Open-domain Question Answering

Danqi Chen

Princeton University

@danqi_chen

Scott Wen-tau Yih

Facebook AI Research

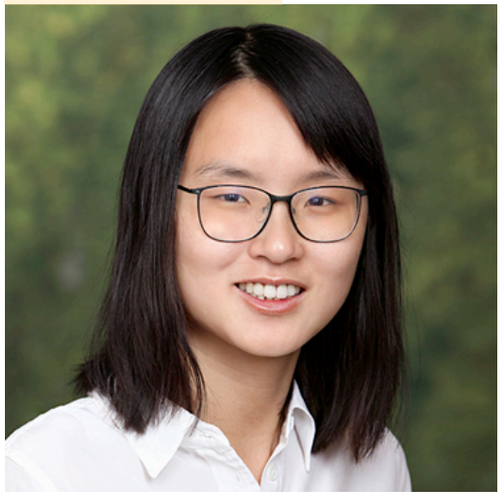
@scottyih

https://github.com/danqi/acl2020-openqa-tutorial

July 5, 2020

Who we are

Danqi



- Assistant prof at Princeton University since 2019 fall
- Working in QA since 2016
- PhD thesis on neural reading comprehension and question answering
- Author of DrQA, CoQA, SpanBERT/ RoBERTa, dense passage retriever (DPR), Stanford Attentive Reader

Scott

- Research scientist at Facebook AI Research
- Working in QA since 2013 (or 2001☺)
- Taught “Question Answering with Knowledge Base, Web and Beyond” in NAACL & SIGIR 2016
- Author of WikiQA, WebQSP, DPR, retrieval-augmented generation (RAG)

Tutorial slides

Check out our latest slides at:

https://github.com/danqi/acl2020-openqa-tutorial

ACL2020 Tutorial: Open-Domain Question Answering

This ACL2020 tutorial will be held on July 5th, 2020, by Danqi Chen `<danqic@cs.princeton.edu>` and Scott Yih `<scottyih@fb.com>`. The video is *NOT* pre-recorded and we will have a 3.5-hour live session at **3-6:30pm PST**. You can find more information below and hope to see you there!

Tutorial Slides

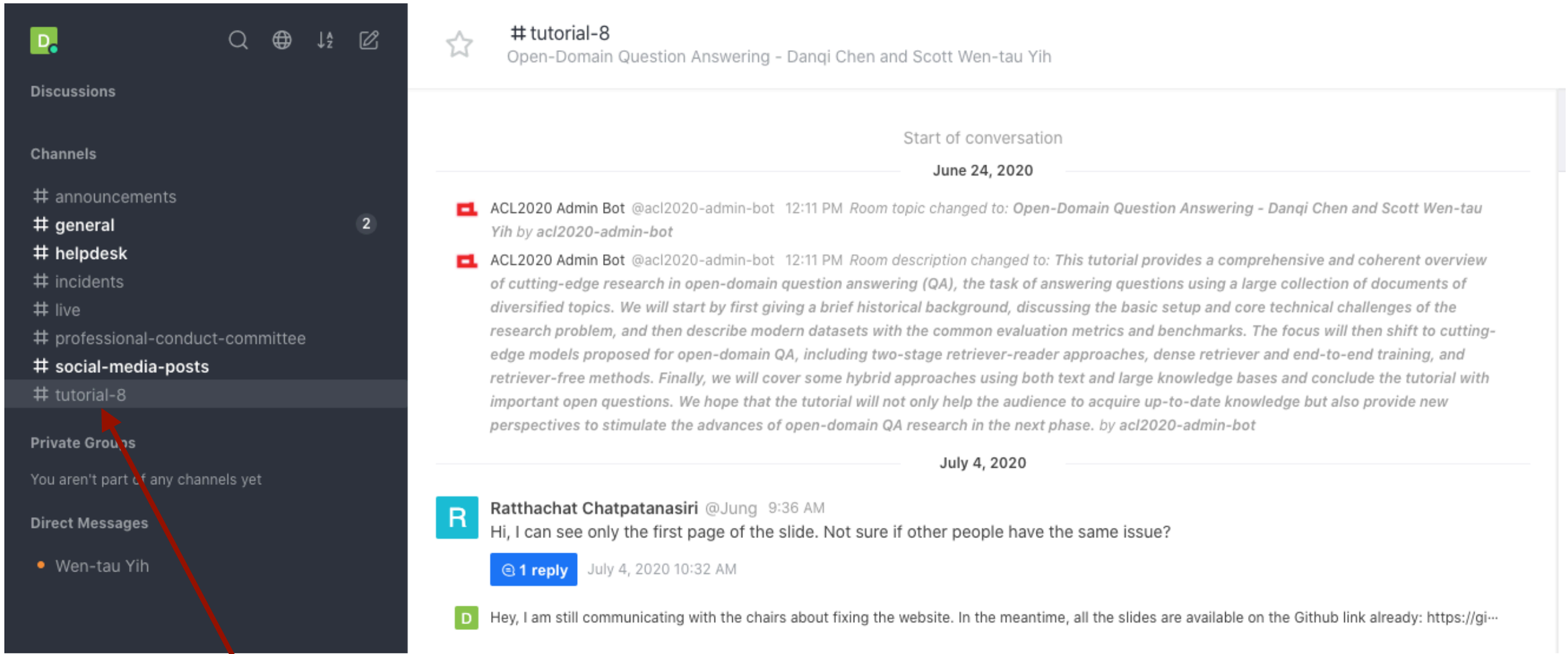
NEW: All the slides are available now!

