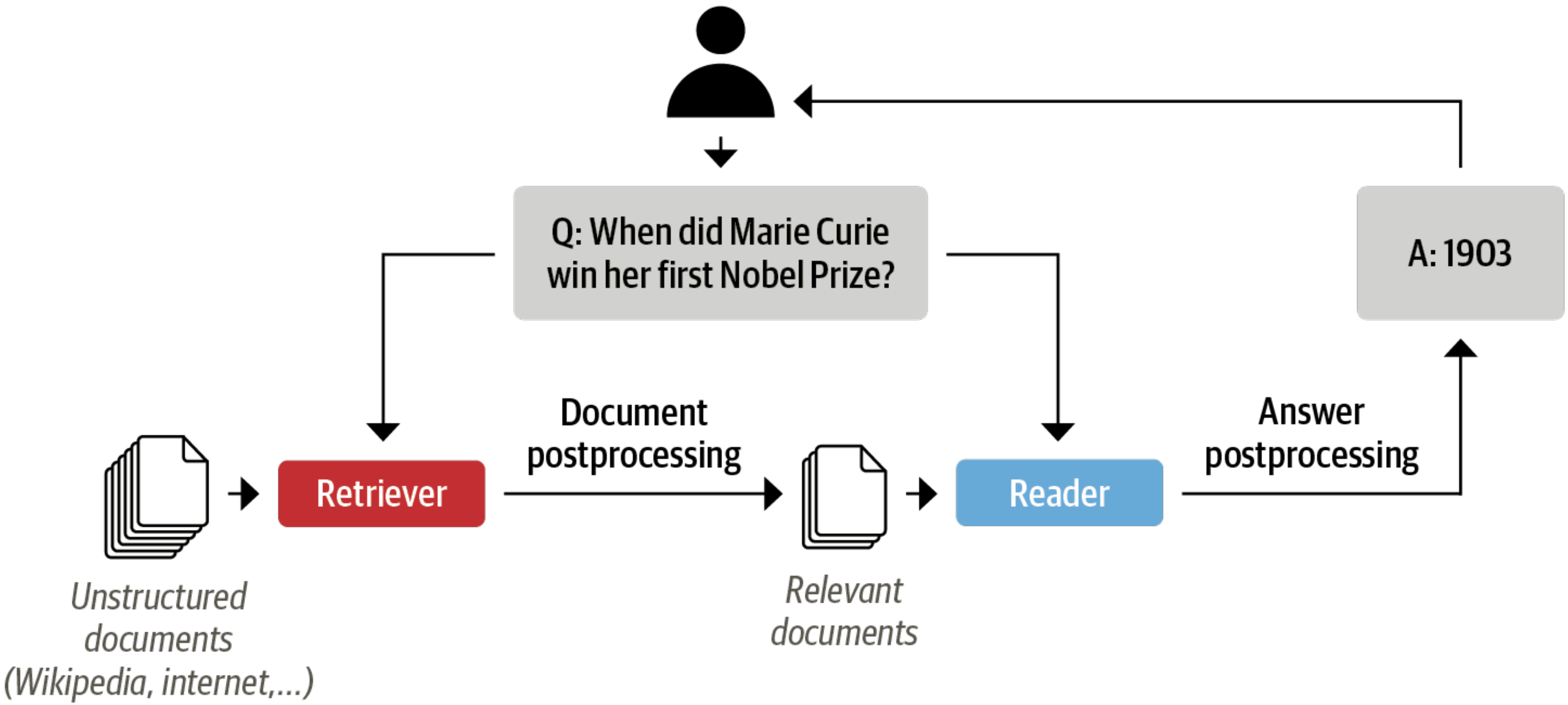


Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

The Retriever-Reader Architecture for Modern QA Systems

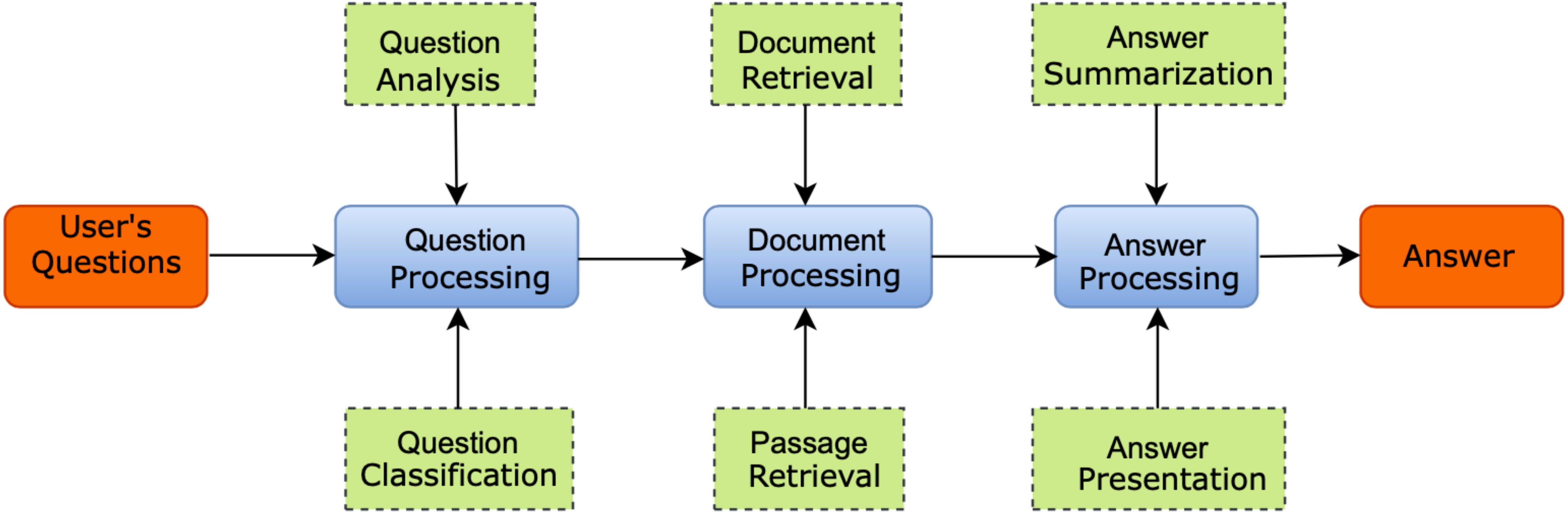
When did Marie Curie win her first Nobel Prize?

1903



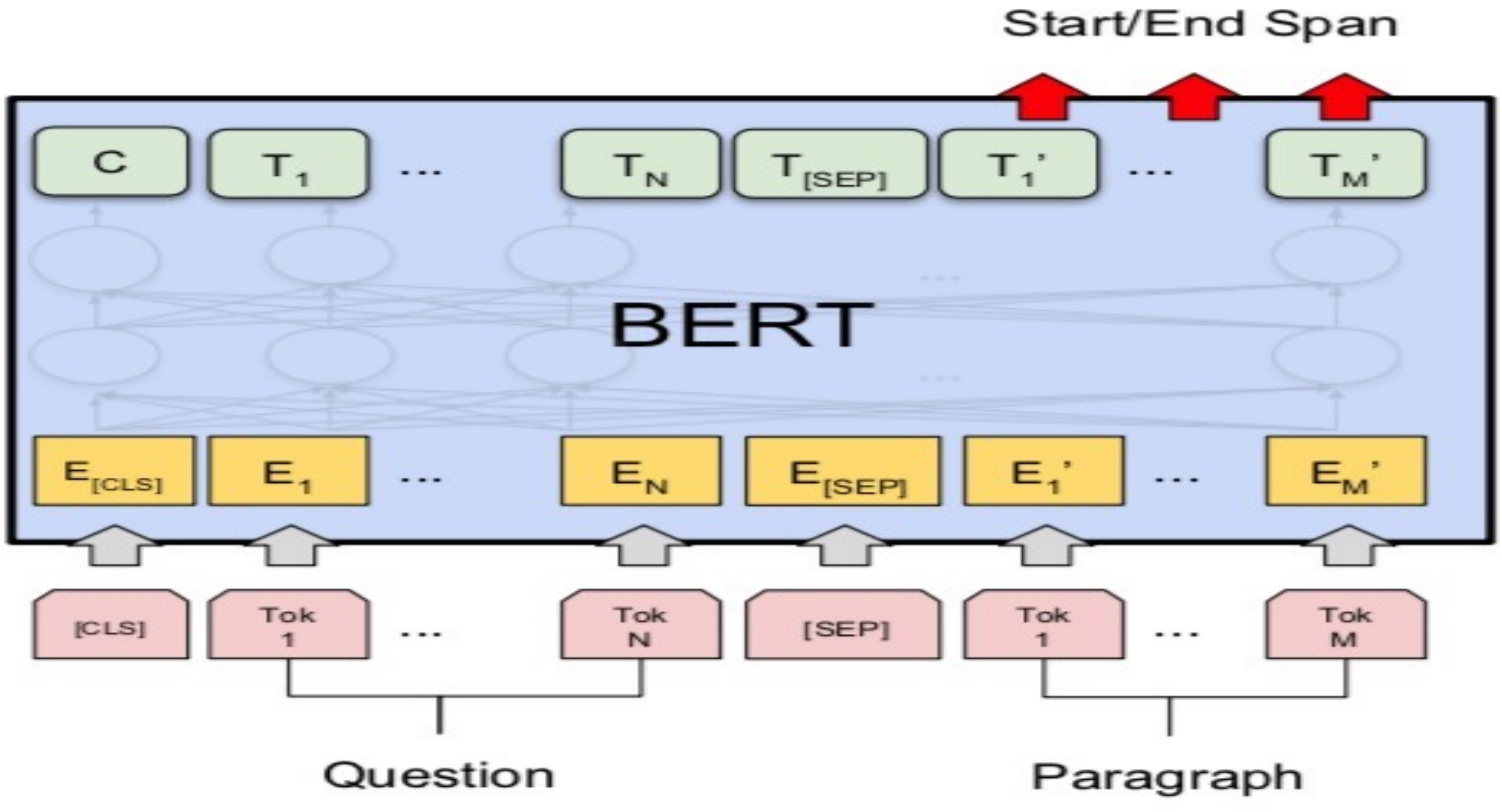
Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Question Answering System (QAS)



Source: Mutabazi, Emmanuel, Jianjun Ni, Guangyi Tang, and Weidong Cao (2021) "A review on medical textual question answering systems based on deep learning approaches." Applied Sciences 11, no. 12 (2021): 5456.

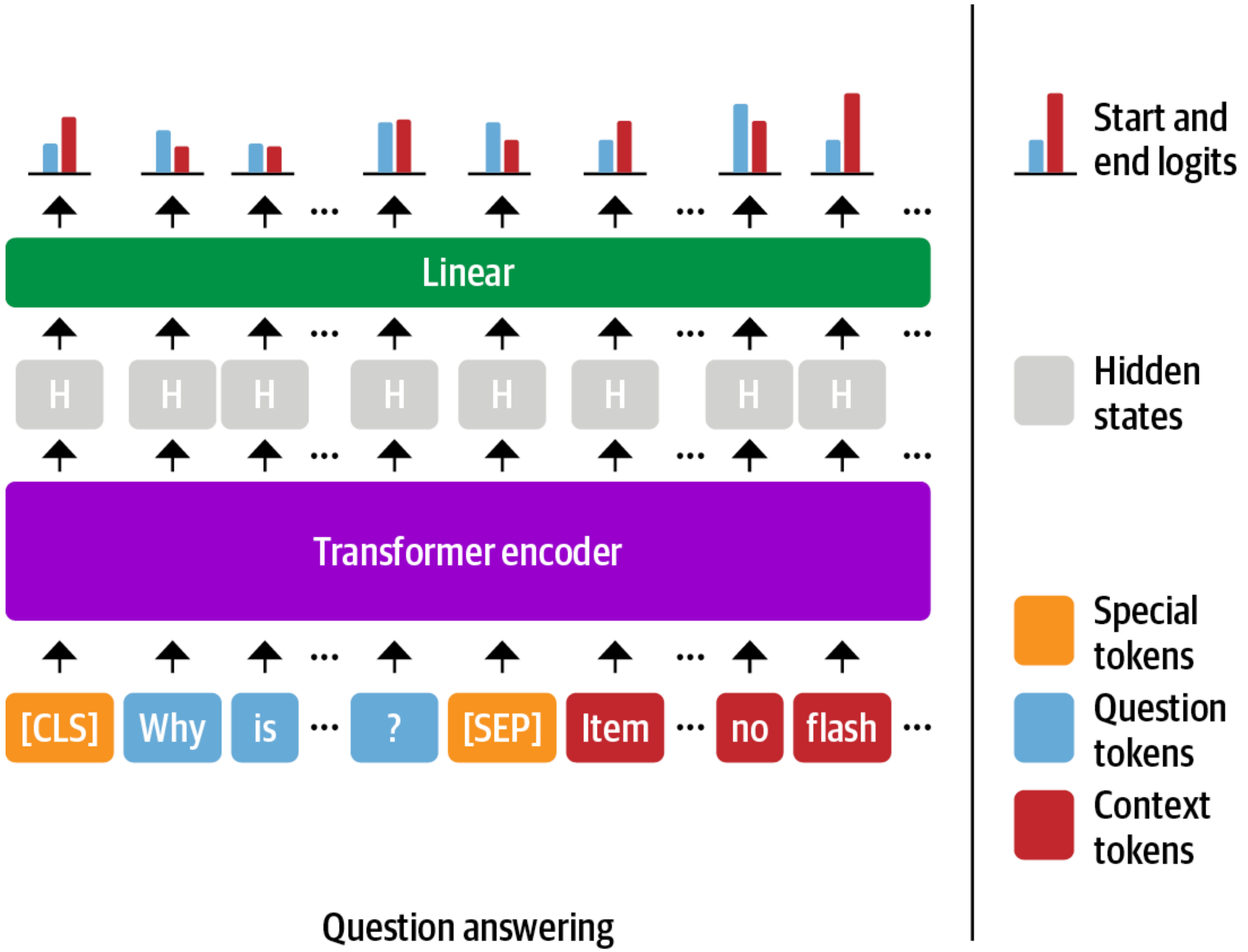
Fine-tuning BERT on Question Answering (QA)



(c) Question Answering Tasks: SQuAD v1.1

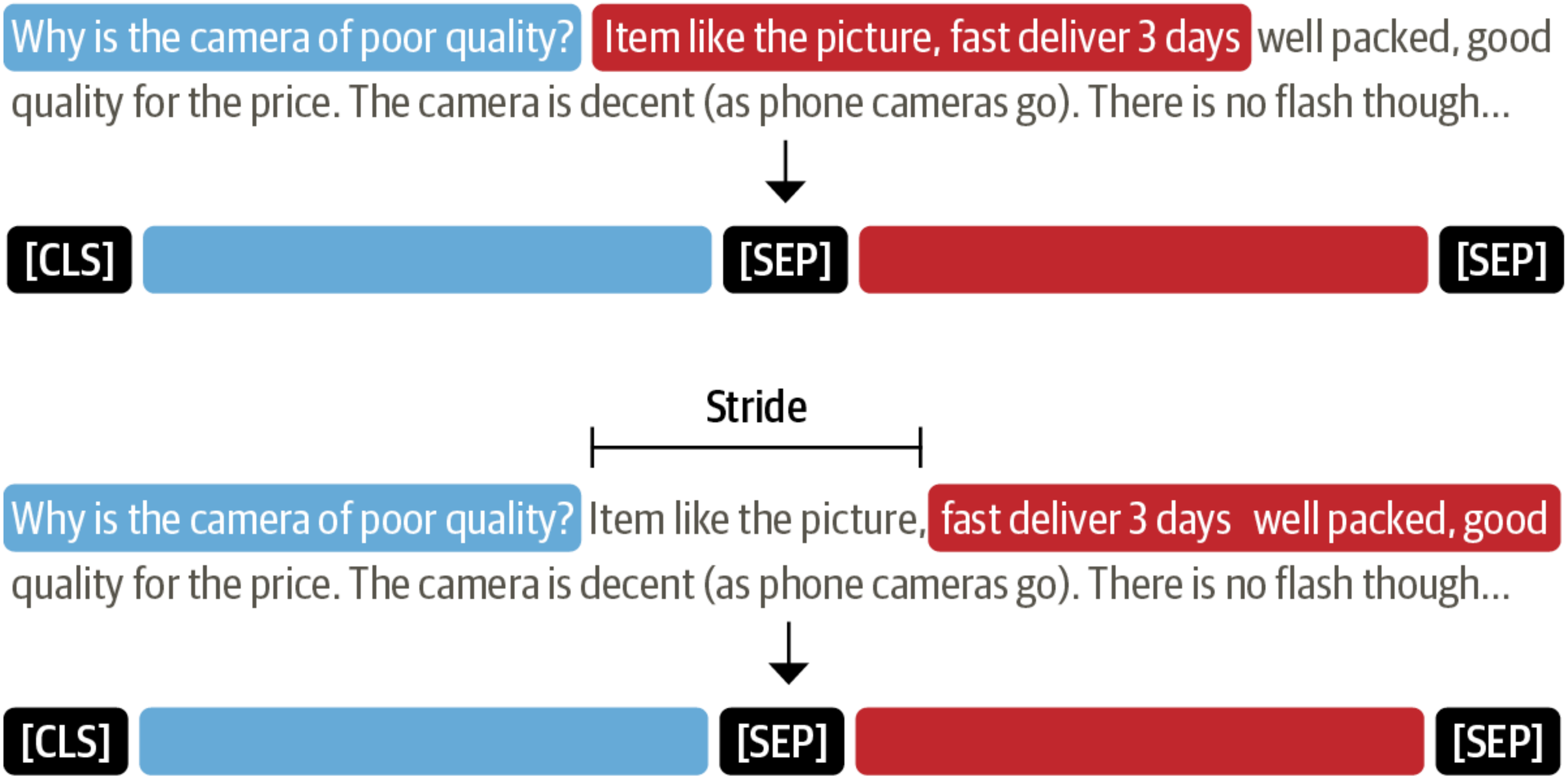
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

The span classification head for QA tasks



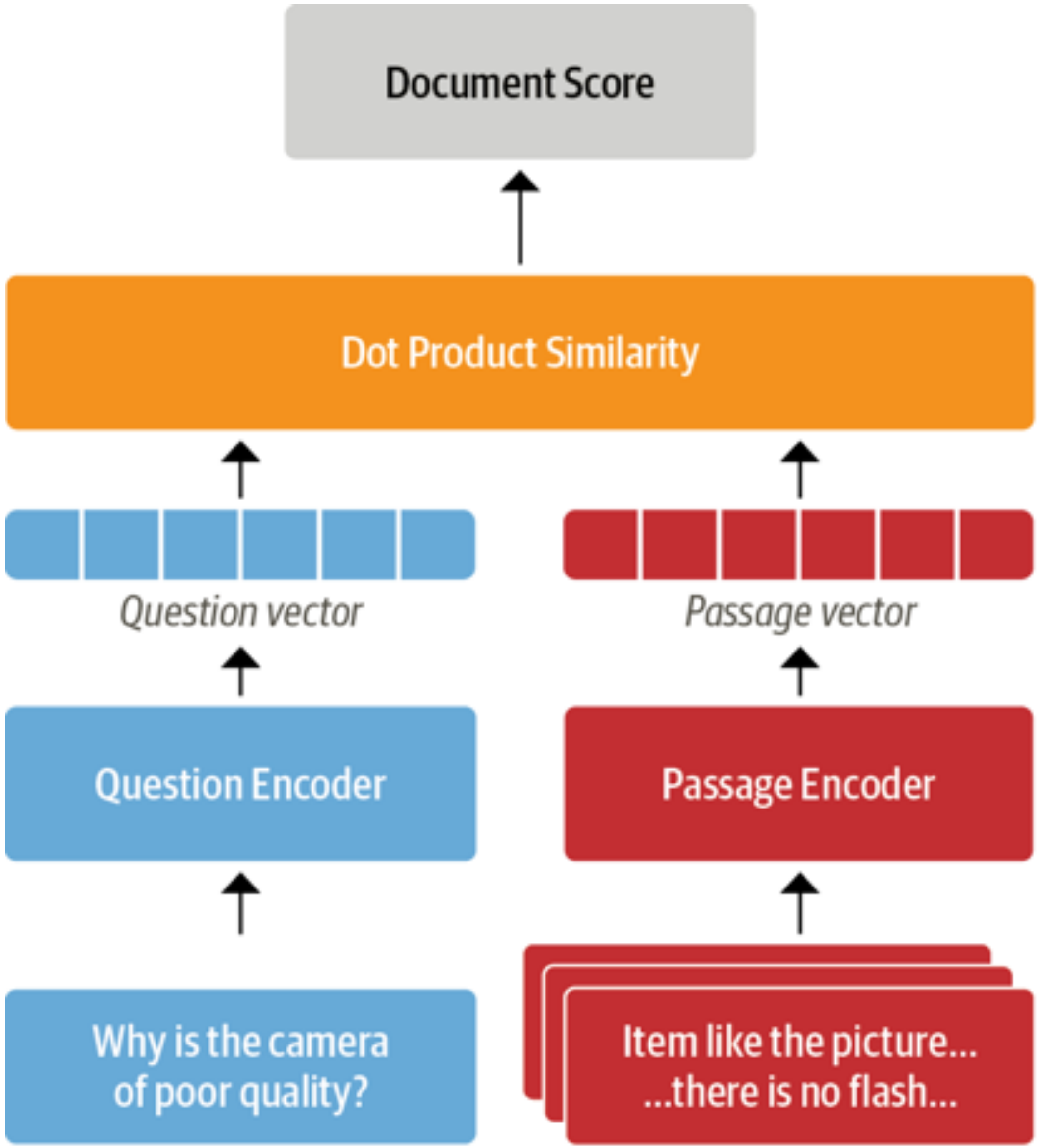
Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Multiple question-context pairs



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

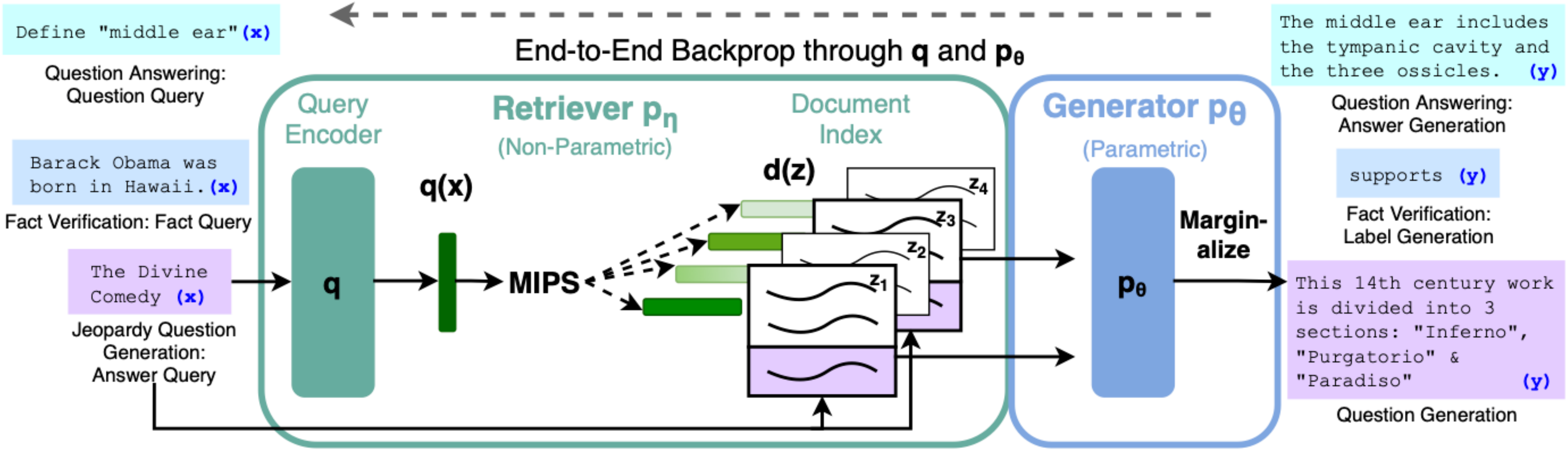
Dense Passage Retrieval (DPR)