We recommend reading our draft slides before the tutorial. We may have some minor last-minute changes, so please check out the latest version before the live session.

1. Introduction & problem definition
2. A history of open-domain (textual) QA
3. Datasets & evaluation
4. Two-stage retriever-reader approaches
5. Dense retriever and end-to-end training
6. Retrieval-free approaches
7. Open-domain QA using KBs and text
8. Open problems and future directions

Participation + Q & A

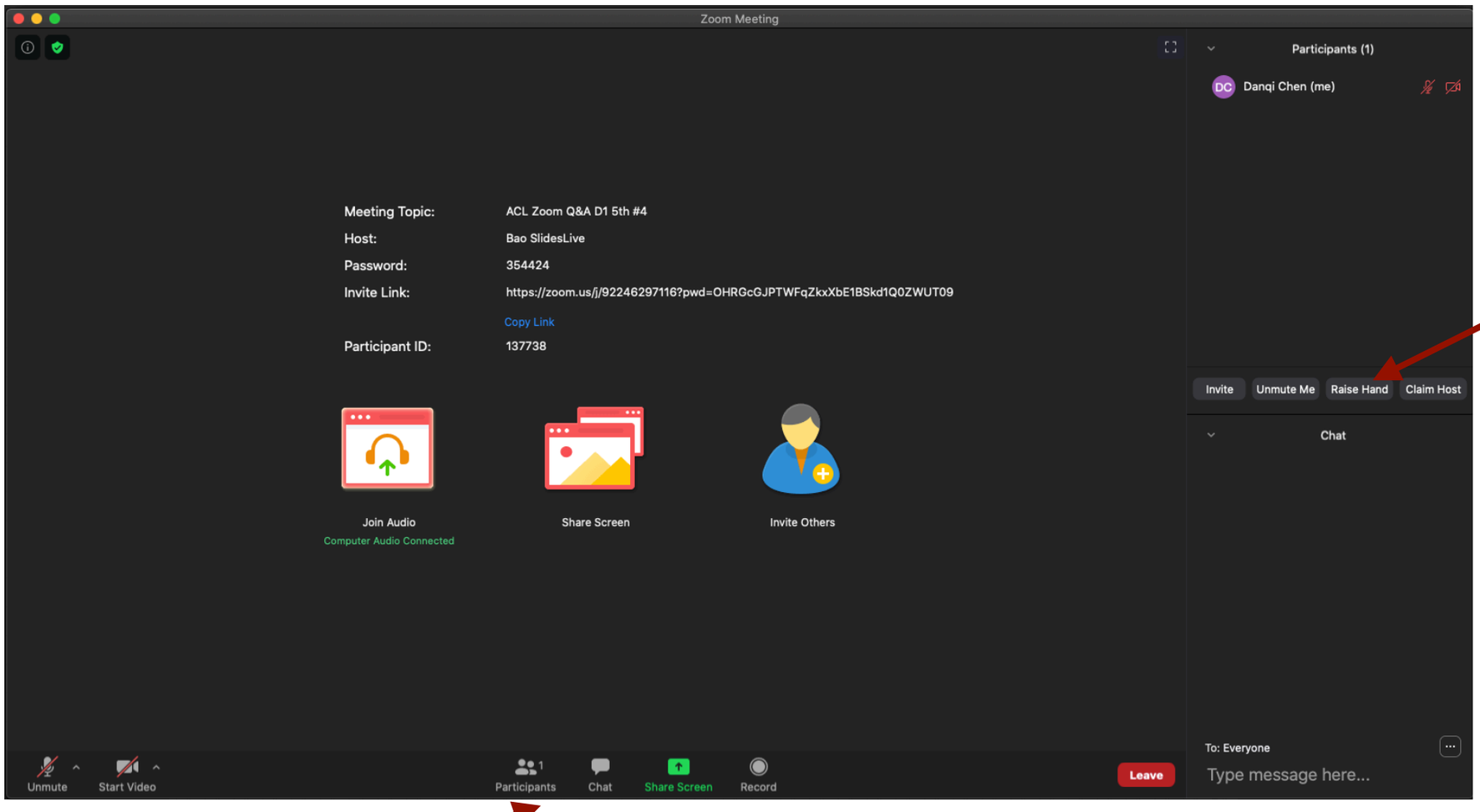
RocketChat: https://acl2020.rocket.chat/channel/tutorial-8