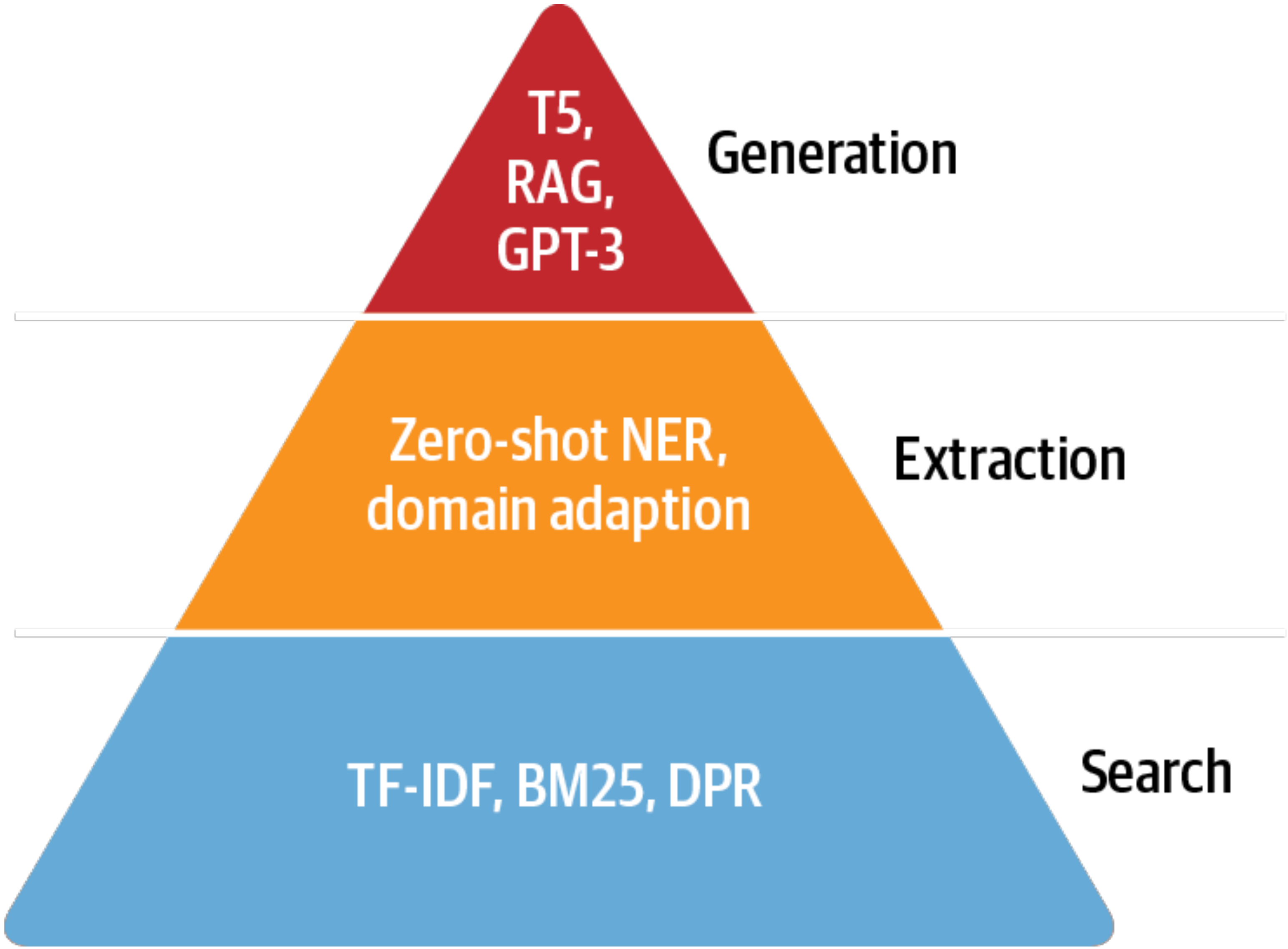
Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Going Beyond Extractive QA
Retrieval-Augmented Generation (RAG)

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



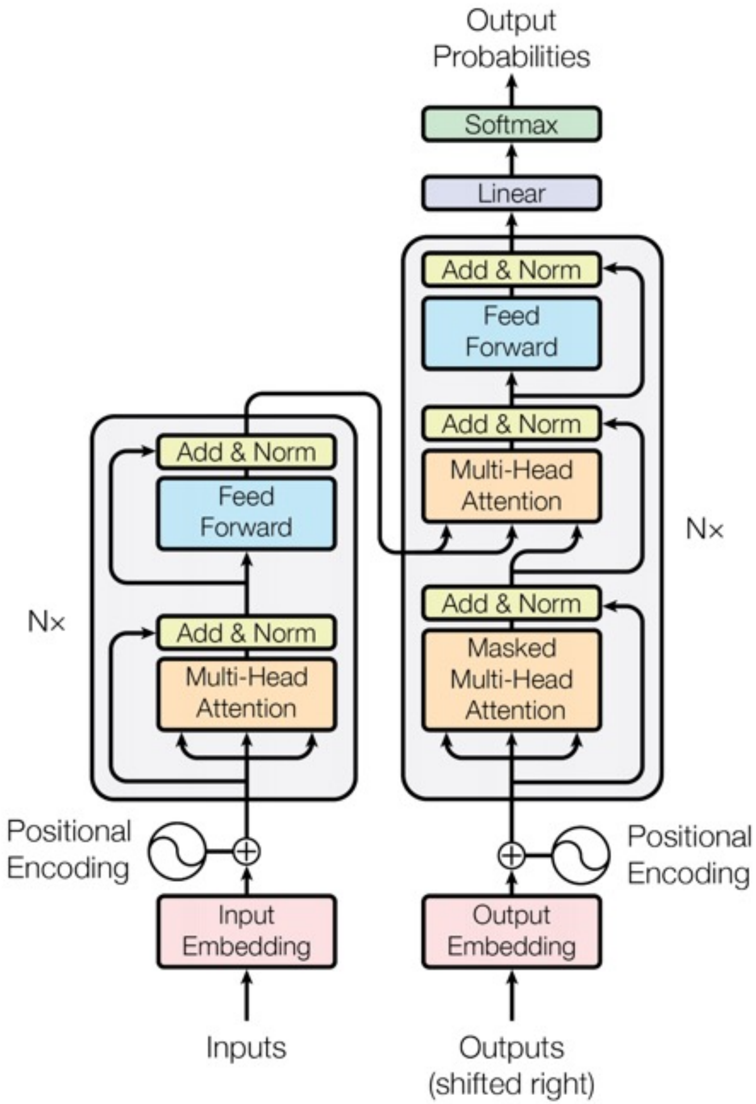
The QA Hierarchy of Needs



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)

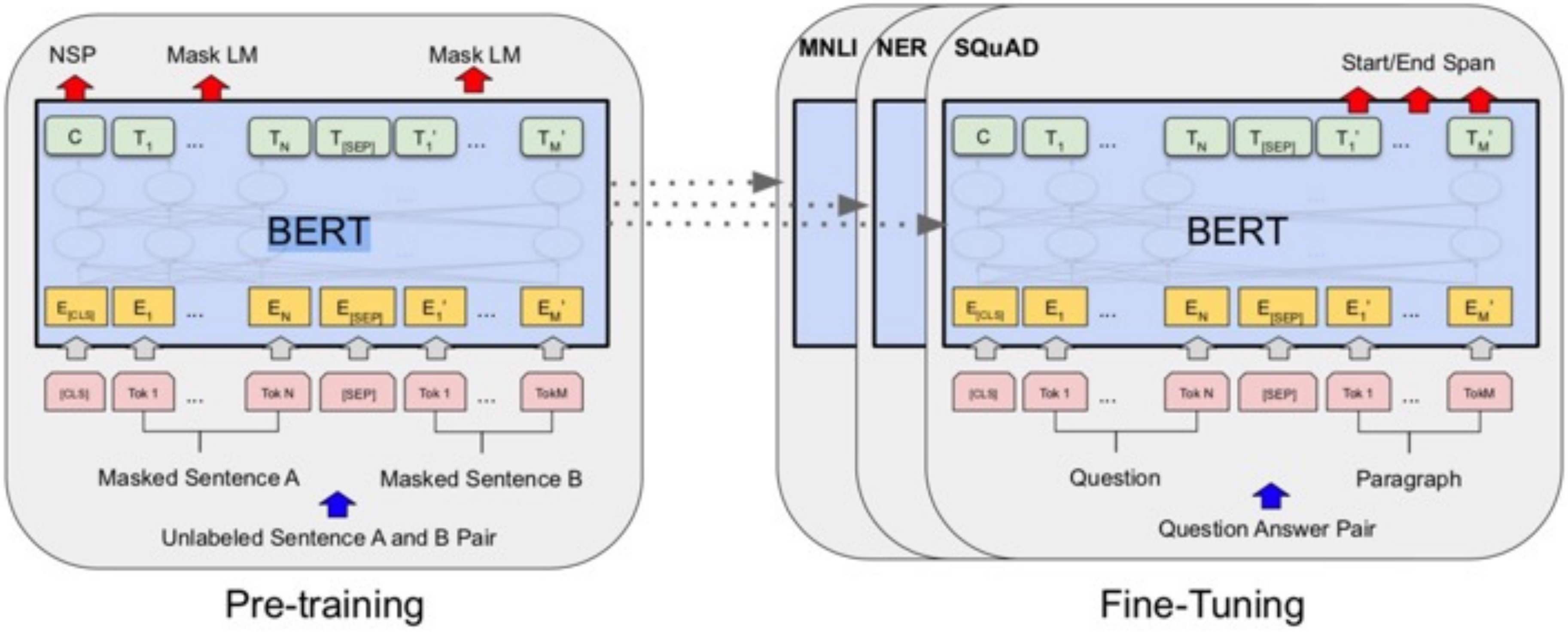


Source: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

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

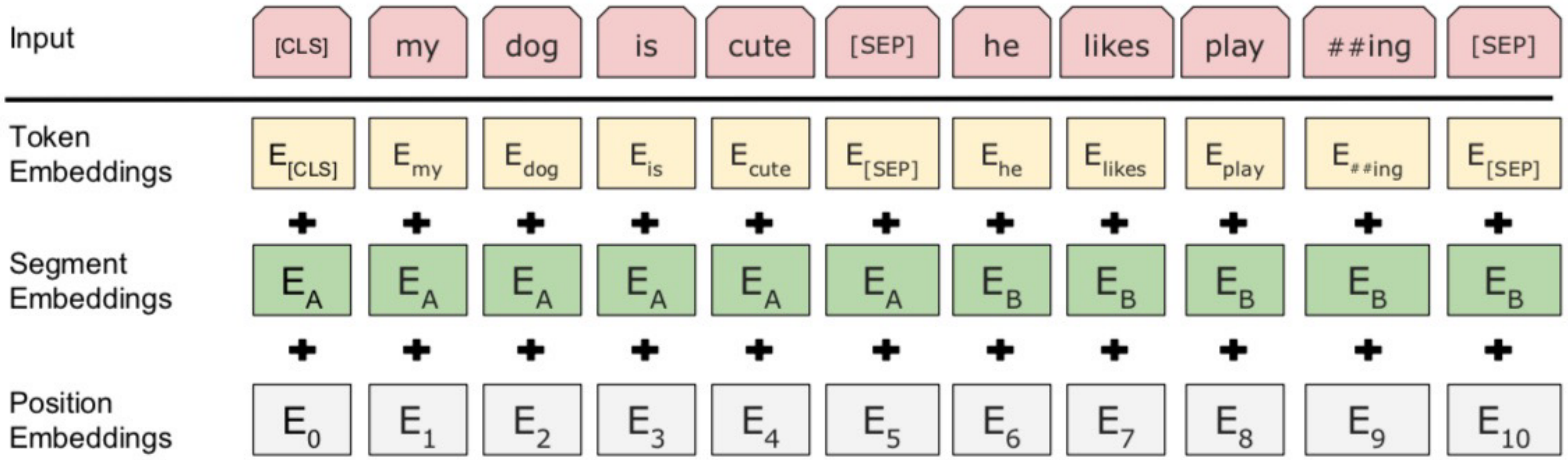


Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

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

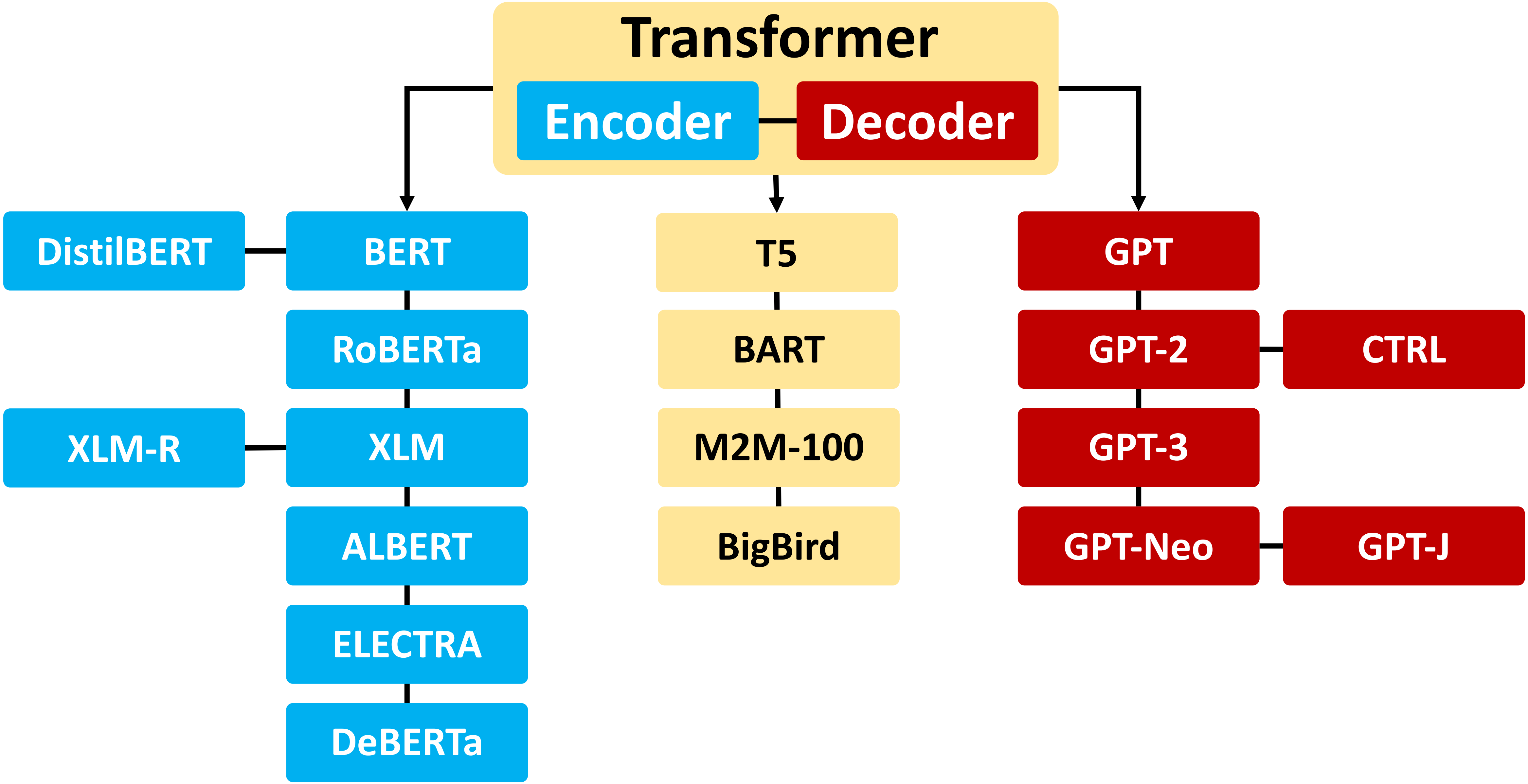
BERT (Bidirectional Encoder Representations from Transformers)

BERT input representation



Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

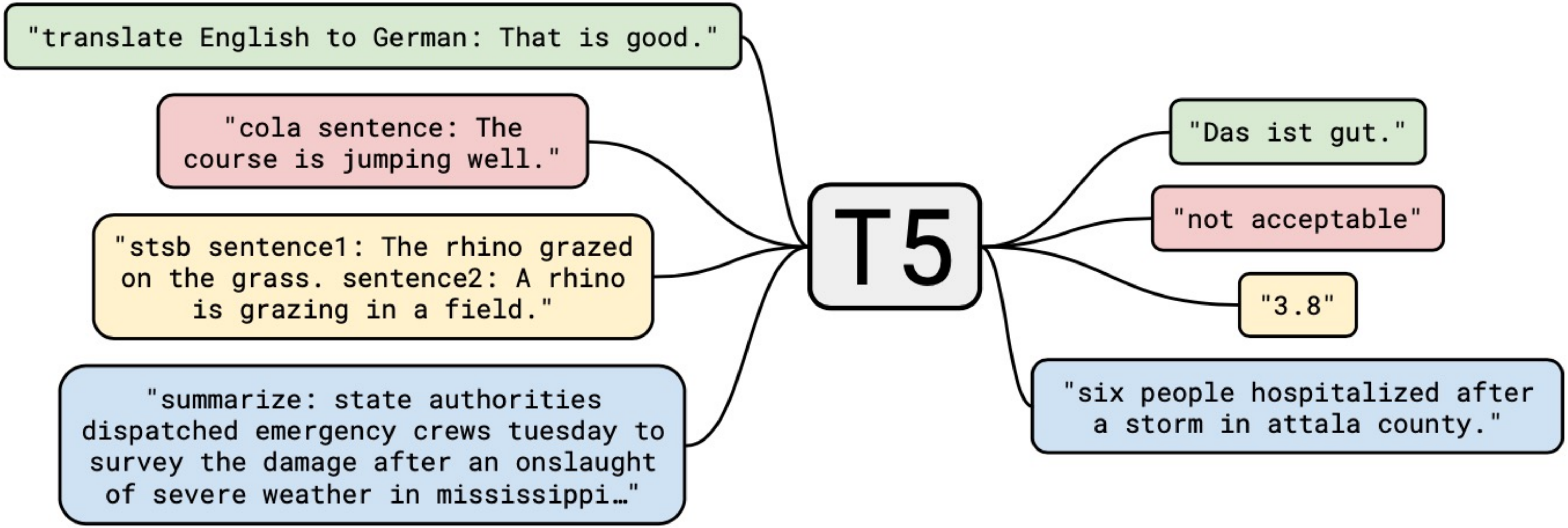
Transformer Models



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

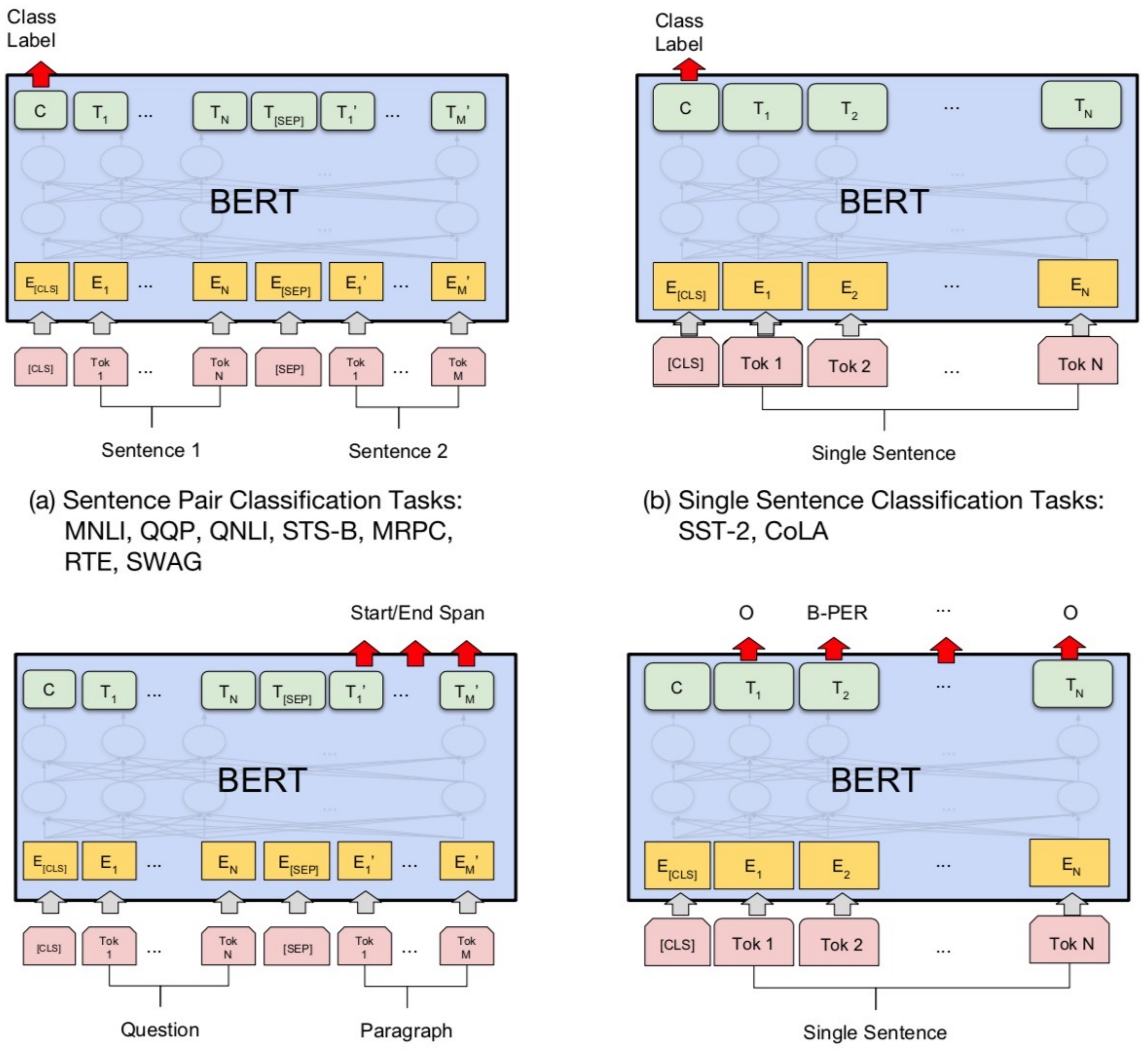
T5

Text-to-Text Transfer Transformer



Source: JColin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2019). "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).

Fine-tuning BERT on Different Tasks



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

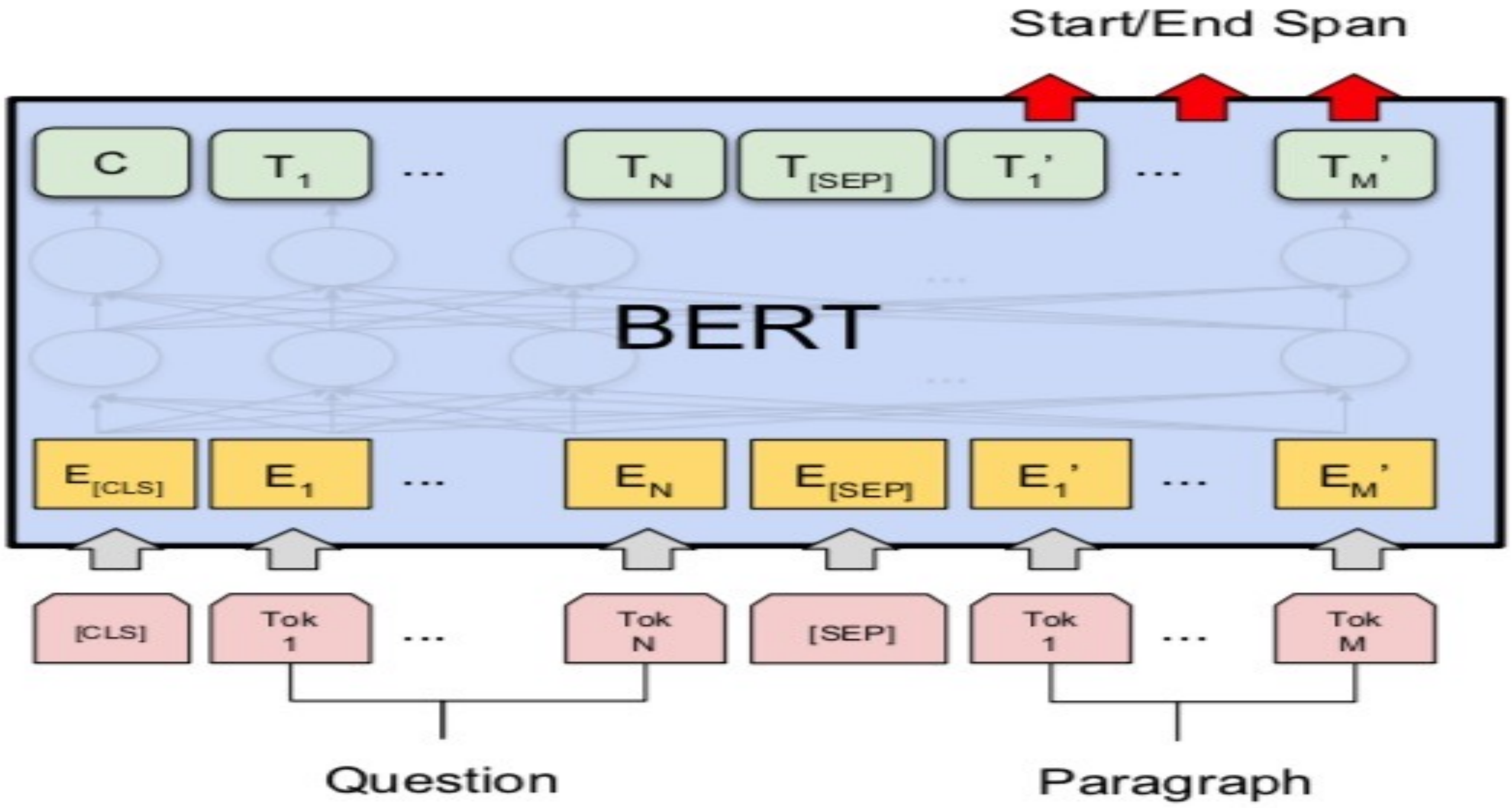
(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Question Answering (QA)