Channel: tutorial-8

Participation + Q & A

Please join us on Zoom if you can!



Use "Raise Hand"

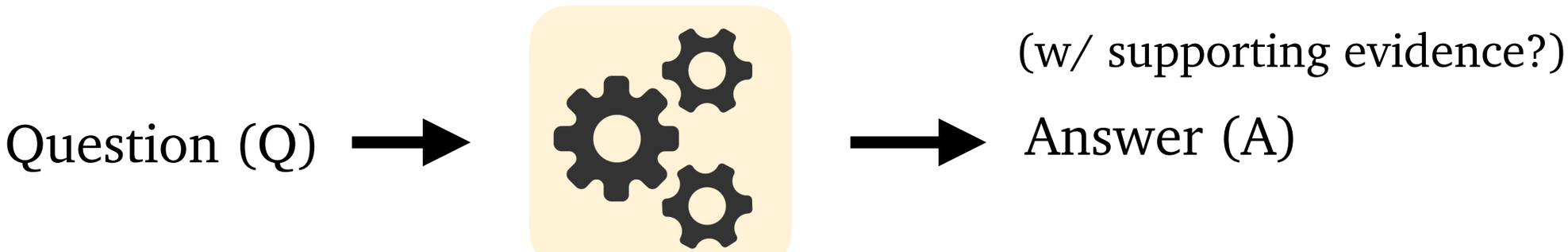
Click "Participants"

We encourage you to turn on your video (you can use virtual background!)

Use RocketChat instead of Zoom Chat!

Open-domain QA

- **Question answering** = build computer systems that **automatically** answer questions posed by humans in a **natural language**



- **Open-domain** = deal with questions about nearly anything, usually rely on *general ontologies* and *world knowledge*

Q: Where does the energy in a nuclear explosion come from?

A: high-speed nuclear reaction

Q: Where is Einstein's house?

A: 112 Mercer St, Princeton, NJ

Q: How many papers were accepted by ACL 2020?

A: 779 papers

Open-domain QA

Knowledge Bases

Tables

Category	Structure	Country	City	Height (metres)	Height (feet)
Mixed use	Burj Khalifa	United Arab Emirates	Dubai	829.8	2,722
Self-supporting tower	Tokyo Skytree	Japan	Tokyo	634	2,080
Mixed use	Shanghai Tower	China	Shanghai	632	2,073
Clock building	Abraj Al Bait Towers	Saudi Arabia	Mecca	601	1,972
Military structure	Large masts of INS Kattabomman	India	Tirunelveli	471	1,545
Mast radiator	Lualualei VLF transmitter	United States	Lualualei, Hawaii	458	1,503
Twin towers	Petronas Twin Towers	Malaysia	Kuala Lumpur	452	1,482
Residential	432 Park Avenue	United States	New York	425.5	1,396
Chimney	Ekibastuz GRES-2 Power Station	Kazakhstan	Ekibastuz	419.7	1,377
Radar	Dimona Radar Facility	Israel	Dimona	400	1,312
Lattice tower	Kiev TV Tower	Ukraine	Kiev	385	1,263
Electricity pylon	Zhoushan Island Overhead Powerline Tie	China	Zhoushan	370	1,214

Web Documents & Wikipedia

Structured

Semi-structured

Unstructured

- This tutorial mostly focuses on open domain **textual** QA
- In Part 6, we will discuss hybrid approaches using both KBs and text

Open-domain QA

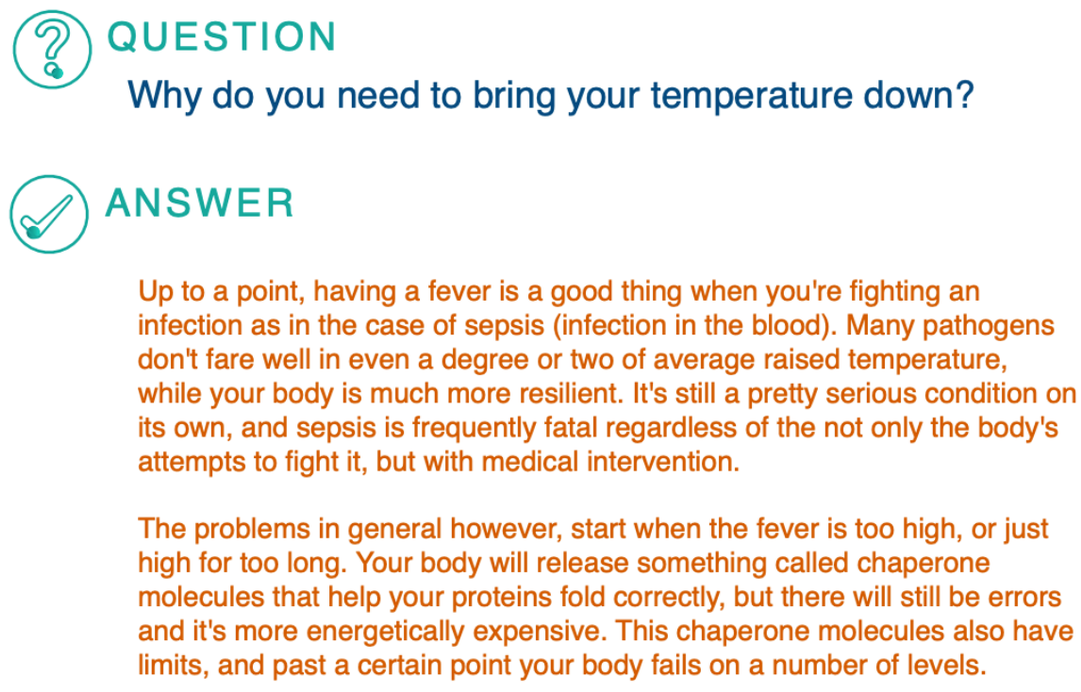
We mostly focus on **factoid question answering**:

- Require systems to return a *short* and *concise* answer to these questions
- In contrast to other QA problems: community question answering, non-factoid or long-form question answering
- Focus more on *retrieval* and *NLU instead of generation*

How do Jellyfish function without brains or nervous systems?

Why can't humans see in the dark?

How to protect yourself from COVID-19?