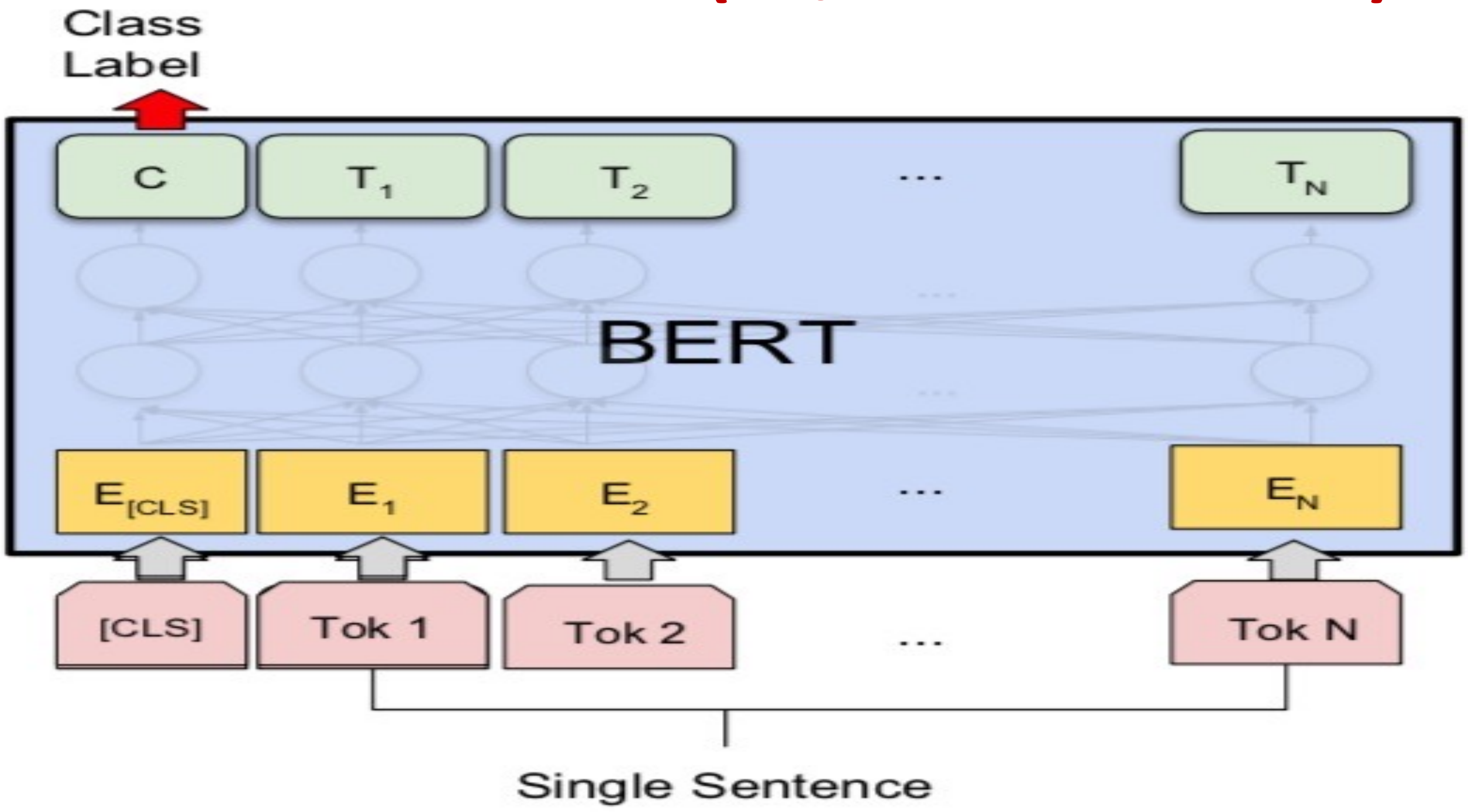


(c) Question Answering Tasks: SQuAD v1.1

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

Intent Detection (ID; Classification)

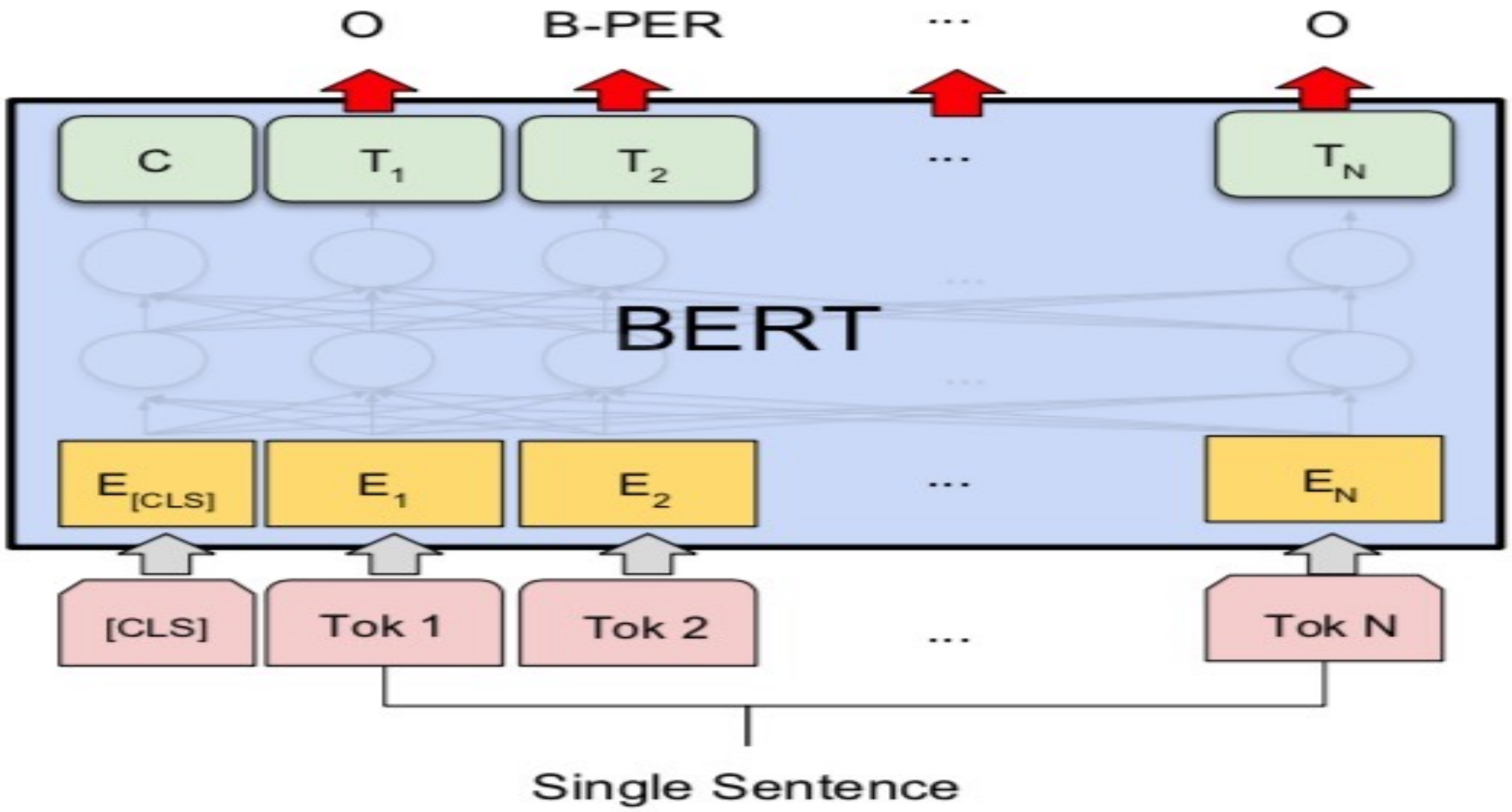


(b) Single Sentence Classification Tasks: SST-2, CoLA

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

Slot Filling (SF)



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

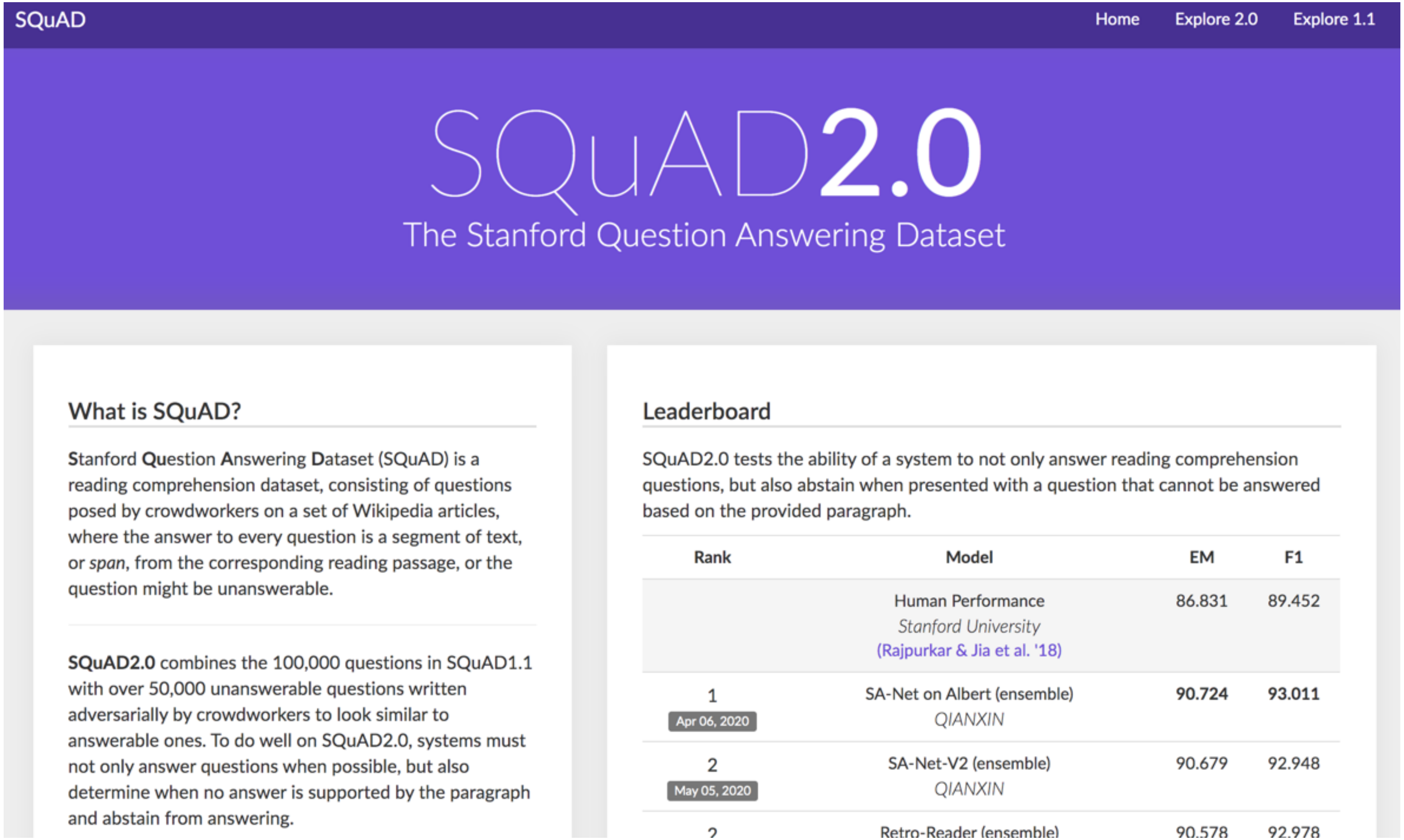
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Question Answering (QA)

SQuAD

Stanford Question Answering Dataset

SQuAD



SQuAD Home Explore 2.0 Explore 1.1

SQuAD2.0

The Stanford Question Answering Dataset

What is SQuAD?

Stanford **Q**uestion **A**nswering **D**ataset (SQuAD) 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.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	**90.724**	**93.011**
2 May 05, 2020	SA-Net-V2 (ensemble) *QIANXIN*	90.679	92.948
2	Retro-Reader (ensemble)	90.578	92.978

https://rajpurkar.github.io/SQuAD-explorer/

SQuAD

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, leaning 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, snow, 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

Source: Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang.
"Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).

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.[2] 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."[3]

https://en.wikipedia.org/wiki/Precipitation

SQuAD (Question Answering)

Paragraph

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?

SQuAD (Question Answering)

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

SQuAD (Question Answering)

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

SQuAD (Question Answering)

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

SQuAD (Question Answering)

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

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM
Disk

Editing

Question Answering

[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 gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
gravity
```

[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 gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
graupel
```

```
#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 gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
within a cloud
```

https://tinyurl.com/aintpupython101

Question Answering

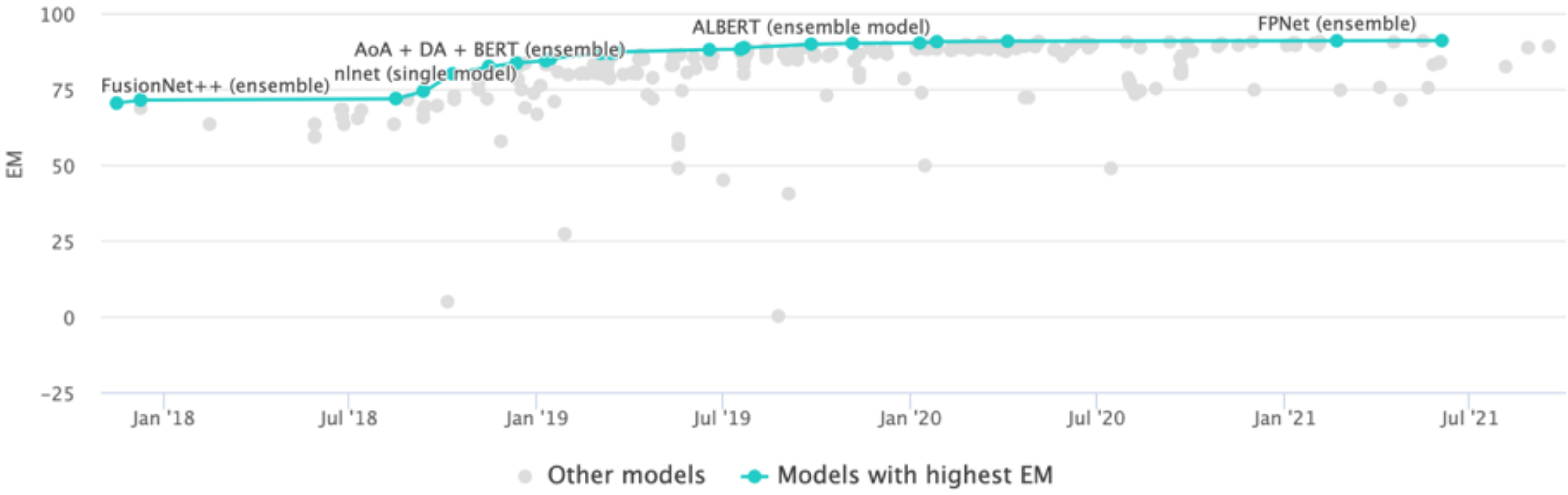
```
from transformers import pipeline
qamodel = pipeline("question-answering", model ='deepset/roberta-base-squad2')
question = "What causes precipitation to fall?"
context = """In meteorology, precipitation is any product of
the condensation of atmospheric water vapor that falls under
gravity. The main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

```
gravity
```

https://tinyurl.com/aintpupython101

Question Answering on SQuAD2.0

SQuAD 2.0 benchmark (Papers with Code)

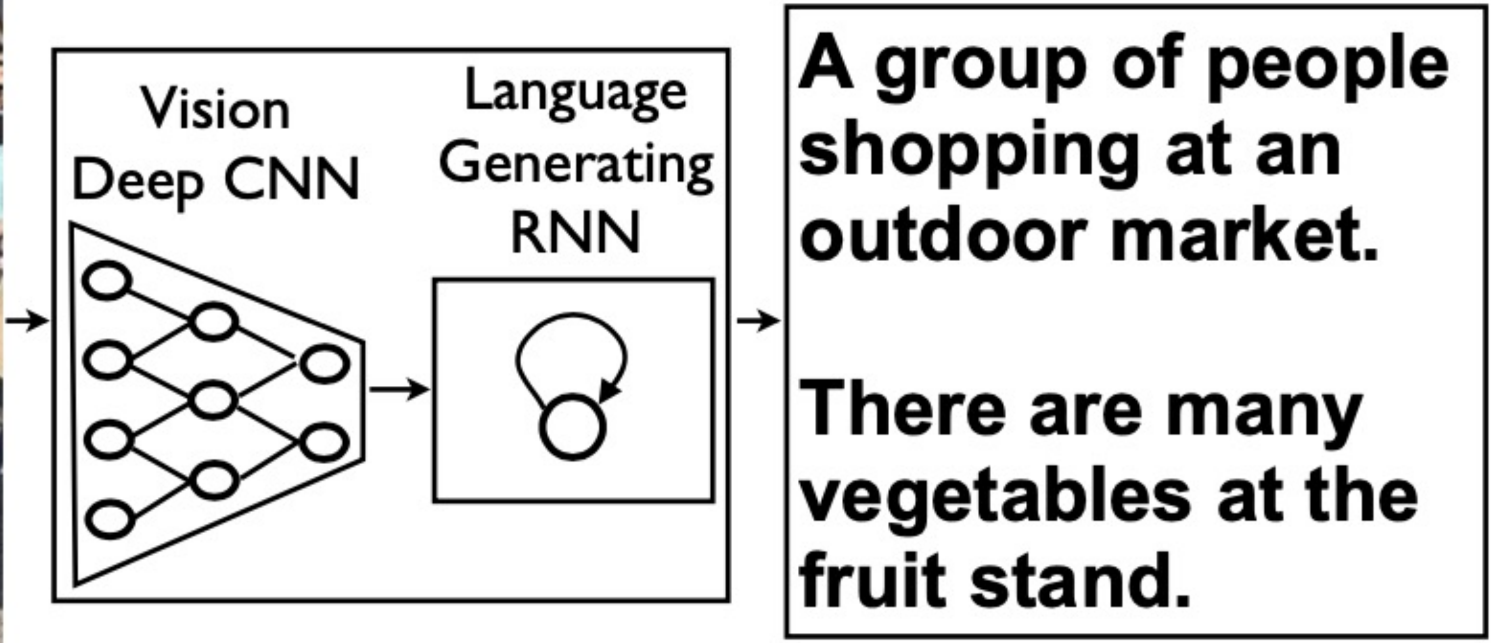


https://paperswithcode.com/sota/question-answering-on-squad20

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)

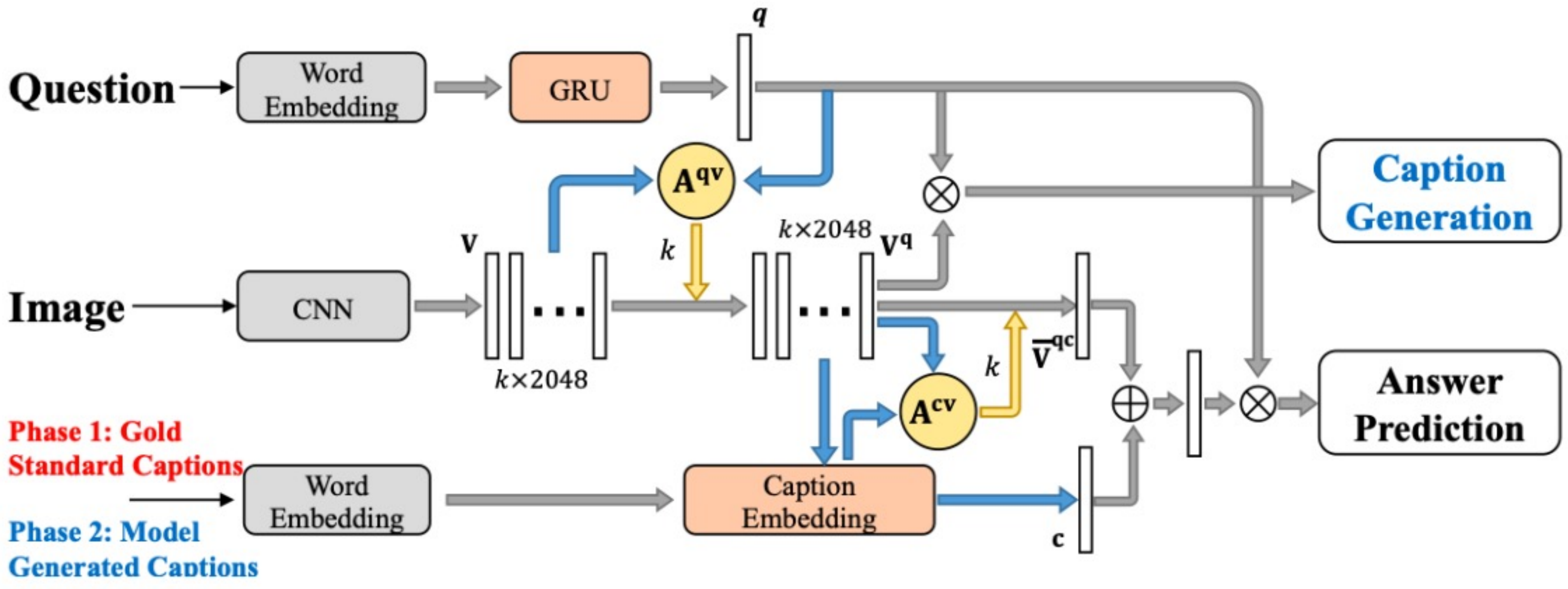
image-to-text description generation



Source: Erkut Erdem, Menekse Kuyu, Semih Yagcioglu, Anette Frank, Letitia Parcalabescu, Barbara Plank, Andrii Babii et al (2022). "Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning." Journal of Artificial Intelligence Research 73 (2022): 1131-1207.

Visual Question Answering

Neural caption generation is employed to aid answer prediction



Source: Erkut Erdem, Menekse Kuyu, Semih Yagcioglu, Anette Frank, Letitia Parcalabescu, Barbara Plank, Andrii Babii et al (2022). "Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning." Journal of Artificial Intelligence Research 73 (2022): 1131-1207.

Dialogue Systems

Conversational Commerce

Chatbot

Dialogue System

Intelligent Agent

Dialogue Subtasks

Browse SoTA > Natural Language Processing > Dialogue

Dialogue subtasks

Dialogue Generation

Dialogue Generation

9 benchmarks

78 papers with code

Dialogue State Tracking

2 benchmarks

51 papers with code

Task-Oriented Dialogue Systems

Task-Oriented Dialogue Systems

2 benchmarks

48 papers with code

Visual Dialog

8 benchmarks

37 papers with code

Goal-Oriented Dialog

1 benchmark

20 papers with code

Dialogue Management

12 papers with code

Dialogue Understanding

11 benchmarks

8 papers with code

Dialogue Act Classification

2 benchmarks

8 papers with code

Short-Text Conversation

Short-Text Conversation

7 papers with code

Goal-Oriented Dialogue Systems

7 papers with code

Source: https://paperswithcode.com/area/natural-language-processing/dialogue

Chatbots: Evolution of UI/UX

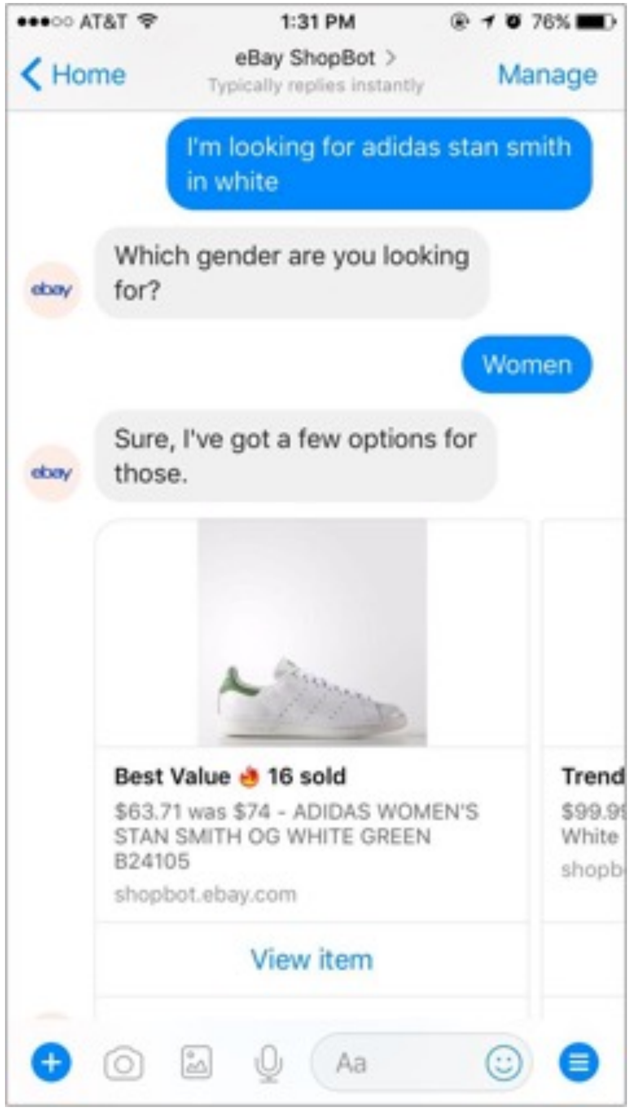
	mid - 80s	mid - 90s	mid - 00s	mid - 10s
Paradigm	**PC**	**Web**	**Smartphone**	**Messaging**
				
Platform *Examples*	**Desktop** *DOS, Windows, Mac OS*	**Browser** *Mosaic, Explorer, Chrome*	**Mobile OS** *iOS, Android*	**Messaging Apps** *WhatsApp, Messenger, Slack*
Applications *Examples*	**Clients**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

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

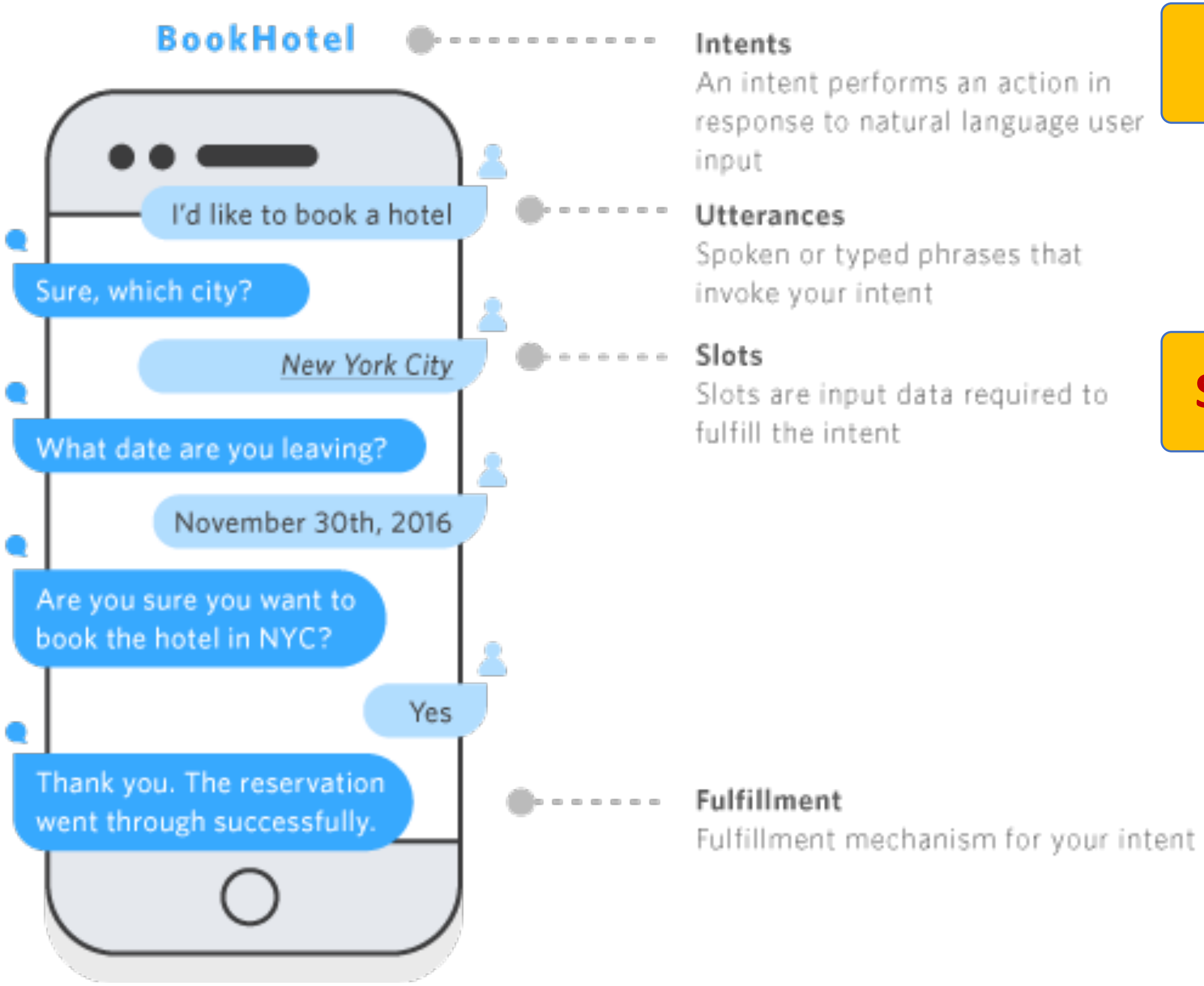
Source: http://www.guided-selling.org/from-e-commerce-to-conversational-commerce/

Conversational Commerce: eBay AI Chatbots



Source: https://www.forbes.com/sites/rachelarthur/2017/07/19/conversational-commerce-ebay-ai-chatbot/

Hotel Chatbot



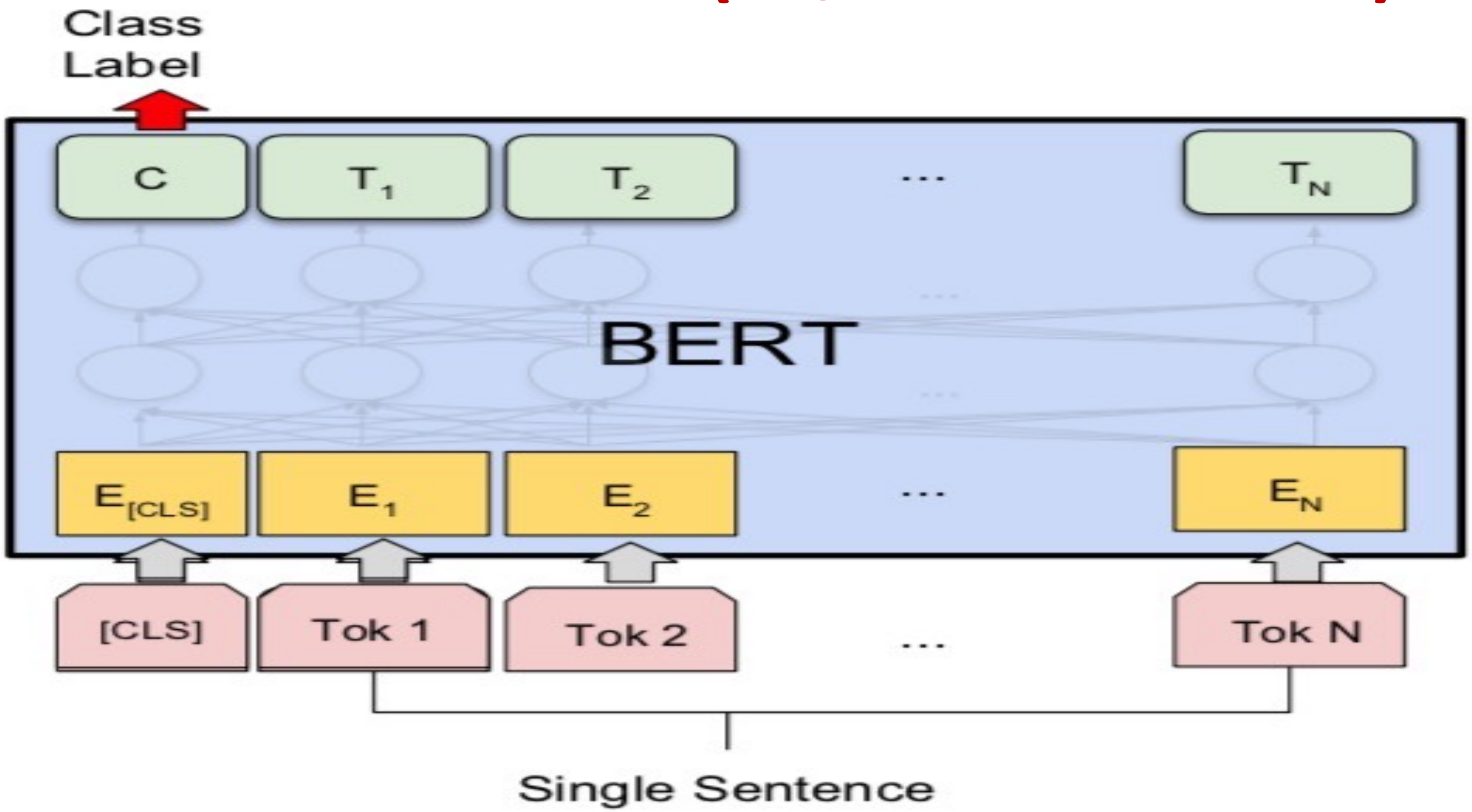
Intent Detection

Slot Filling

Source: https://sdtimes.com/amazon/guest-view-capitalize-amazon-lex-available-general-public/

Fine-tuning BERT on Dialogue

Intent Detection (ID; Classification)

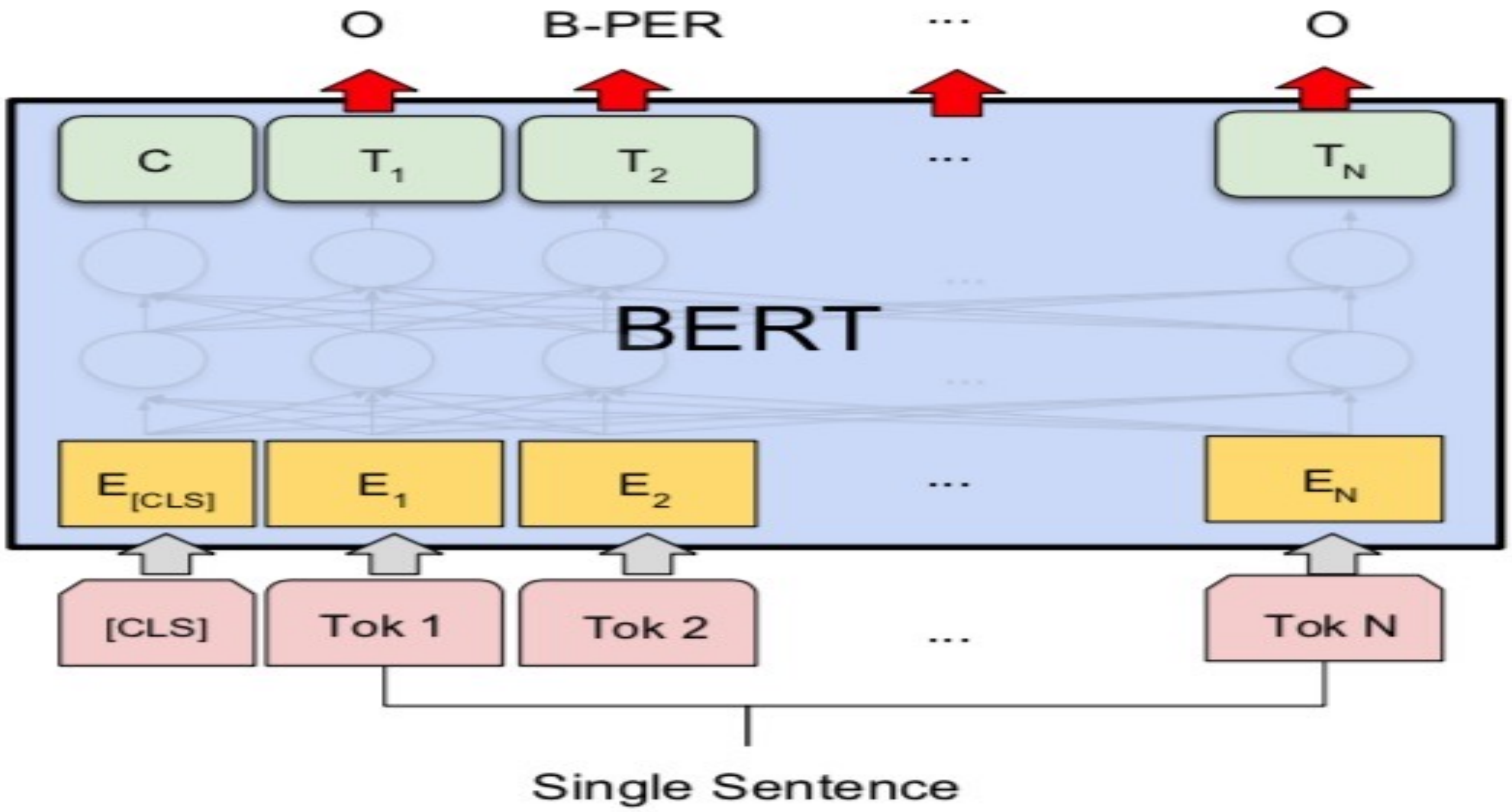


(b) Single Sentence Classification Tasks: SST-2, CoLA

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

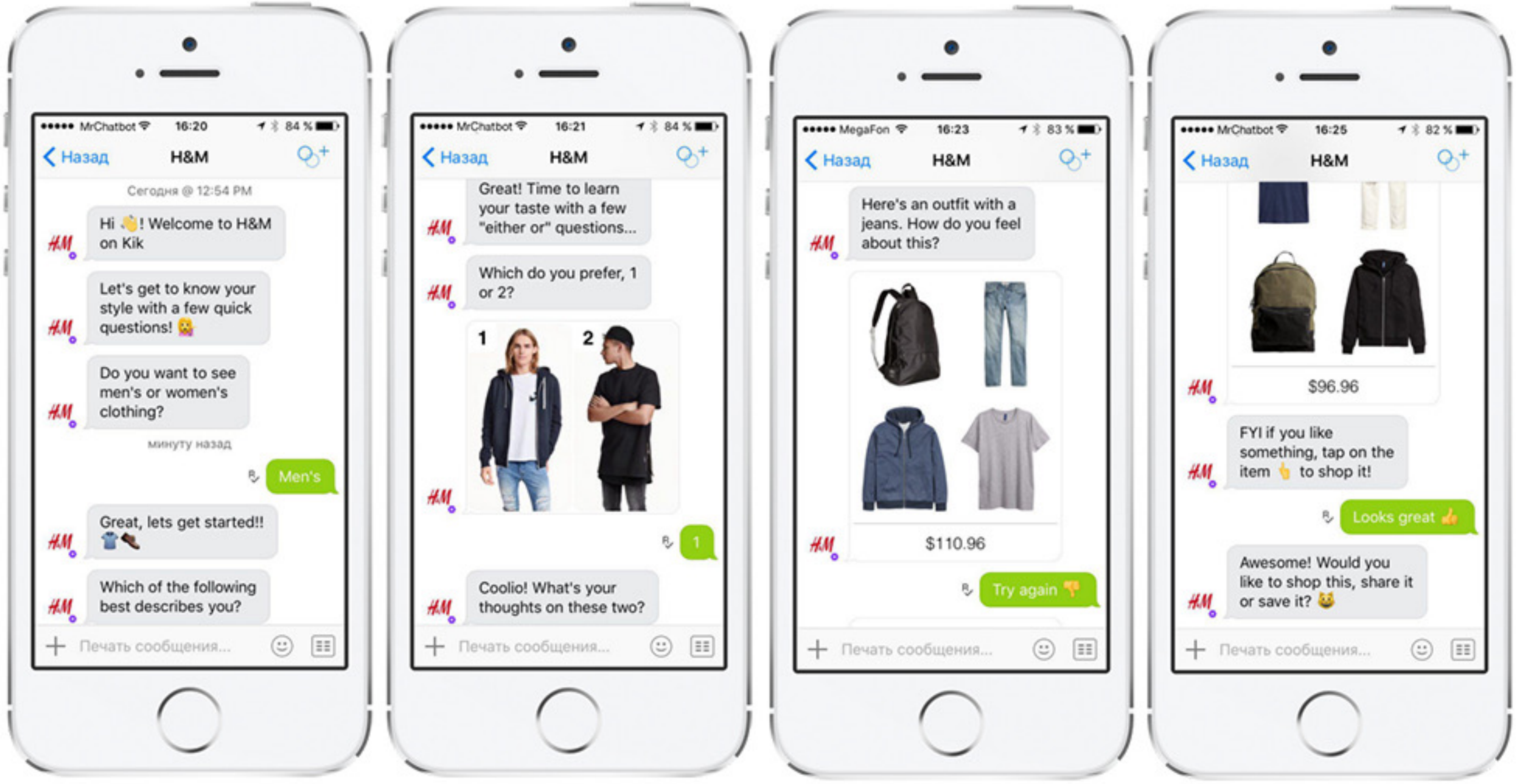
Slot Filling (SF)



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

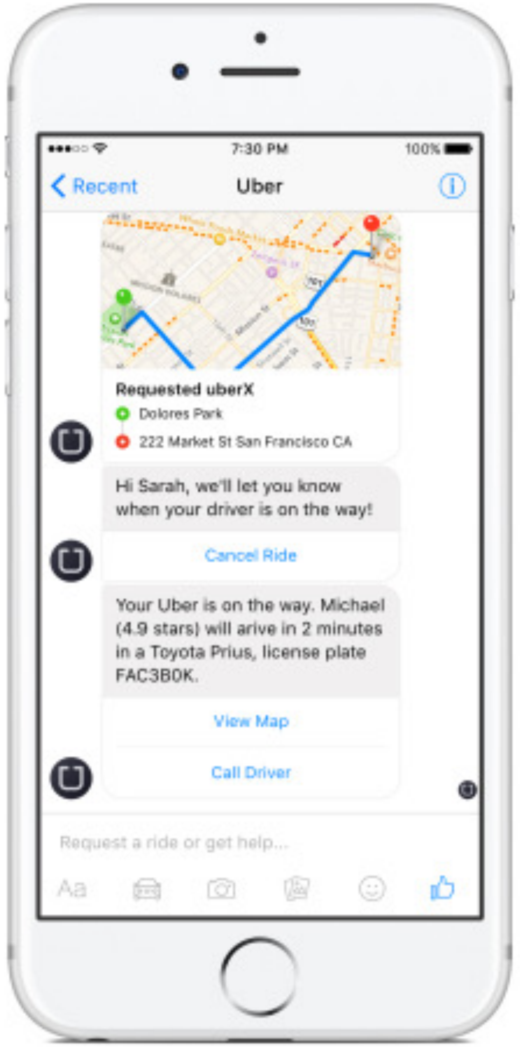
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

H&M's Chatbot on Kik



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

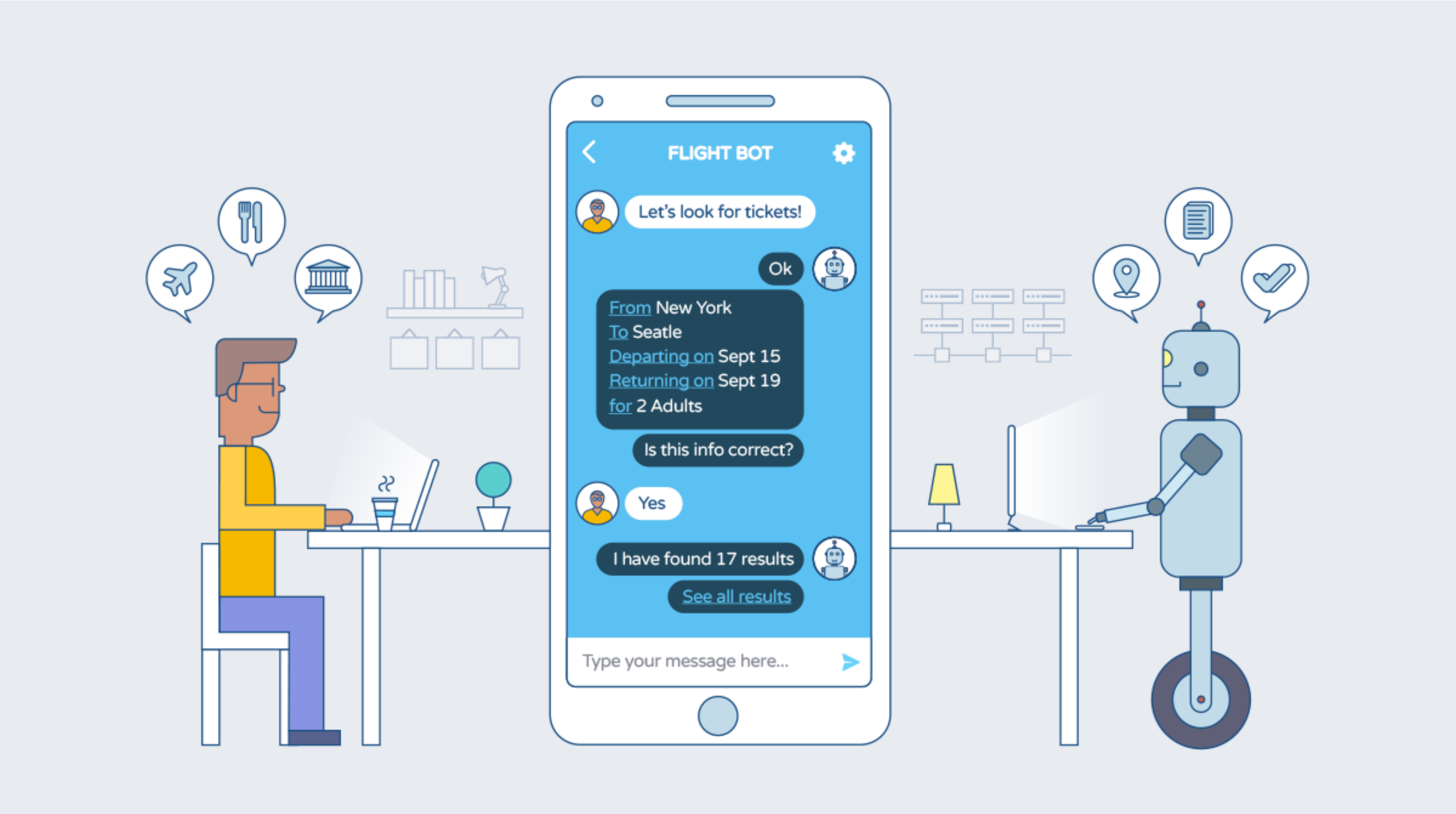
Uber’s Chatbot on Facebook’s Messenger



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

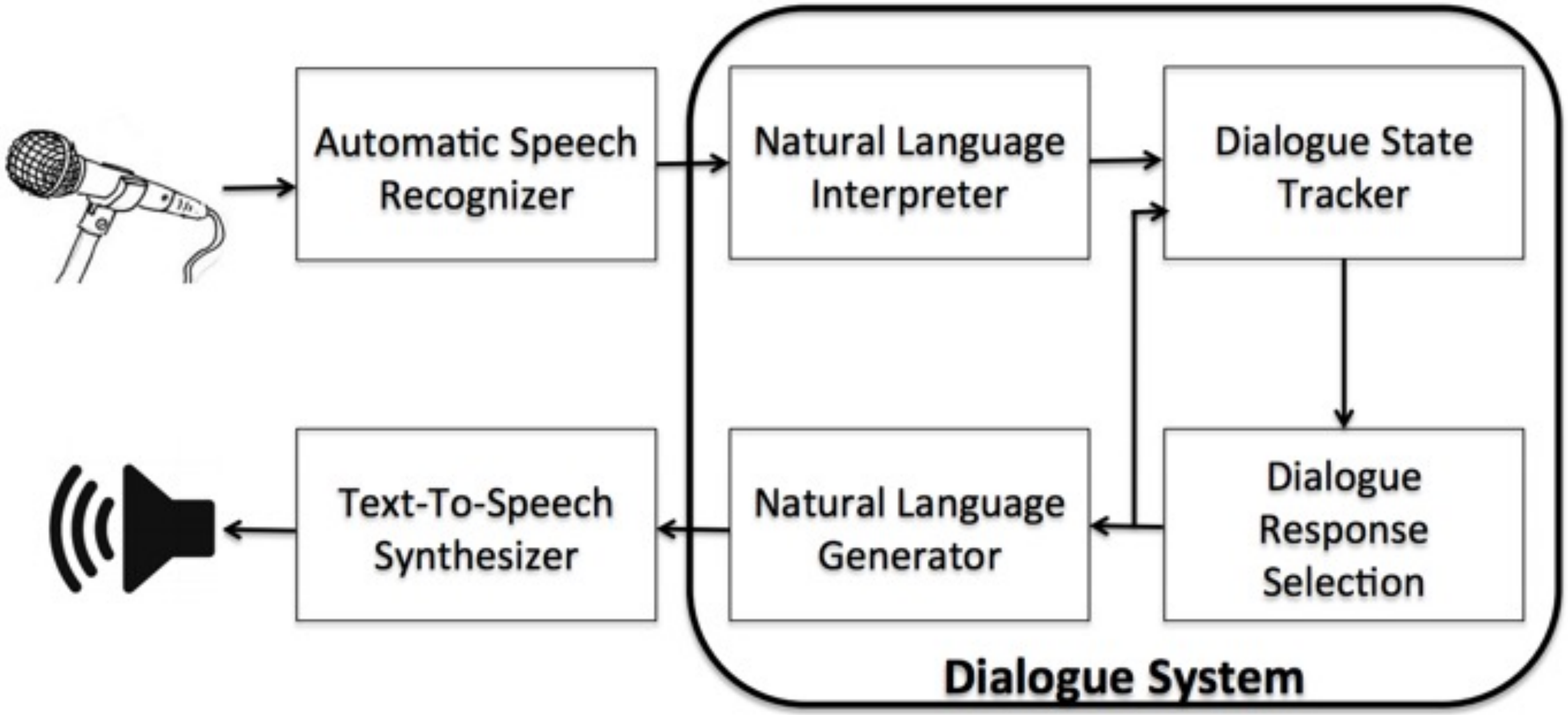
Source: http://www.guided-selling.org/from-e-commerce-to-conversational-commerce/

Chatbot



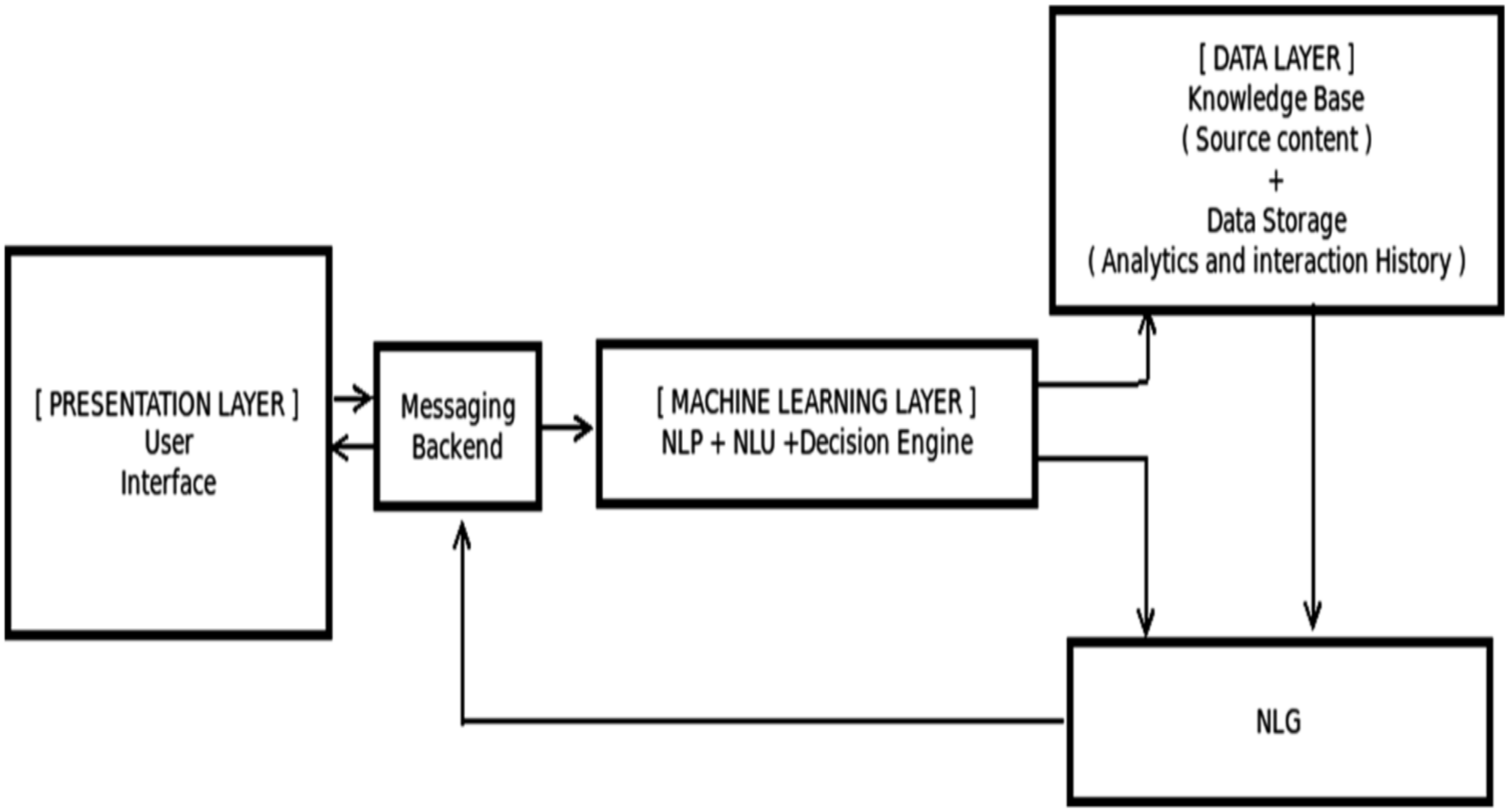
Source: https://www.mdsdecoded.com/blog/the-rise-of-chatbots/

Dialogue System



Source: Serban, I. V., Lowe, R., Charlin, L., & Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.