Prager, 2007. Open-Domain Question Answering

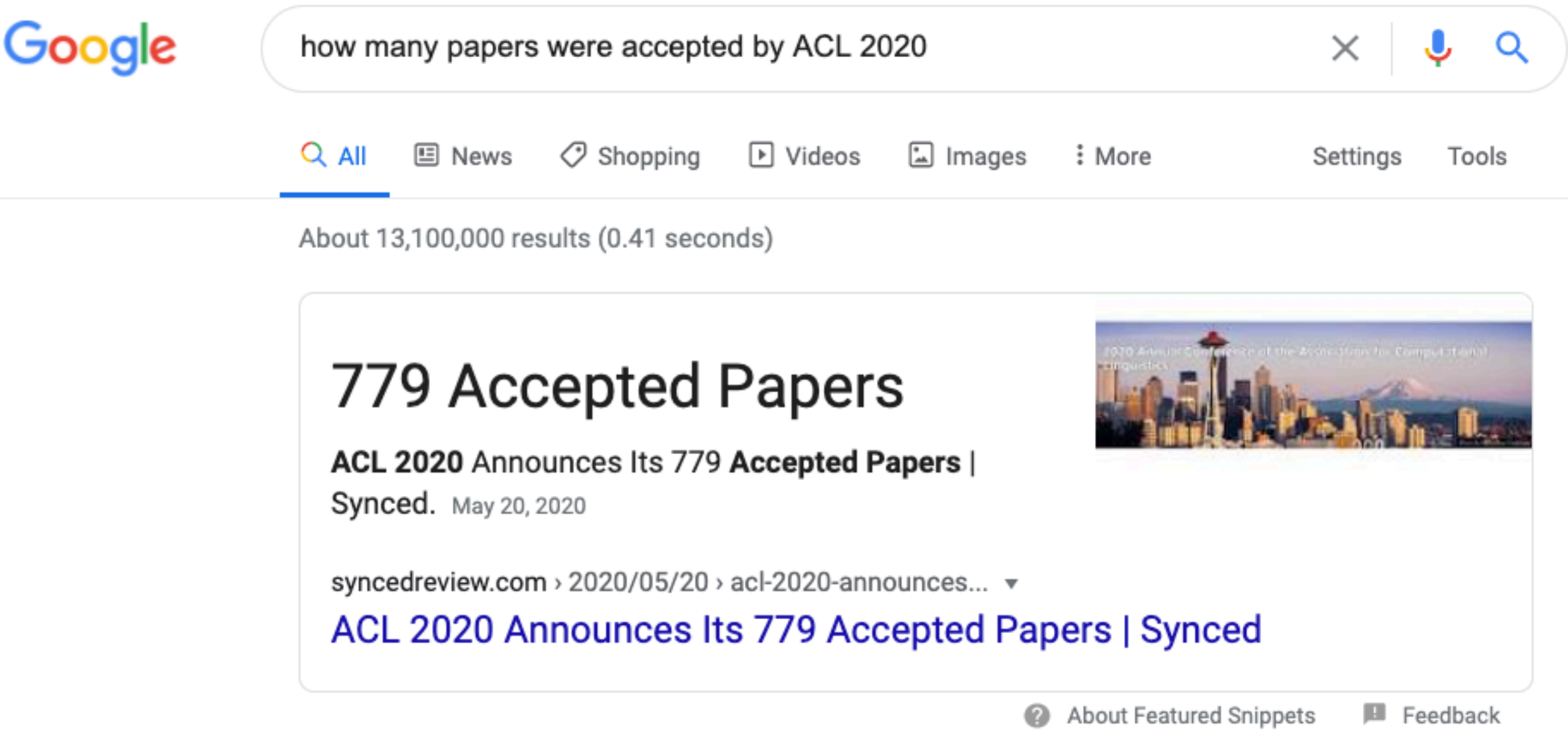
Fan et al., 2019. ELI5: Long Form Question Answering