Overall Architecture of Intelligent Chatbot



Source: Borah, Bhriguraj, Dhrubajyoti Pathak, Priyankoo Sarmah, Bidisha Som, and Sukumar Nandi. "Survey of Textbased Chatbot in Perspective of Recent Technologies." In International Conference on Computational Intelligence, Communications, and Business Analytics, pp. 84-96. Springer, Singapore, 2018.

Can machines think?

(Alan Turing ,1950)

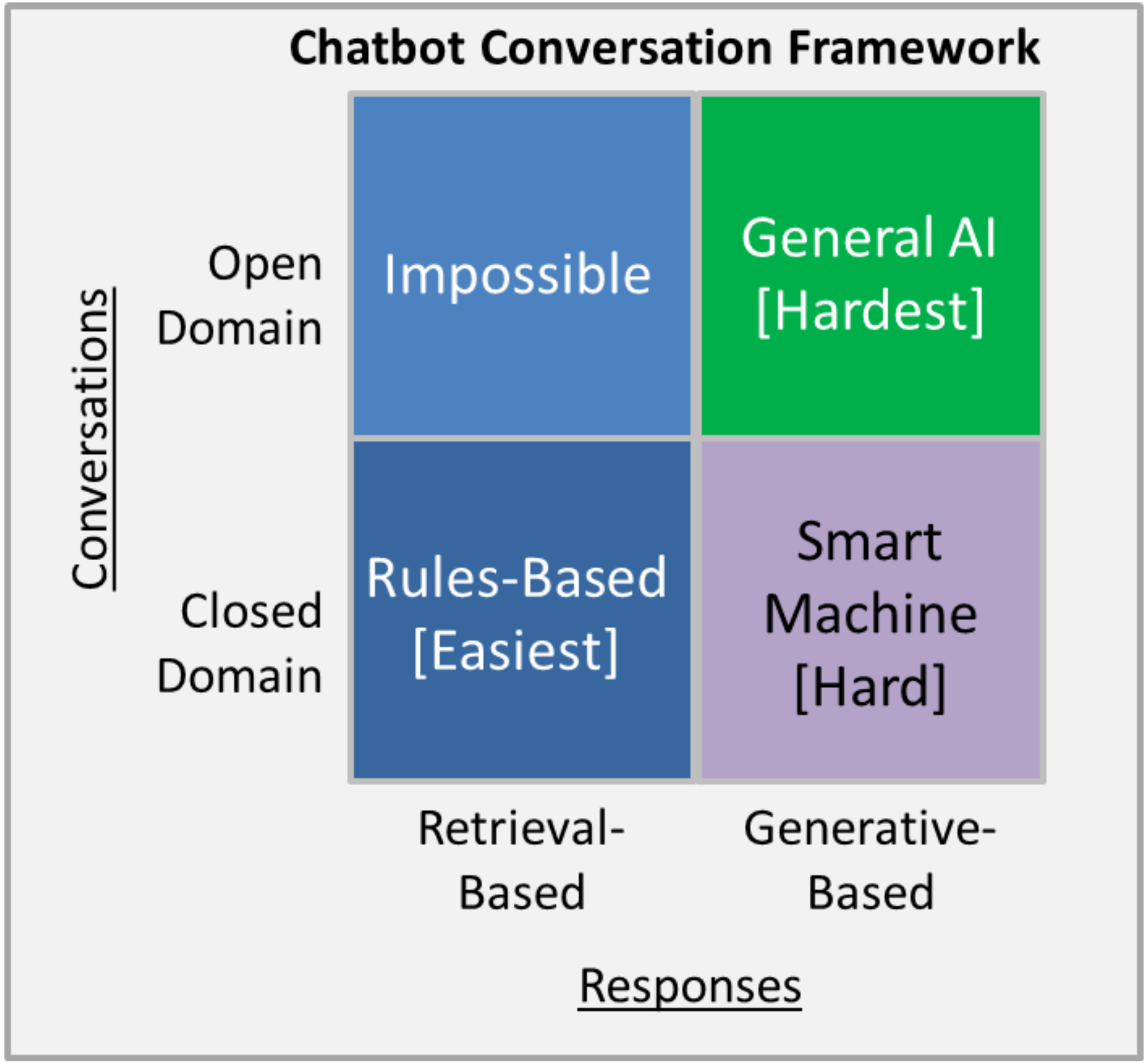
Source: Cahn, Jack. "CHATBOT: Architecture, Design, & Development." PhD diss., University of Pennsylvania, 2017.

Chatbot

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

Source: Cahn, Jack. "CHATBOT: Architecture, Design, & Development." PhD diss., University of Pennsylvania, 2017.

Chatbot Conversation Framework

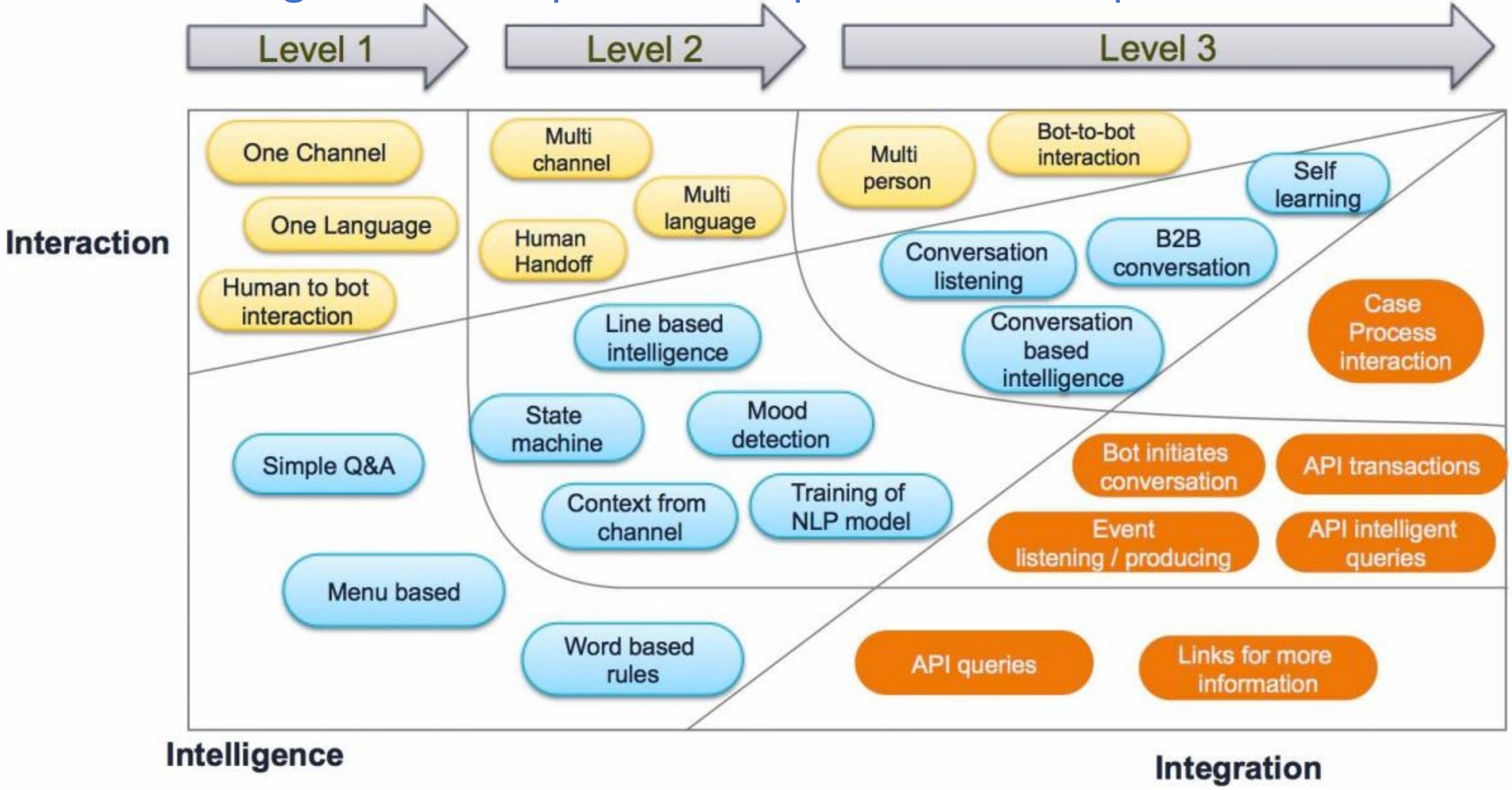


Source: https://chatbotslife.com/ultimate-guide-to-leveraging-nlp-machine-learning-for-you-chatbot-531ff2dd870c

Chatbots

Bot Maturity Model

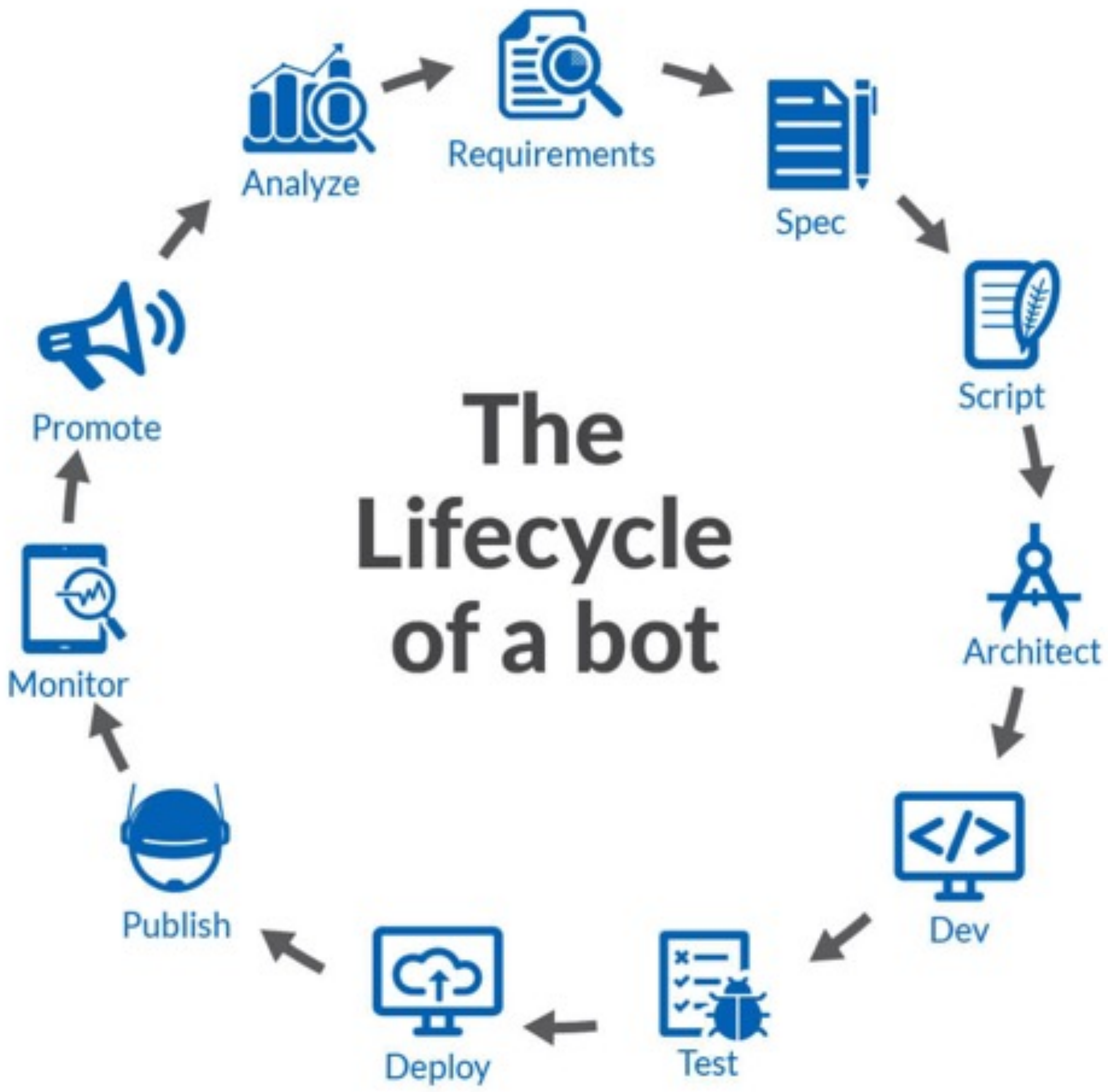
Customers want to have simpler means to interact with businesses and get faster response to a question or complaint.



Source: https://www.capgemini.com/2017/04/how-can-chatbots-meet-expectations-introducing-the-bot-maturity/

Bot Life Cycle and Platform Ecosystem

The Bot Lifecycle

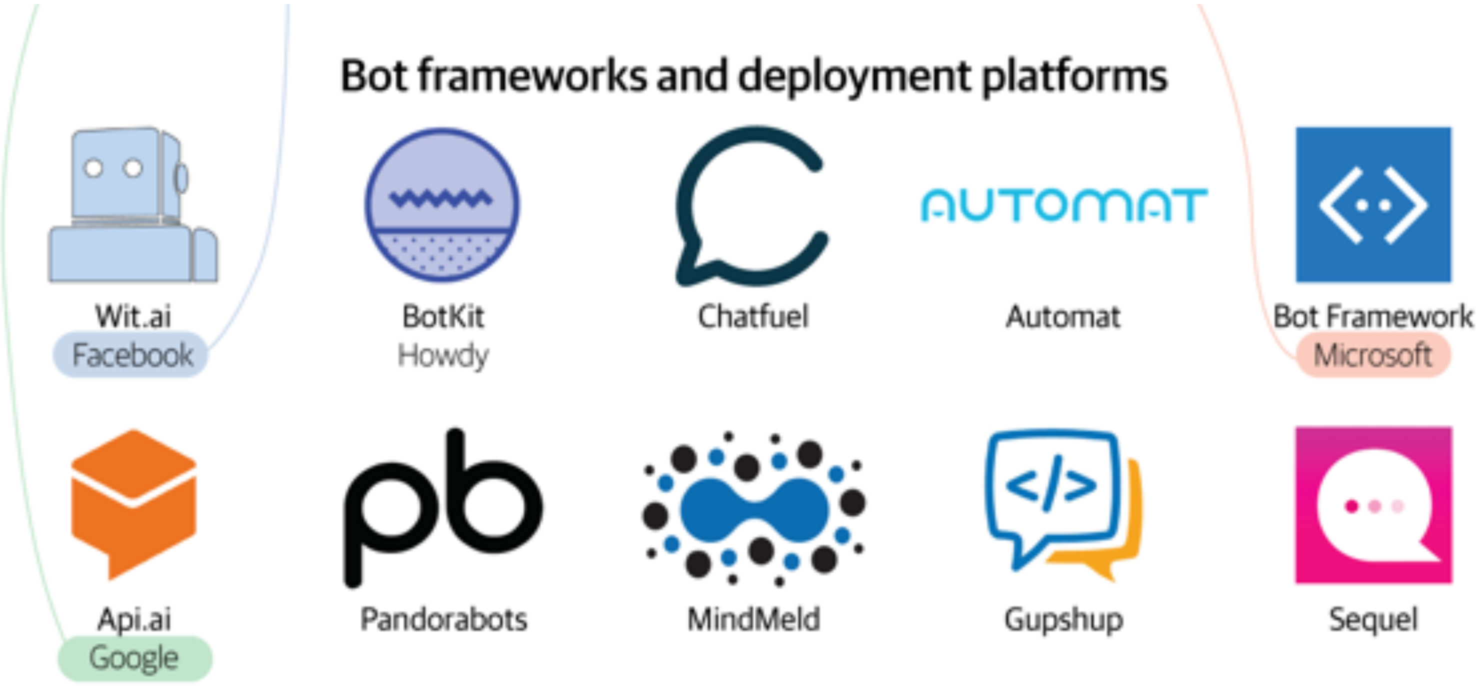


Source: https://chatbotsmagazine.com/the-bot-lifecycle-1ff357430db7

The bot platform ecosystem
and the emerging giants

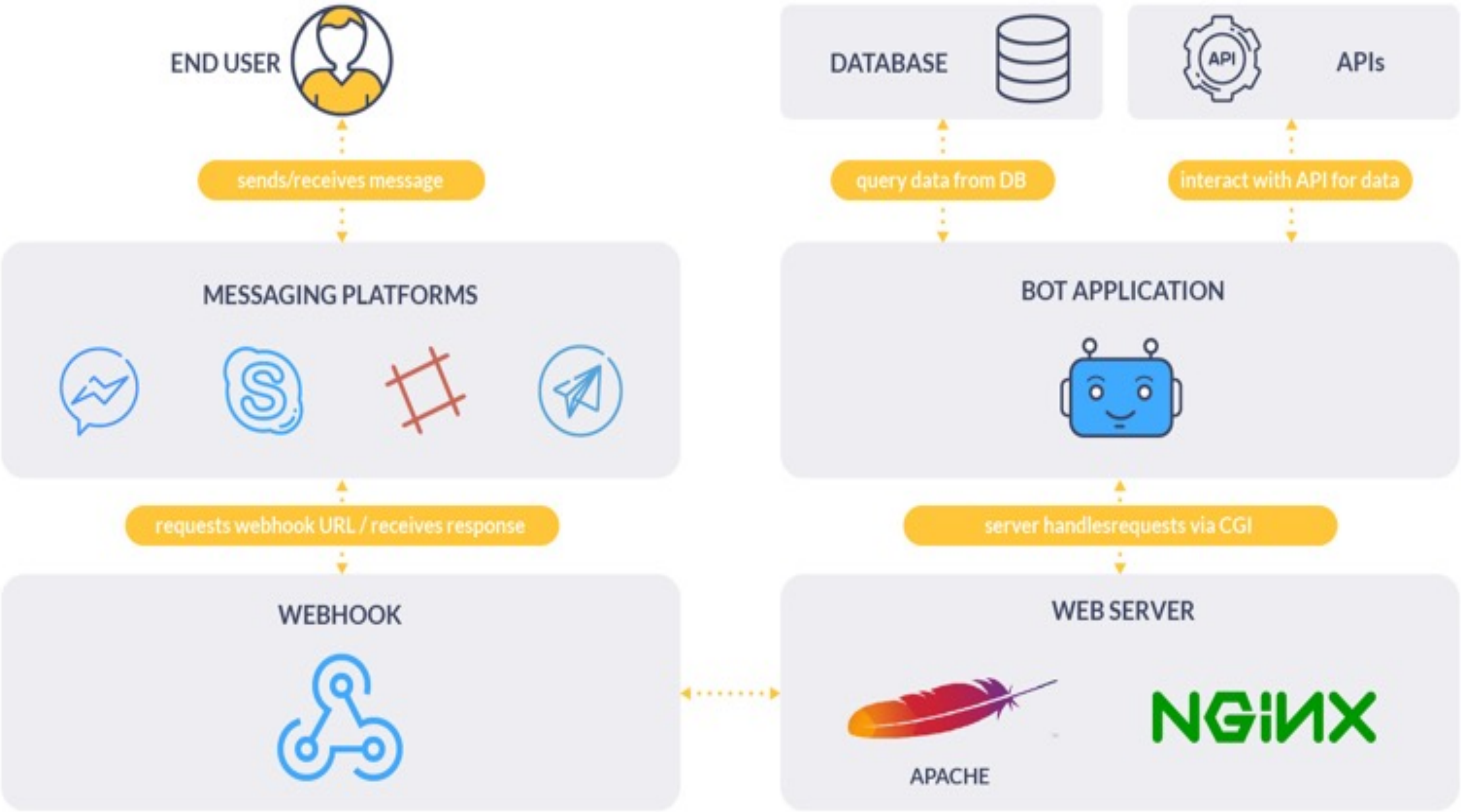
Nearly every large software company has announced some sort of bot strategy in the last year. Here's a look at a handful of leading platforms that developers might use to send messages, interpret natural language, and deploy bots, with the emerging bot-ecosystem giants highlighted.

Source: https://www.oreilly.com/ideas/infographic-the-bot-platform-ecosystem



Source: https://www.oreilly.com/ideas/infographic-the-bot-platform-ecosystem

How to Build Chatbots



Source: Igor Bobriakov (2018), https://activewizards.com/blog/a-comparative-analysis-of-chatbots-apis/

Chatbot Frameworks and AI Services

- Bot Frameworks
  - Botkit
  - Microsoft Bot Framework
  - Rasa NLU
- AI Services
  - Wit.ai
  - api.ai
  - LUIS.ai
  - IBM Watson

Source: Igor Bobriakov (2018), https://activewizards.com/blog/a-comparative-analysis-of-chatbots-apis/

Chatbot Frameworks

Comparison Table of Most Prominent Bot Frameworks

	Botkit	Microsoft Bot Framework	RASA NLU
Built-in integration with messaging platforms	✓	✓	✗
NLP support	✗ but possible to integrate with middlewares	✗ but have close bonds with LUIS.ai	✓
Out-of-box bots ready to be deployed	✓	✗	✗
Programming Language	JavaScript (Node)	JavaScript (Node), C#	Python

Created by ActiveWizards

Source: Igor Bobriakov (2018), https://activewizards.com/blog/a-comparative-analysis-of-chatbots-apis/

Comparison of Most Prominent AI Services

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

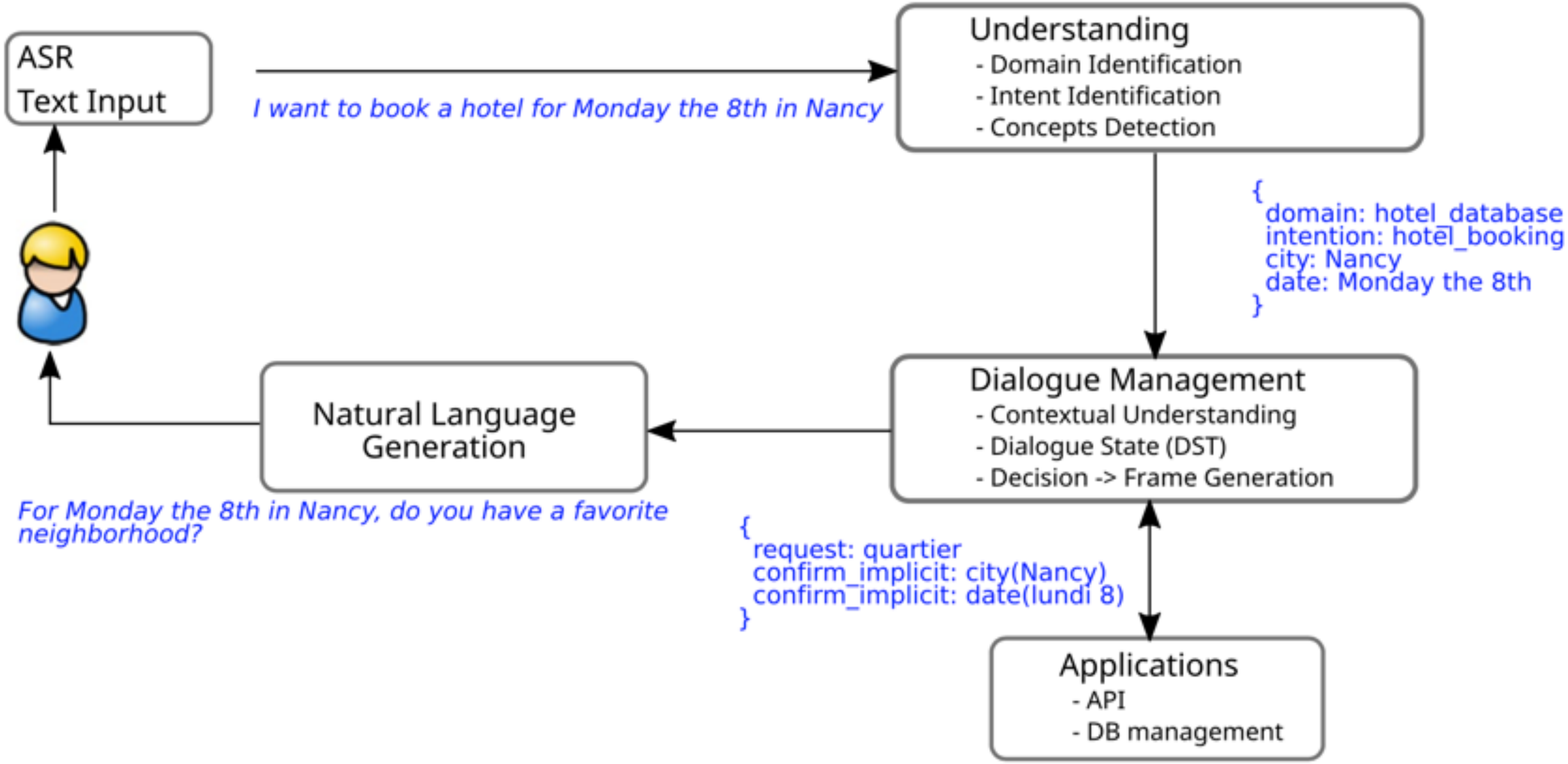
Created by ActiveWizards

Source: Igor Bobriakov (2018), https://activewizards.com/blog/a-comparative-analysis-of-chatbots-apis/

Task-Oriented Dialogue System

Task-Oriented Dialogue System

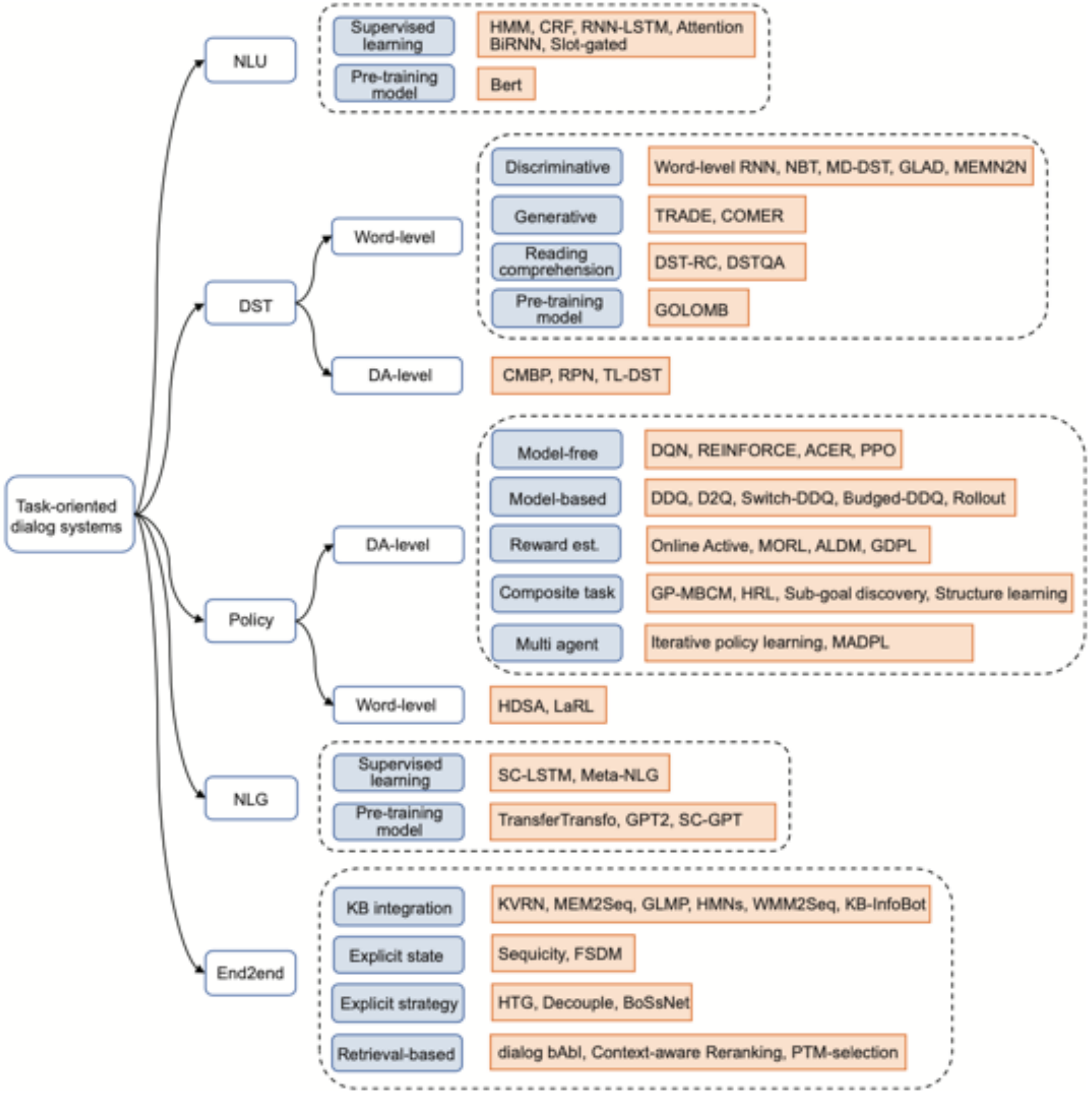
(Deriu et al., 2021)



Source: Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.

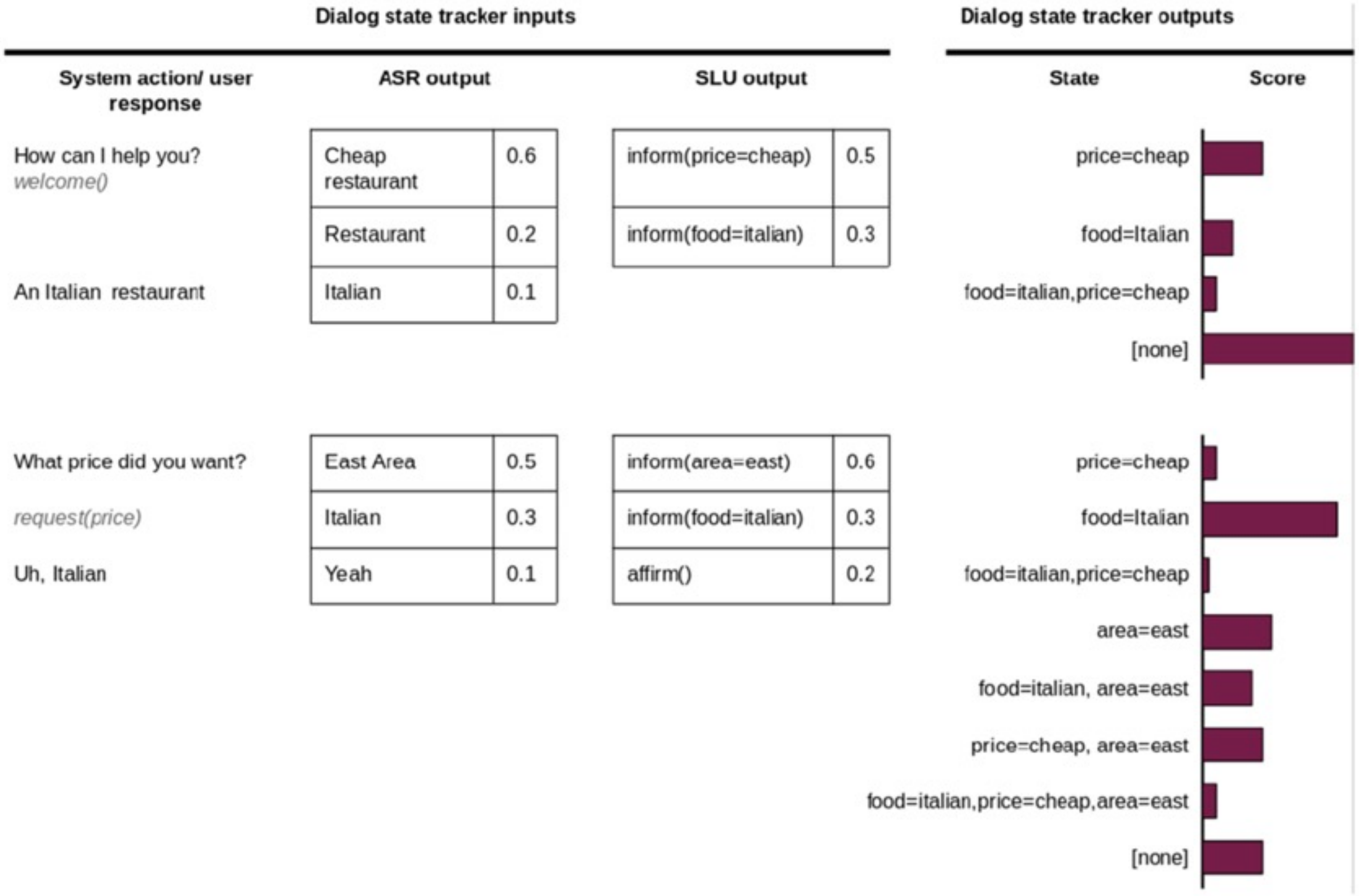
Task-Oriented Dialogue Systems

(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)



Source: Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.

Dialogue Acts

(Young et al., 2010)

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

Source: Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.

Sample Dialogue Acts

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

Source: Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.

Dialogue
on
Airline Travel
Information System
(ATIS)

The ATIS (Airline Travel Information System) Dataset

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

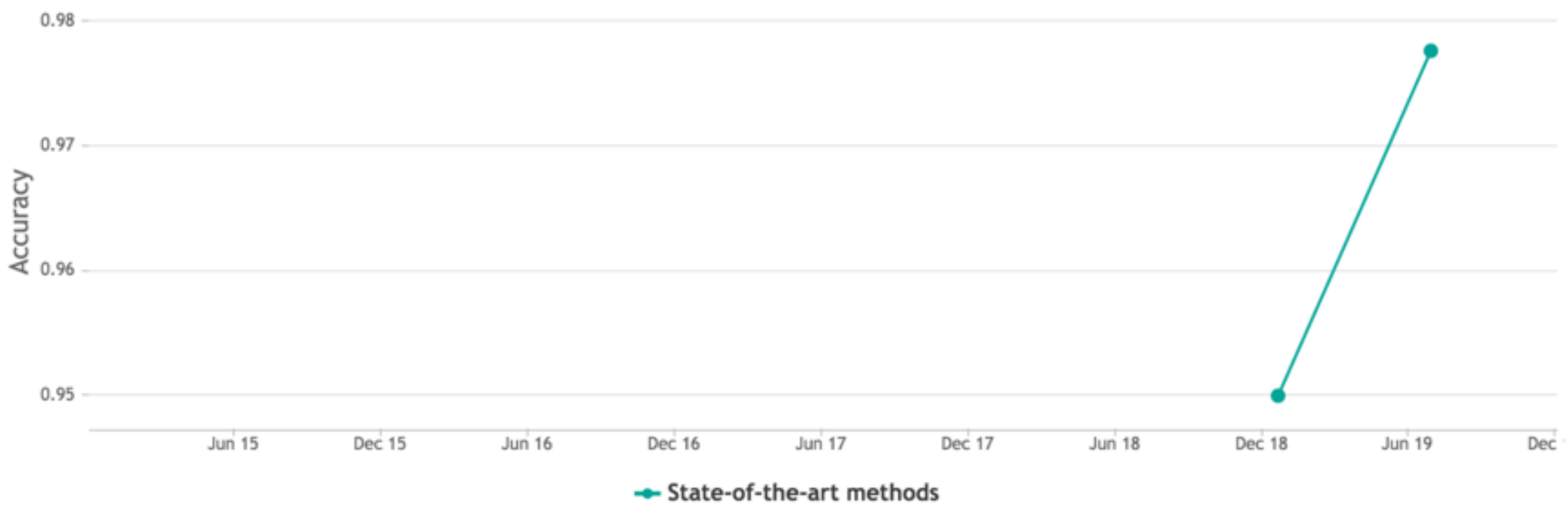
Sentence	what	flights	leave	from	phoenix
Slots	O	O	O	O	B-fromloc
Intent	atis_flight				

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

Source: Haihong, E., Peiqing Niu, Zhongfu Chen, and Meina Song. "A novel bi-directional interrelated model for joint intent detection and slot filling." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5467-5471. 2019.

Intent Detection on ATIS State-of-the-art

Intent Detection on ATIS

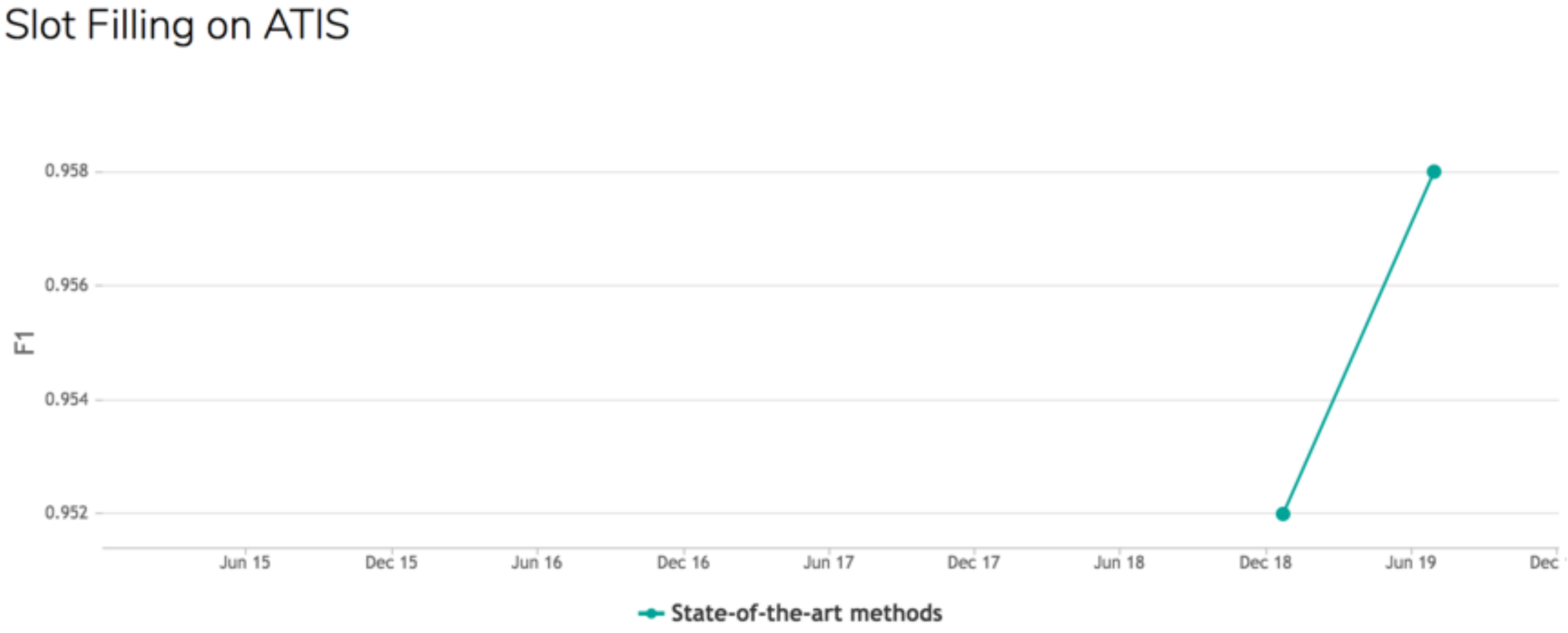


Edit

RANK	METHOD	ACCURACY	PAPER TITLE	YEAR	PAPER	CODE
1	SF-ID	0.9776	A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling	2019		
2	Capsule-NLU	0.950	Joint Slot Filling and Intent Detection via Capsule Neural Networks	2018		

Source: https://paperswithcode.com/sota/intent-detection-on-atis

Slot Filling on ATIS
State-of-the-art



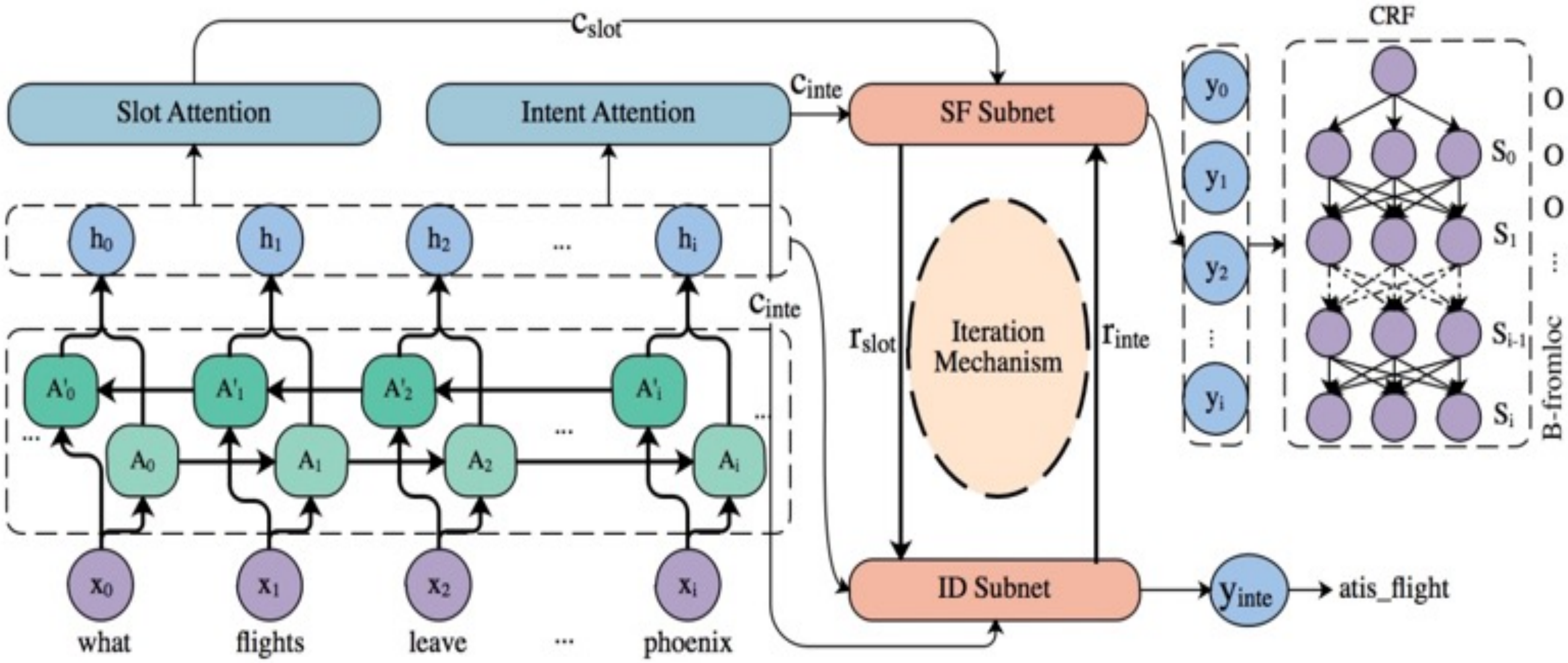
Edit

RANK	METHOD	F1	PAPER TITLE	YEAR	PAPER	CODE
1	**SF-ID**	0.958	A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling	2019		
2	**Capsule-NLU**	0.952	Joint Slot Filling and Intent Detection via Capsule Neural Networks	2018		

Source: https://paperswithcode.com/sota/slot-filling-on-atis

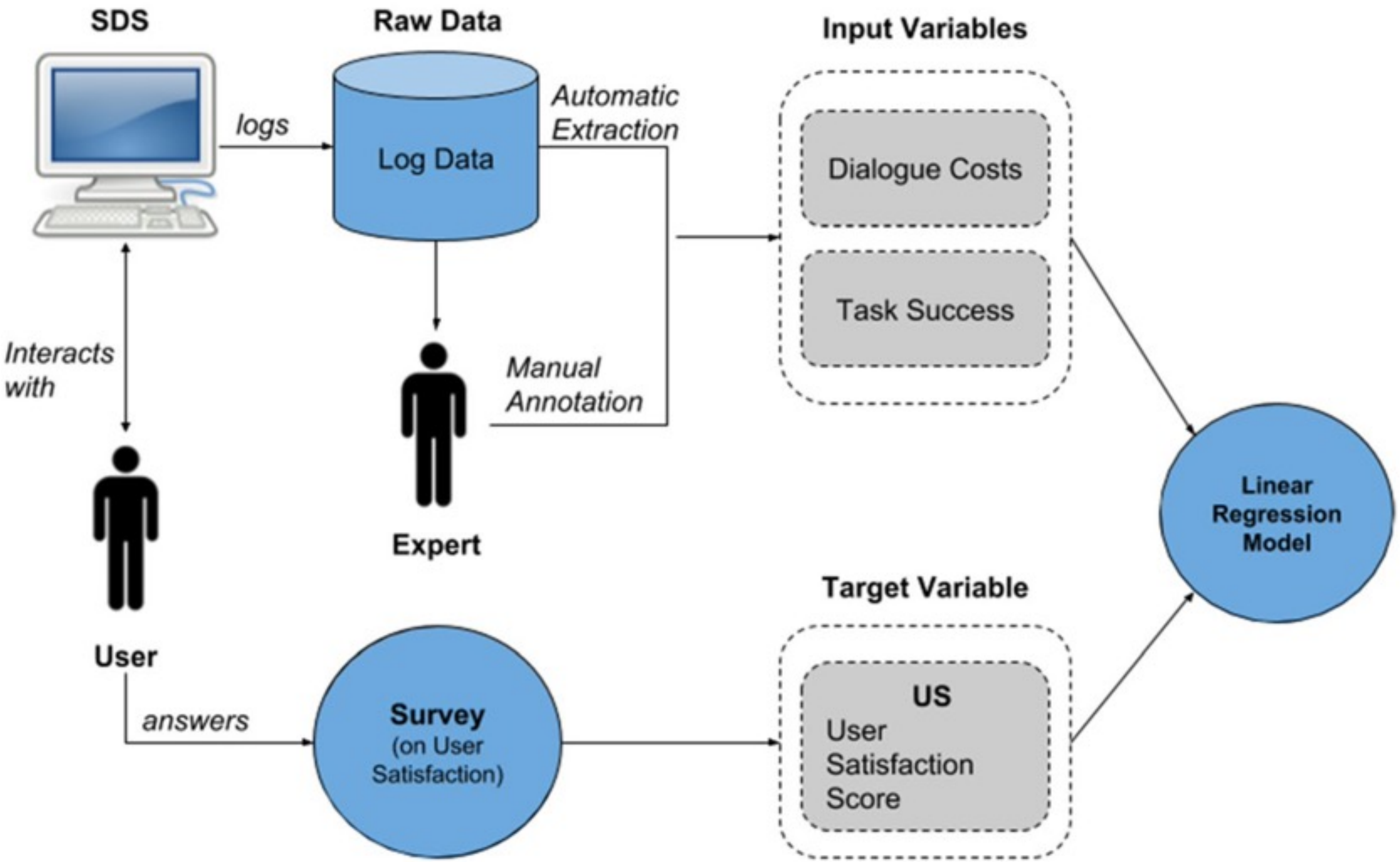
SF-ID Network (E et al., 2019)
Slot Filling (SF)
Intent Detection (ID)

A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling



Source: Haihong, E., Peiqing Niu, Zhongfu Chen, and Meina Song. "A novel bi-directional interrelated model for joint intent detection and slot filling." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5467-5471. 2019.

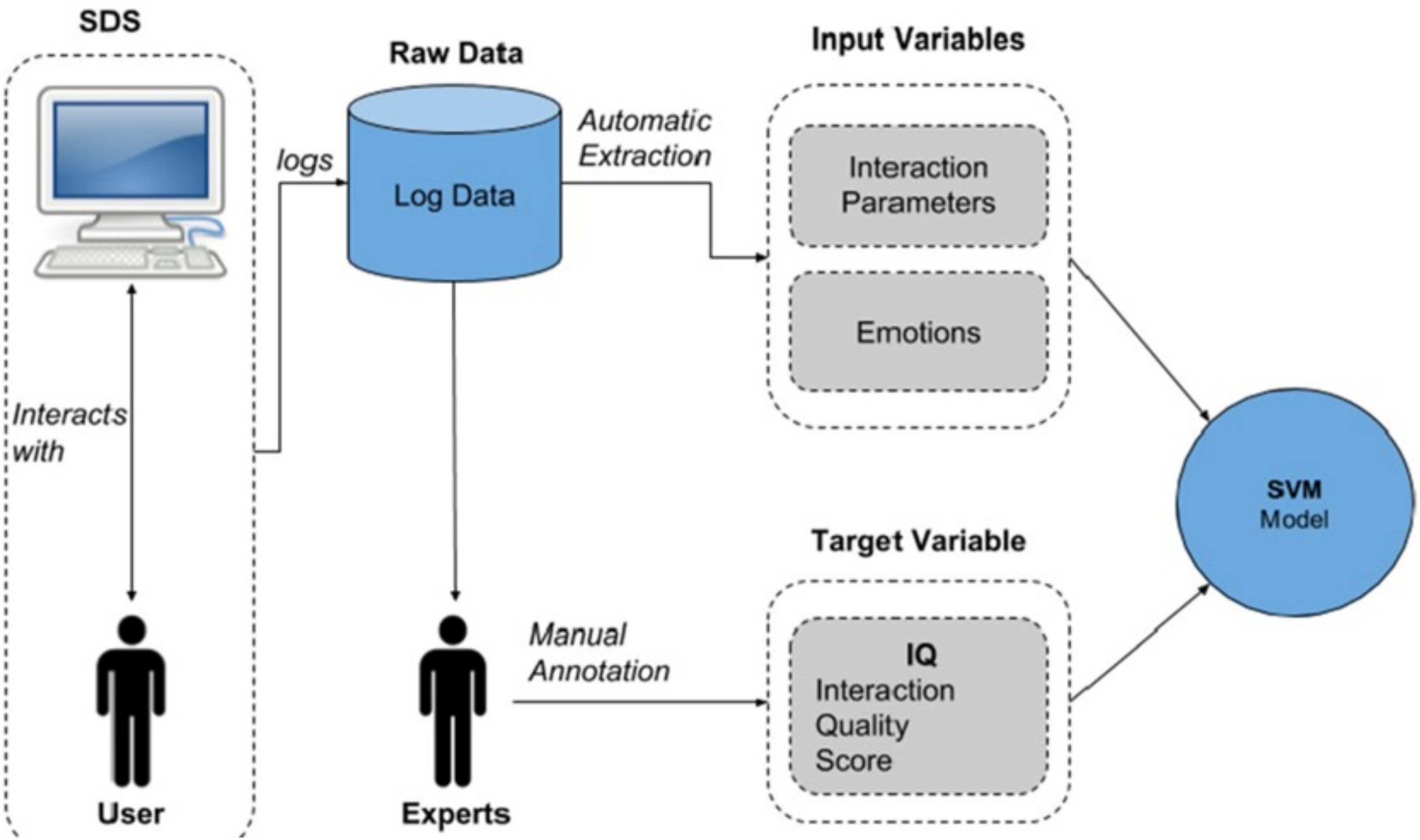
PARAdigm for Dialog System Evaluation
PARADISE Framework (Walker et al. 1997)



Source: Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.

Interaction Quality procedure

(Schmitt and Ultes, 2015)



Source: Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.

Datasets for task-oriented dialogue systems

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

Source: Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.

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>
<tr><th>MultiWOZ</th><th>CrossWOZ</th></tr>
<tr><td>usr: I’m looking for a college type attraction.
...
usr: I would like to visit in town centre please.
...
usr: Can you find an Indian restaurant for me that is also in the town centre?</td><td rowspan="3">usr: 你好，可以帮我推荐一个评分是4.5分以上的景点吗?
Hello, could you recommend an attraction with a rating of 4.5 or higher?
sys: 天安门城楼，簋街小吃和北京欢乐谷都是很不错的地方呢。
Tiananmen, Gui Street, and Beijing Happy Valley are very nice places.
usr: 我喜欢北京欢乐谷，你知道这个景点周边的酒店都是什么吗?
I like Beijing Happy Valley. What hotels are around this attraction?
sys: 那可多了，有A酒店, B酒店, C酒店。
There are many, such as hotel A, hotel B, and hotel C.
usr: 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?
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>
<tr><th>Schema</th></tr>
<tr><td>usr: I want a hotel in San Diego and I want to check out on Thursday next week.
...
usr: I need a one way flight to go there.</td></tr>
</table>

Source: Zhu, Qi, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. "Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset." arXiv preprint arXiv:2002.11893 (2020).

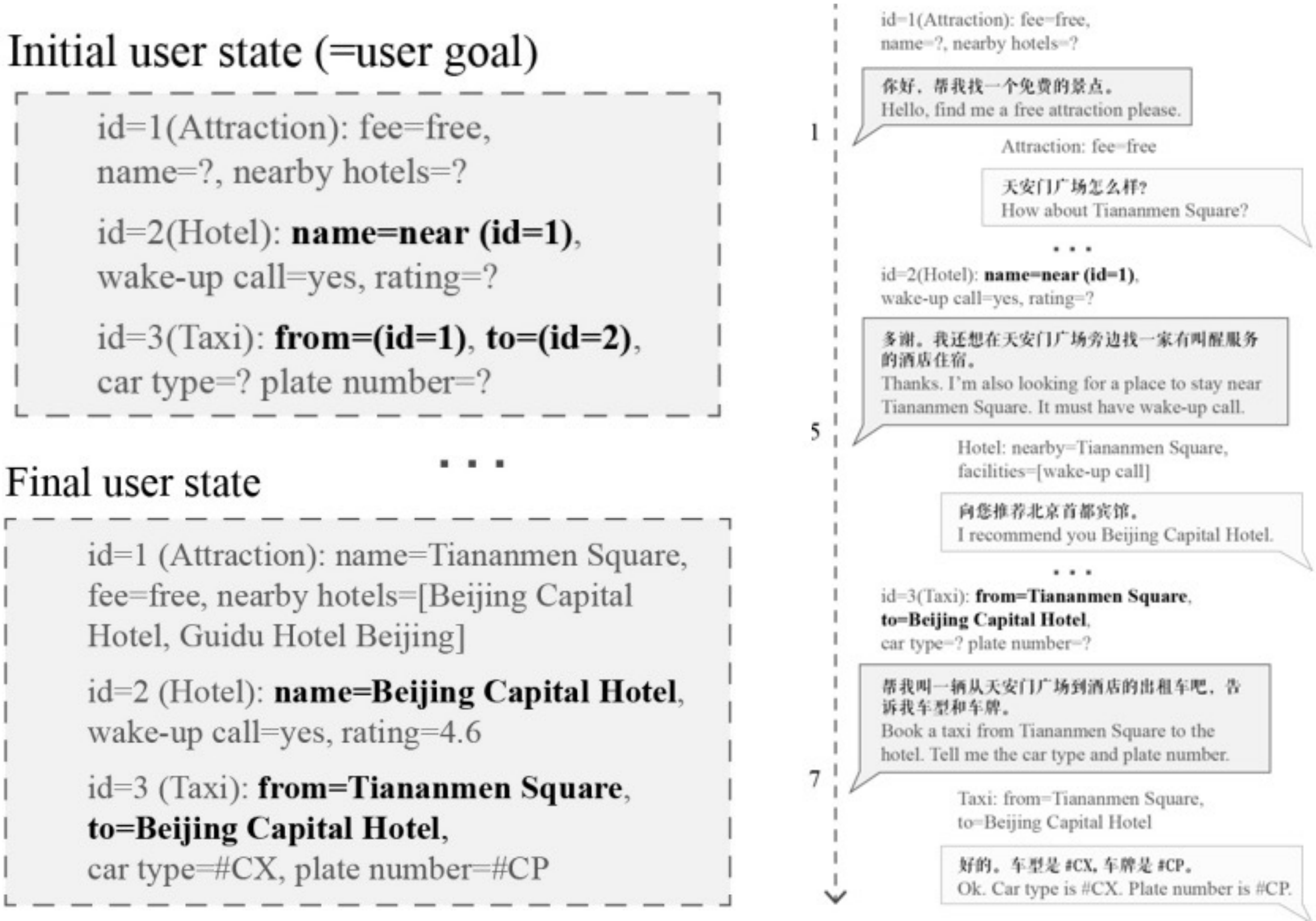
CrossWOZ:

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

Type	Single-domain goal					Multi-domain goal		
Dataset	DSTC2	WOZ 2.0	Frames	KVRET	M2M	MultiWOZ	Schema	**CrossWOZ**
Language	EN	EN	EN	EN	EN	EN	EN	**CN**
Speakers	H2M	H2H	H2H	H2H	M2M	H2H	M2M	**H2H**
# Domains	1	1	1	3	2	7	16	**5**
# Dialogues	1,612	600	1,369	2,425	1,500	8,438	16,142	**5,012**
# Turns	23,354	4,472	19,986	12,732	14,796	115,424	329,964	**84,692**
Avg. domains	1	1	1	1	1	1.80	1.84	**3.24**
Avg. turns	14.5	7.5	14.6	5.3	9.9	13.7	20.4	**16.9**
# Slots	8	4	61	13	14	25	214	**72**
# Values	212	99	3,871	1363	138	4,510	14,139	**7,871**

Source: Zhu, Qi, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. "Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset." arXiv preprint arXiv:2002.11893 (2020).

Task-Oriented Dialogue

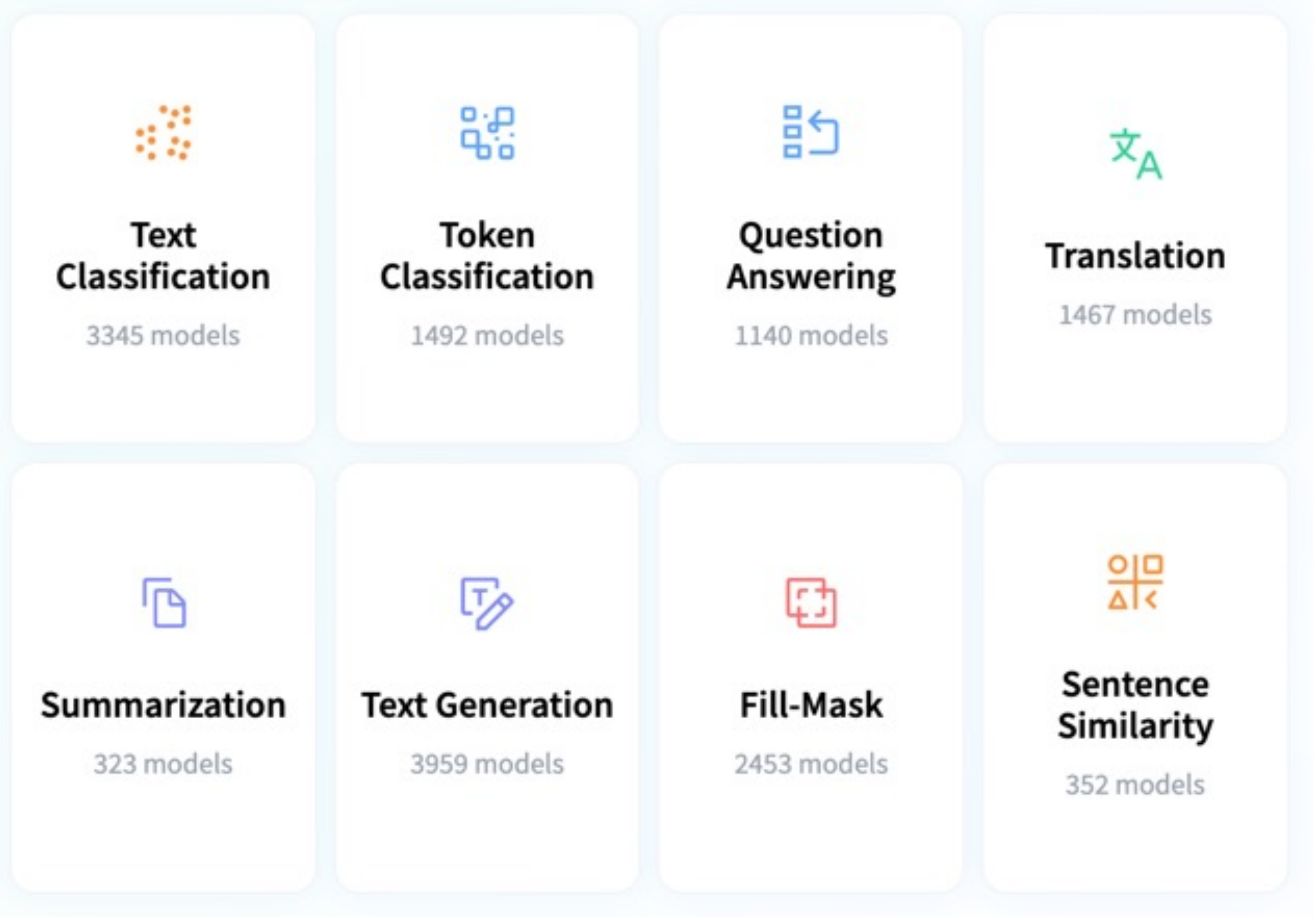


Source: Zhu, Qi, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. "Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset." arXiv preprint arXiv:2002.11893 (2020).

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

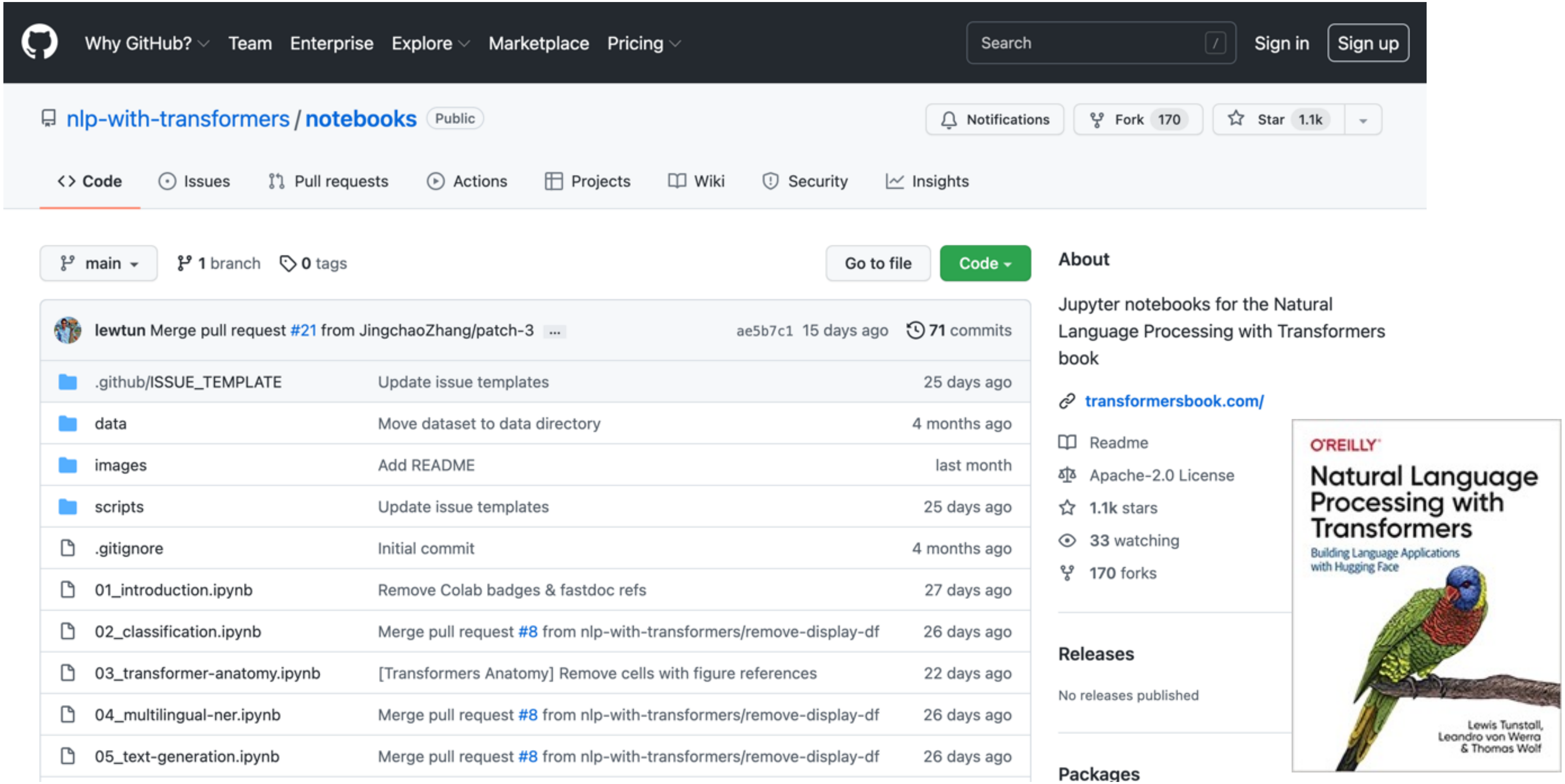
Speaker	Utterance	Dialog State Update
User System	Hello! I am looking for a local guesthouse in the centre. 你好！我在市中心找一家本地宾馆。 OK. I am glad to recommend Alexander Bed and Breakfast to you. 好的。这边很高兴向您推荐亚历山大住宿加早餐旅馆。	hotel: {area: centre, type: guesthouse} 旅馆: {区域: 中心, 类型: 宾馆}
User System	Where is it? 它在哪里? It is at 56 Saint Barnabas Road. 圣巴纳巴斯路56号。	hotel: {name: alexander bed and breakfast} 旅馆: {名称: 亚历山大住宿加早餐旅馆}
User System	I also wish to have a meal in a local European restaurant in the centre. 我还想在市中心的一家本地欧洲餐厅吃饭。 You can choose Eraina. 您可以选择伊莱娜。	restaurant: {food: european, area: centre} 餐厅: {食物: 欧洲的, 区域: 中心}
User System	Please give me its address. 那请给我它的地址。 It is in Free School Lane City Centre. 市中心自由校园巷。	restaurant: {name: eraina} 餐厅: {名称: 伊莱娜}
User System	Ok. I'll go there. I need to book a taxi from Alexander Bed and Breakfast to Eraina after 07:00. 好。那我去那里。我还要预订一辆7:00时从亚历山大住宿加早餐旅馆到伊莱娜的出租车。 Well. I find a yellow Skoda. 好的。是一辆黄色的斯柯达。	taxi: {leaveAt: 07:00, destination: eraina, departure: alexander bed and breakfast} 出租车: {出发时间: 07:00, 目的地: 伊莱娜, 出发地: 亚历山大住宿加早餐旅馆}
User System	How about its phone number? 它的电话号码是多少? It is 78519675253. 78519675253。	No update
User System	Thank you for your help. Bye! 谢谢你帮忙。再见！ A pleasure. Bye bye! 我很乐意。再见！	No update

Hugging Face Tasks
Natural Language Processing



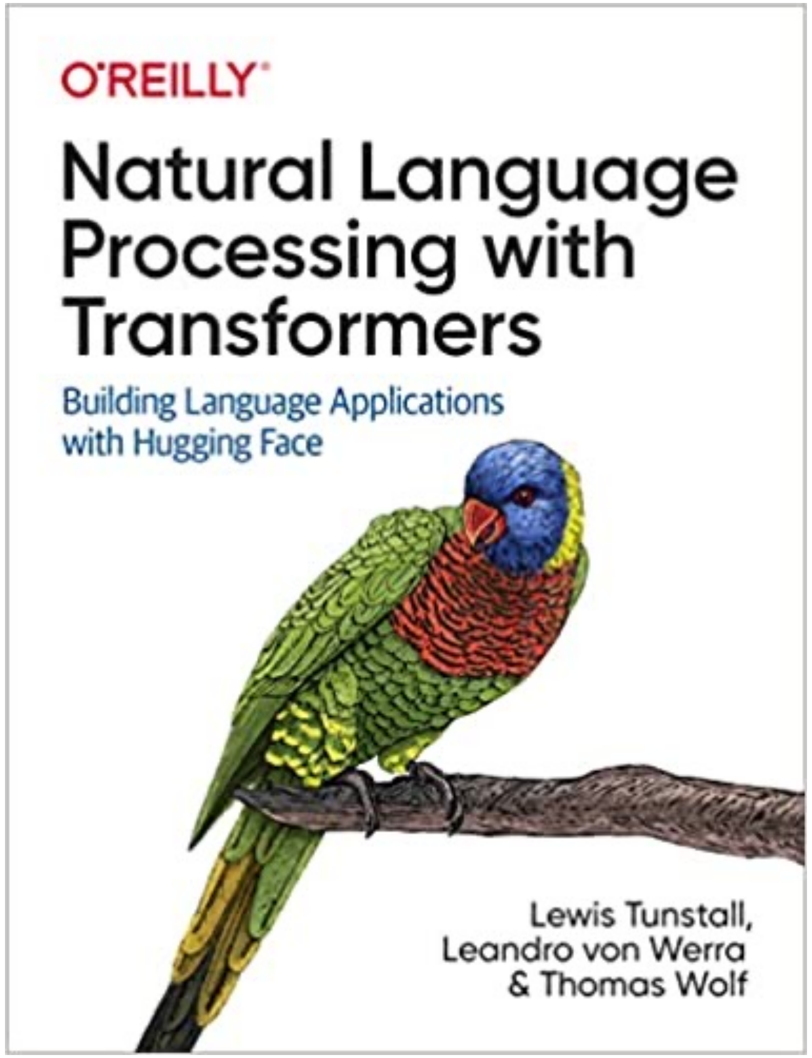
https://huggingface.co/tasks

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:

Chapter	Colab	Kaggle	Gradient	Studio Lab
Introduction	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Classification	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Transformer Anatomy	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Multilingual Named Entity Recognition	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Generation	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Summarization	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Question Answering	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Making Transformers Efficient in Production	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Dealing with Few to No Labels	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Training Transformers from Scratch	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Future Directions	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab

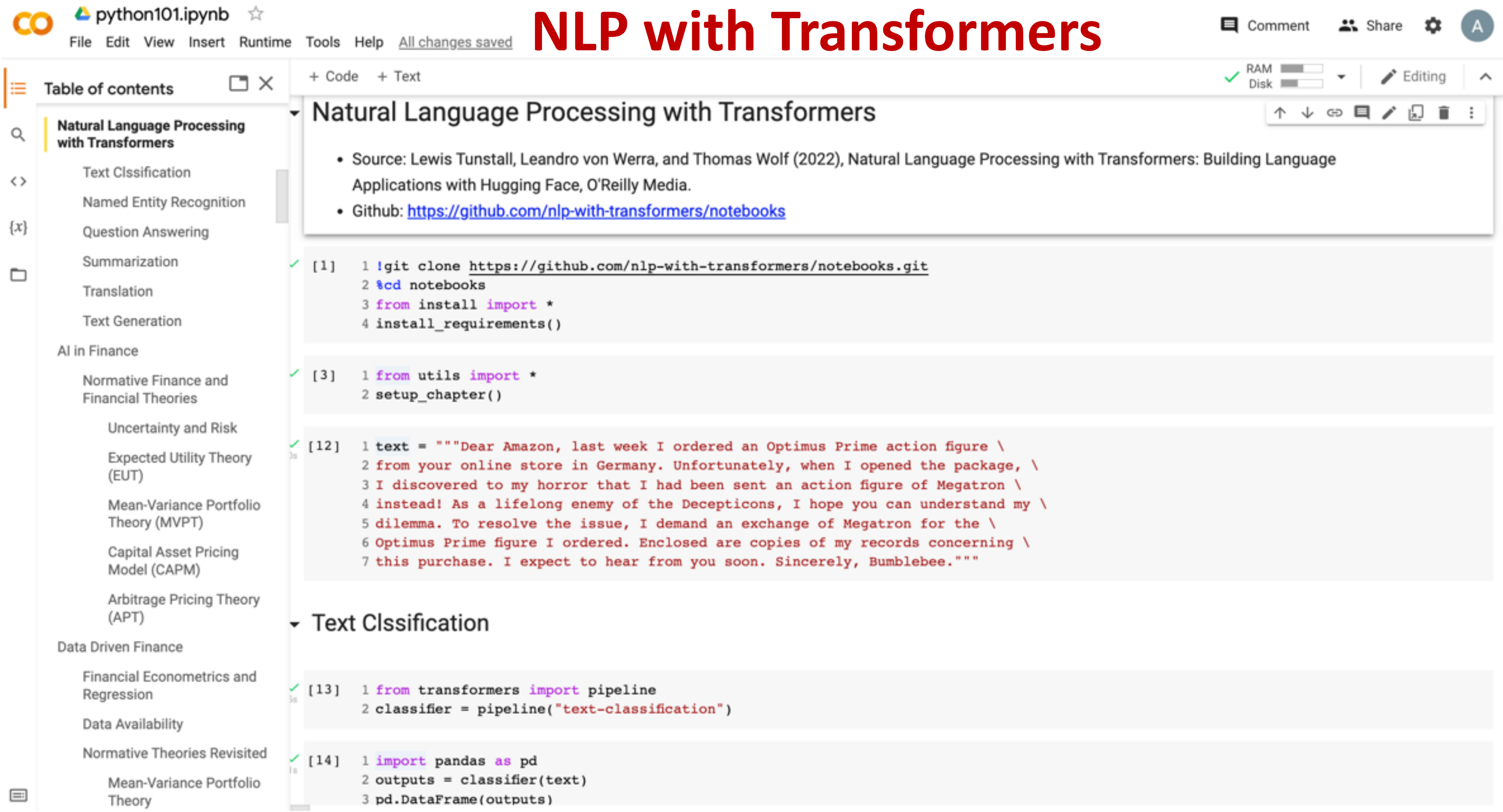
Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using Kaggle, Gradient, or SageMaker Studio Lab. These platforms tend to provide more performant GPUs like P100s, all for free!

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

Python in Google Colab (Python101)

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

NLP with Transformers



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

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

Text Classification

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM
Disk

Editing

Table of contents

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

[10]
```
!nvidia-smi
```

```
# Uncomment and run this cell if you're on Colab or Kaggle
!git clone https://github.com/nlp-with-transformers/notebooks.git
%cd notebooks
from install import *
install_requirements()
```

[12]
```
# hide
from utils import *
setup_chapter()
```

The Dataset

[13]
```
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)

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Multilingual 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

```
[ ]
#NER: https://huggingface.co/tasks/token-classification
!pip install transformers
from transformers import pipeline
classifier = pipeline("ner")
classifier("Hello I'm Omar and I live in Zürich.")
```

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

```
No model was supplied, defaulted to dbmdz/bert-large-cased-finetuned-conll03-english (https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-eng
[{'end': 14,
  'entity': 'I-PER',
  'index': 5,
  'score': 0.99770516,
  'start': 10,
  'word': 'Omar'},
 {'end': 35,
  'entity': 'I-LOC',
  'index': 10,
  'score': 0.9968976,
  'start': 29,
  'word': 'Zürich'}]
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk Editing

Text Summarization

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/nlp-with-transformers/notebooks

```python
#Source: https://huggingface.co/tasks/summarization
!pip install transformers
from transformers import pipeline
classifier = pipeline("summarization")
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 thar
classifier(text, max_length=30)
```

```
No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 (https://huggingface.co/sshleifer/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'}]
```

```python
#!pip install transformers
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
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 \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
from transformers import pipeline
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Text Generation

Comment Share

+ Code + Text

RAM Disk Editing

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

[9]

```
#Source: https://huggingface.co/tasks/text-generation
#!pip install transformers
from transformers import pipeline
generator = pipeline('text-generation', model = 'gpt2')
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.\n\nBut then, one day, I'm not trying to teach the language in my head.\n\n"},
 {'generated_text': "Hello, I'm a language model. I'm an implementation for the type system. I'm working with types and programming language constructs. I a
 {'generated_text': "Hello, I'm a language modeler, not a programmer. As you know, languages are not a linear model. The thing that jumps out at"}]
```

```
from transformers import pipeline
generator = pipeline('text-generation', model = 'gpt2')
outputs = generator("Once upon a time", max_length = 30, num_return_sequences=3)
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
```

[1]

```
from transformers import pipeline
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

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

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Question Answering and Dialogue Systems

Editing

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

Question Answering

```
[3]
!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)
```

```
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}
```

```
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

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Question Answering

RAM Disk Editing

[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 gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
gravity
```

[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 gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
graupel
```

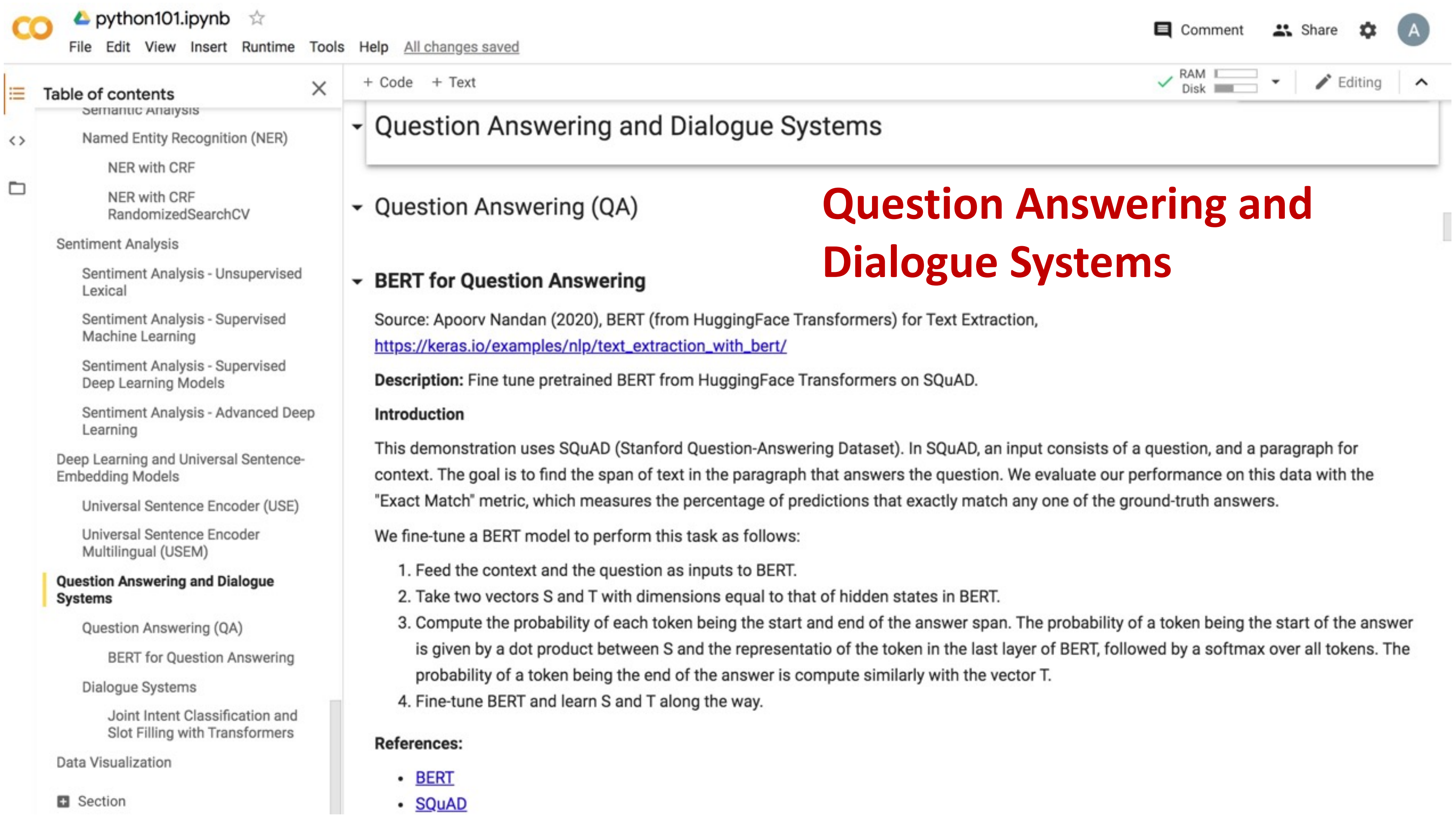
```
#from transformers import pipeline
#qamodel = pipeline("question-answering", model ='deepset/roberta-base-squad2')
question = "Where do water droplets collide with ice crystals to form precipitation?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water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

```
within a cloud
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

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

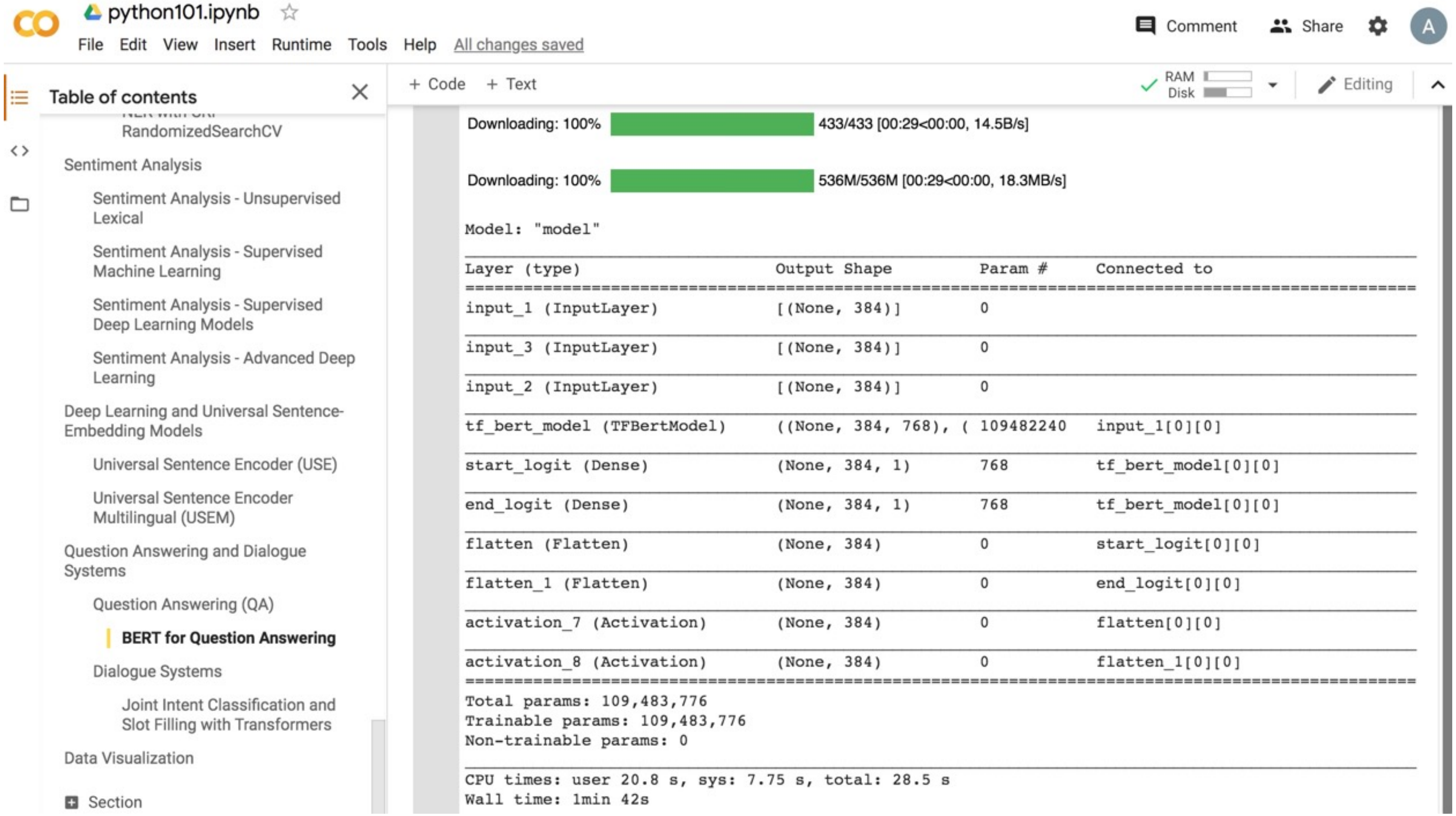


Question Answering and Dialogue Systems

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

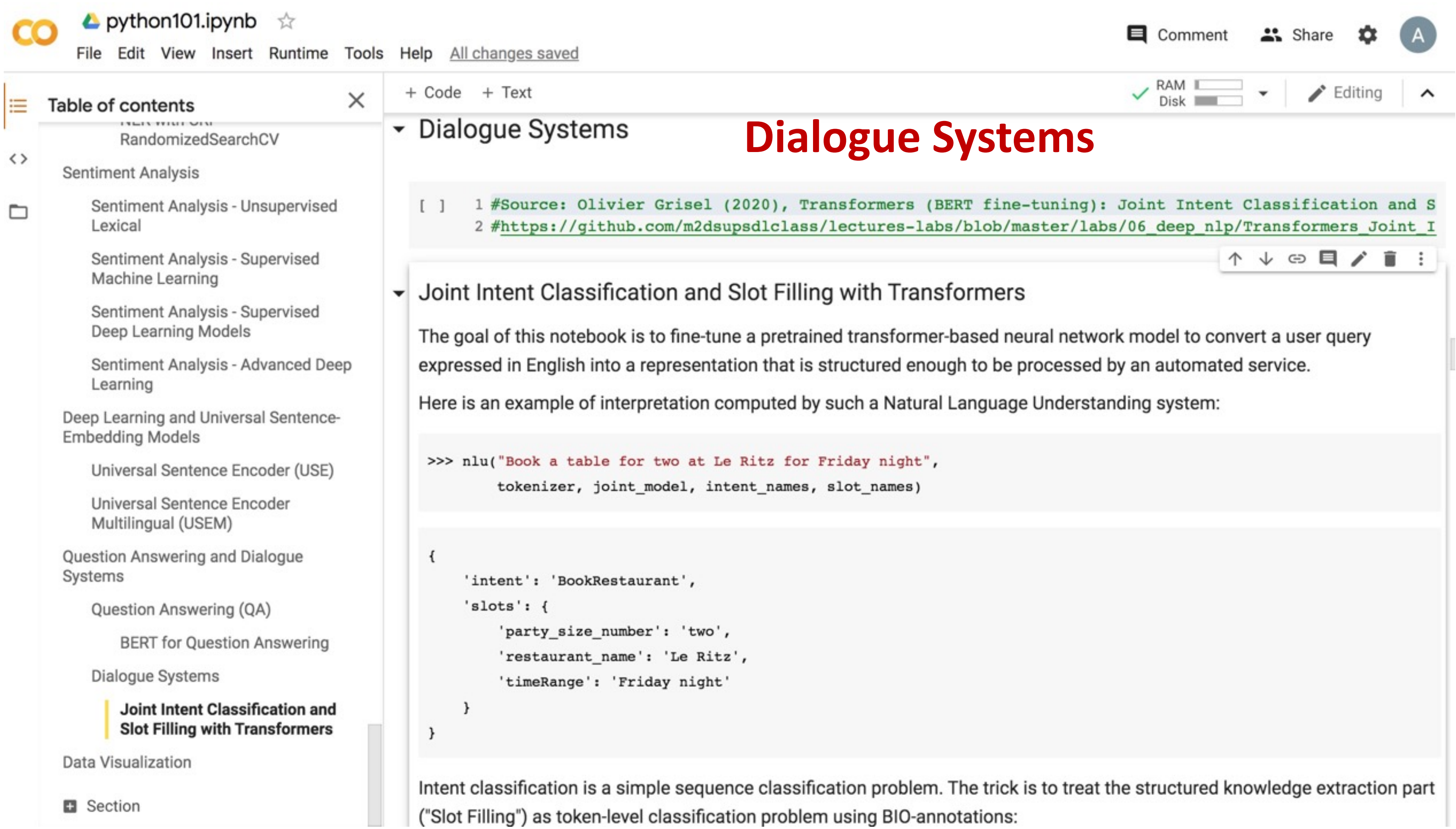
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

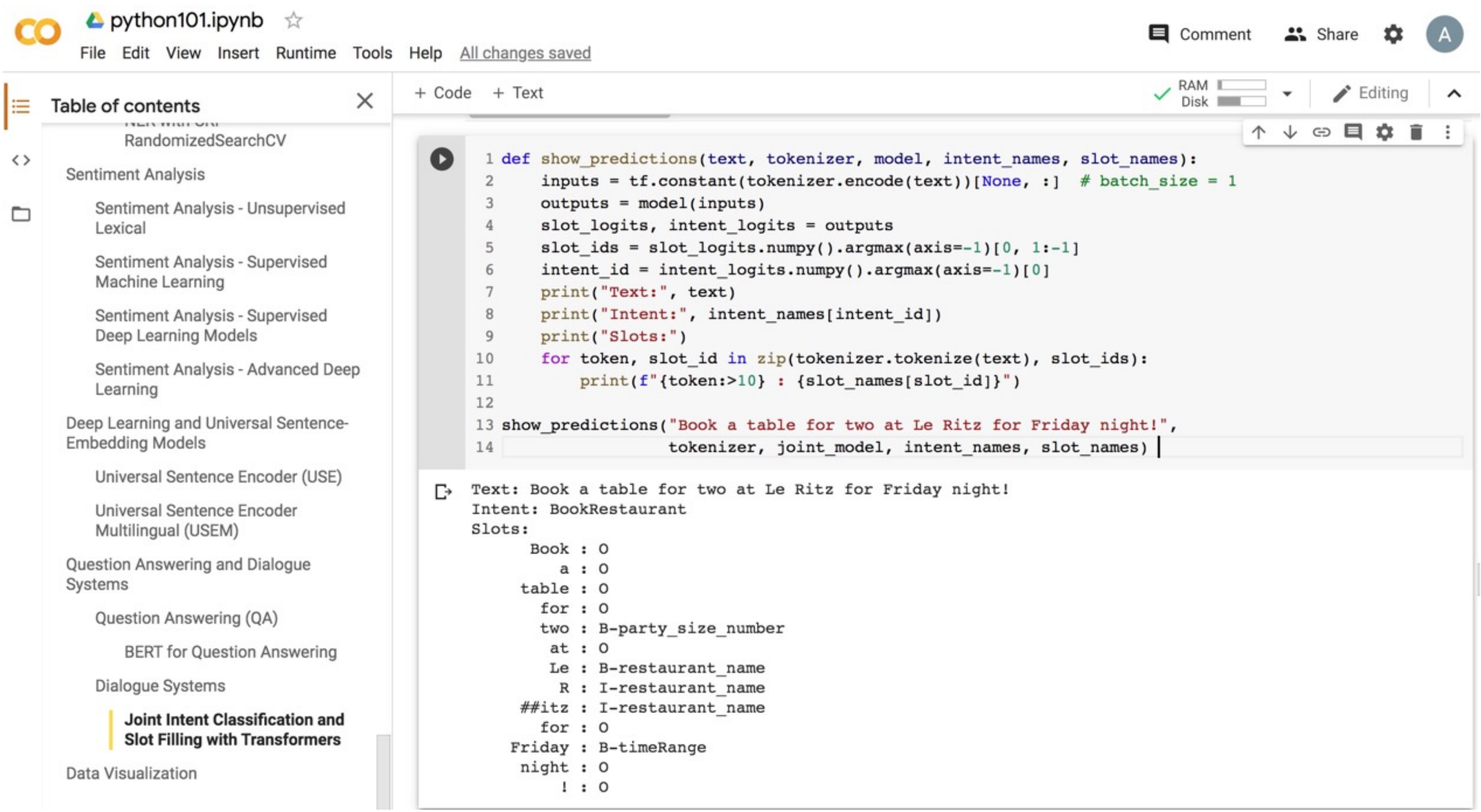
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

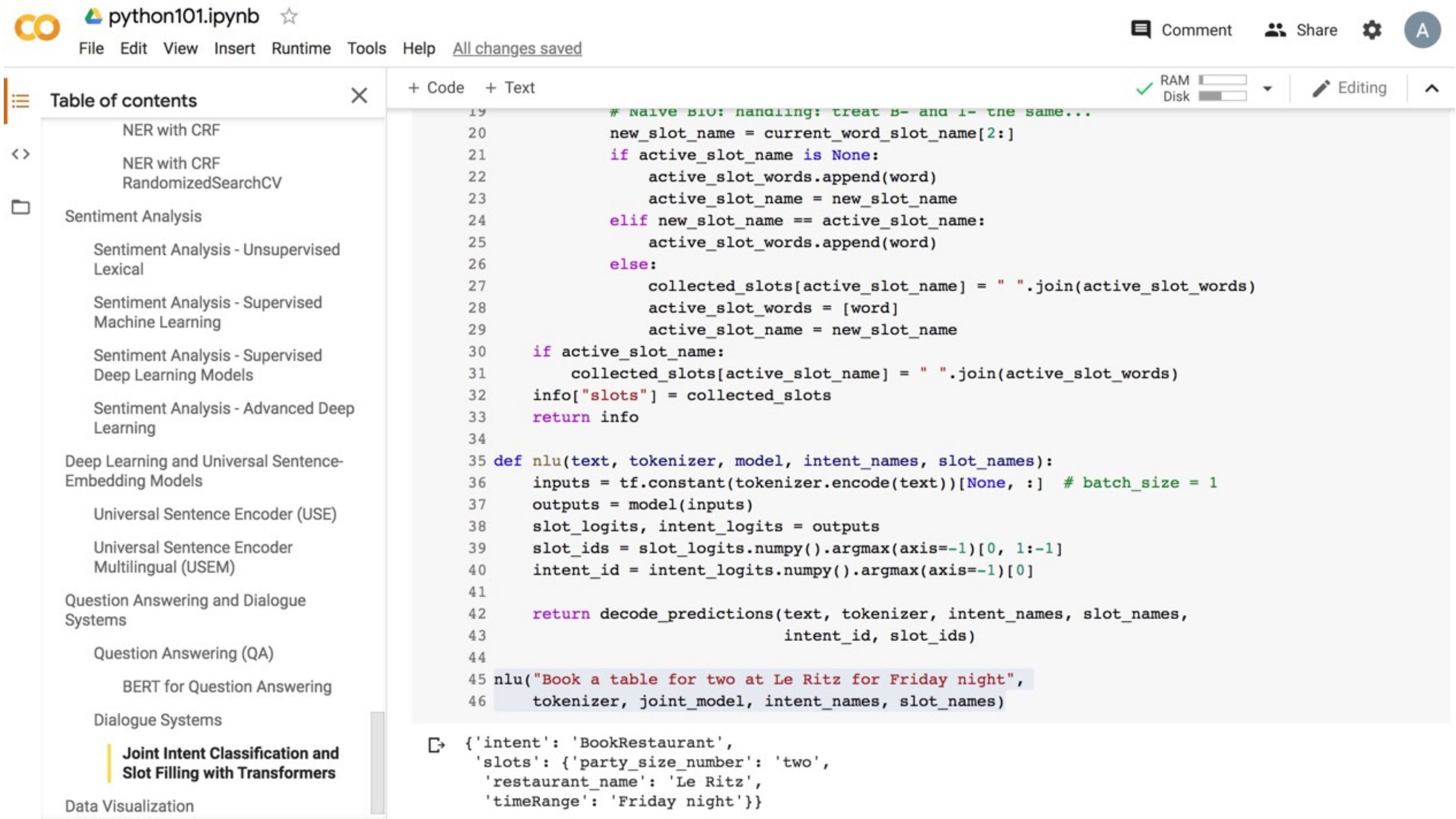
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

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



https://tinyurl.com/aintpupython101

Question Answering

```
!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}
```

https://tinyurl.com/aintpupython101

Question Answering

```
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
```

https://tinyurl.com/aintpupython101

Question Answering

```
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'])
```

```
gravity
```

https://tinyurl.com/aintpupython101

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.
- Savaş Yıldırım and Meysam Asgari-Chenaghlu (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.
- Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.
- Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress.
- Mutabazi, Emmanuel, Jianjun Ni, Guangyi Tang, and Weidong Cao (2021) "A review on medical textual question answering systems based on deep learning approaches." Applied Sciences 11, no. 12 (2021): 5456.
- Deriu, Jan, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak (2021). "Survey on evaluation methods for dialogue systems." Artificial Intelligence Review 54, no. 1 (2021): 755-810.
- Day, Min-Yuh and Chi-Sheng Hung, "AI Affective Conversational Robot with Hybrid Generative-based and Retrieval-based Dialogue Models", in Proceedings of The 20th IEEE International Conference on Information Reuse and Integration for Data Science (IEEE IRI 2019), Los Angeles, CA, USA, July 30 - August 1, 2019.
- Day, Min-Yuh, Chi-Sheng Hung, Yi-Jun Xie, Jhih-Yi Chen, Yu-Ling Kuo and Jian-Ting Lin (2019), "IMTKU Emotional Dialogue System for Short Text Conversation at NTCIR-14 STC-3 (CECG) Task", The 14th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-14), Tokyo, Japan, June 10-13, 2019.
- Zhou, Hao, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. "Emotional chatting machine: emotional conversation generation with internal and external memory." arXiv preprint arXiv:1704.01074 (2017).
- Yu, Kai, Zijian Zhao, Xueyang Wu, Hongtao Lin, and Xuan Liu. "Rich Short Text Conversation Using Semantic Key Controlled Sequence Generation." IEEE/ACM Transactions on Audio, Speech, and Language Processing (2018)
- Borah, Bhriguraj, Dhrubajyoti Pathak, Priyankoo Sarmah, Bidisha Som, and Sukumar Nandi. "Survey of Textbased Chatbot in Perspective of Recent Technologies." In International Conference on Computational Intelligence, Communications, and Business Analytics, pp. 84-96. Springer, Singapore, 2018.
- Haihong, E., Peiqing Niu, Zhongfu Chen, and Meina Song. "A novel bi-directional interrelated model for joint intent detection and slot filling." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5467-5471. 2019.
- Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "Squad: 100,000+ questions for machine comprehension of text." arXiv preprint arXiv:1606.05250 (2016).
- Zhu, Qi, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. "Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset." arXiv preprint arXiv:2002.11893 (2020).
- Zeng, Zhaohao, Sosuke Kato, Tetsuya Sakai, and Inho Kang (2020), "Overview of the NTCIR-15 Dialogue Evaluation (DialEval-1) Task", Proceedings of NTCIR-15, 2020.
- Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018), Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning, O'Reilly.
- 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.
- Jake VanderPlas (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
- The Super Duper NLP Repo, https://notebooks.quantumstat.com/
- Jay Alammar (2018), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/
- Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/
- NLP with Transformer, https://github.com/nlp-with-transformers/notebooks
- Min-Yuh Day (2022), Python 101, https://tinyurl.com/aintpupython101