Search engines: from keyword matching to question answering

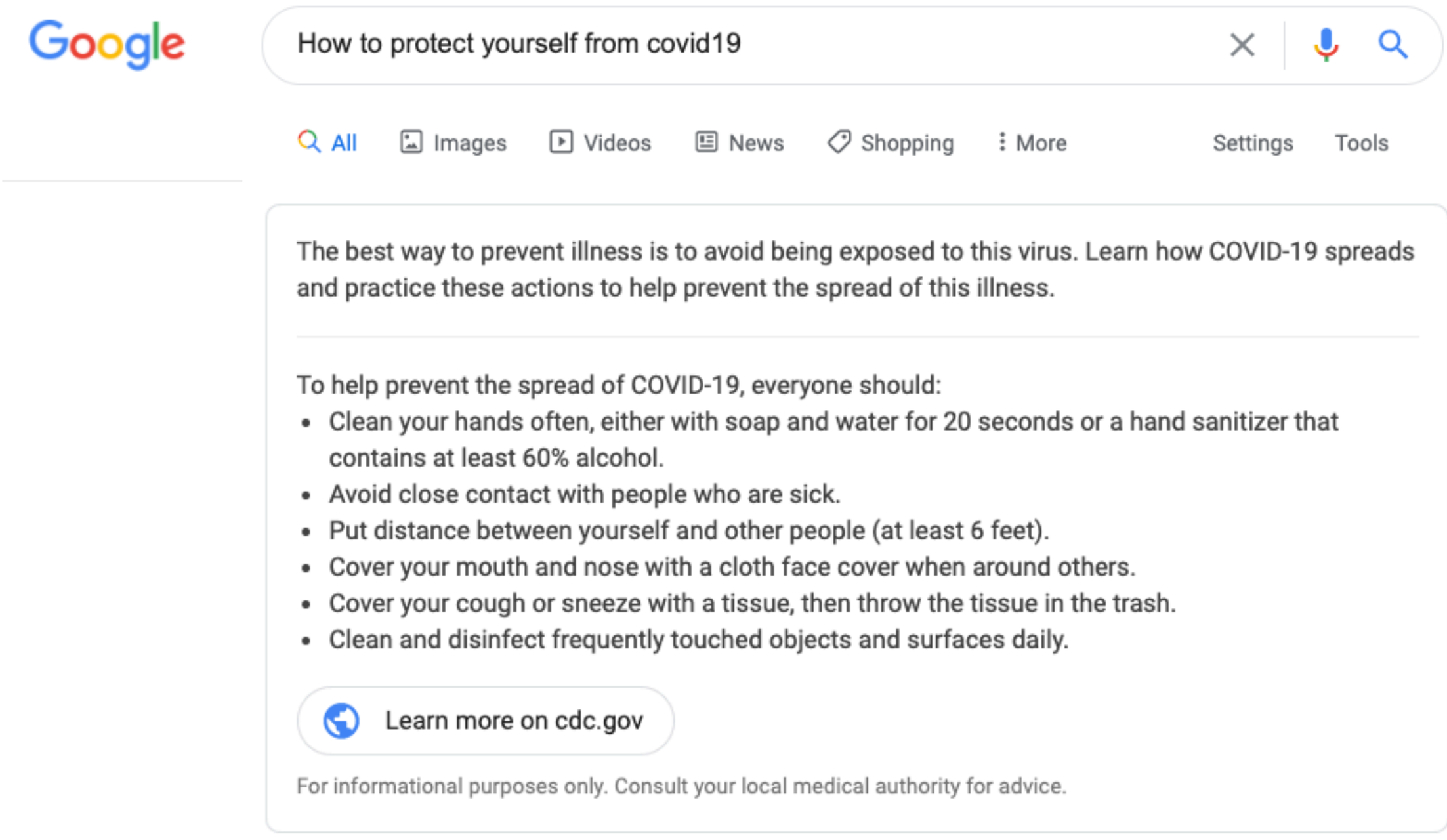
Search needs a shake-up

Etzioni, 2011. Search needs a shake-up

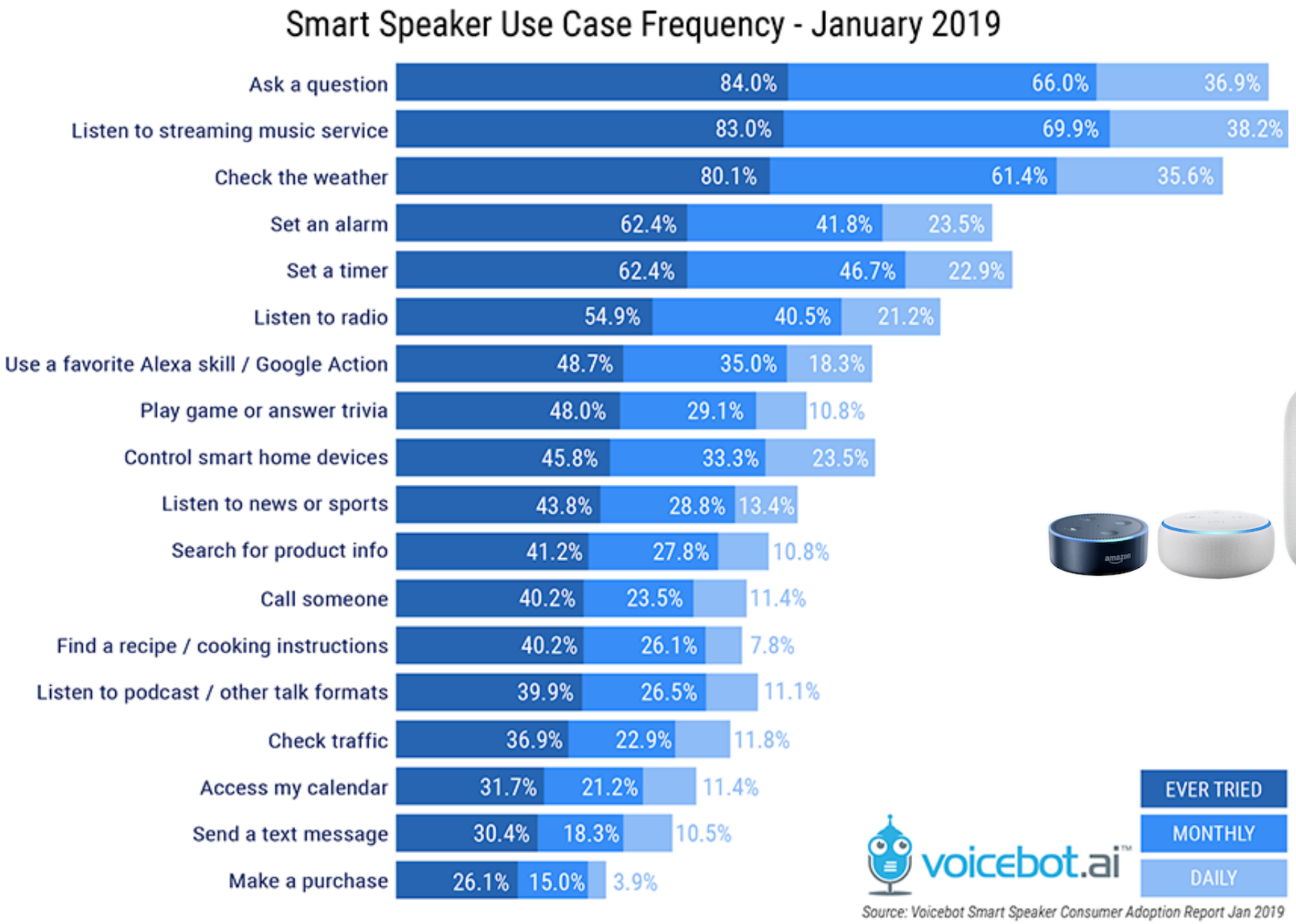
Search engines: from keyword matching to question answering



Search engines: from keyword matching to question answering



People ask lots of questions on digital personal assistants



https://voicebot.ai/2019/03/12/smart-speaker-owners-agree-that-questions-music-and-weather-are-killer-apps-what-comes-next/

Why give this tutorial today?

NAACL 2001

Open-Domain Textual Question Answering

Sanda Harabagiu and Dan Moldovan
Department of Computer Science and Engineering, Southern Methodist University

TREC QA 1999–2001 competitions and participant systems

EACL 2003

Question Answering Techniques for the World Wide Web

Jimmy Lin and Boris Katz
MIT Artificial Intelligence Laboratory

WWW-based QA

NAACL 2012

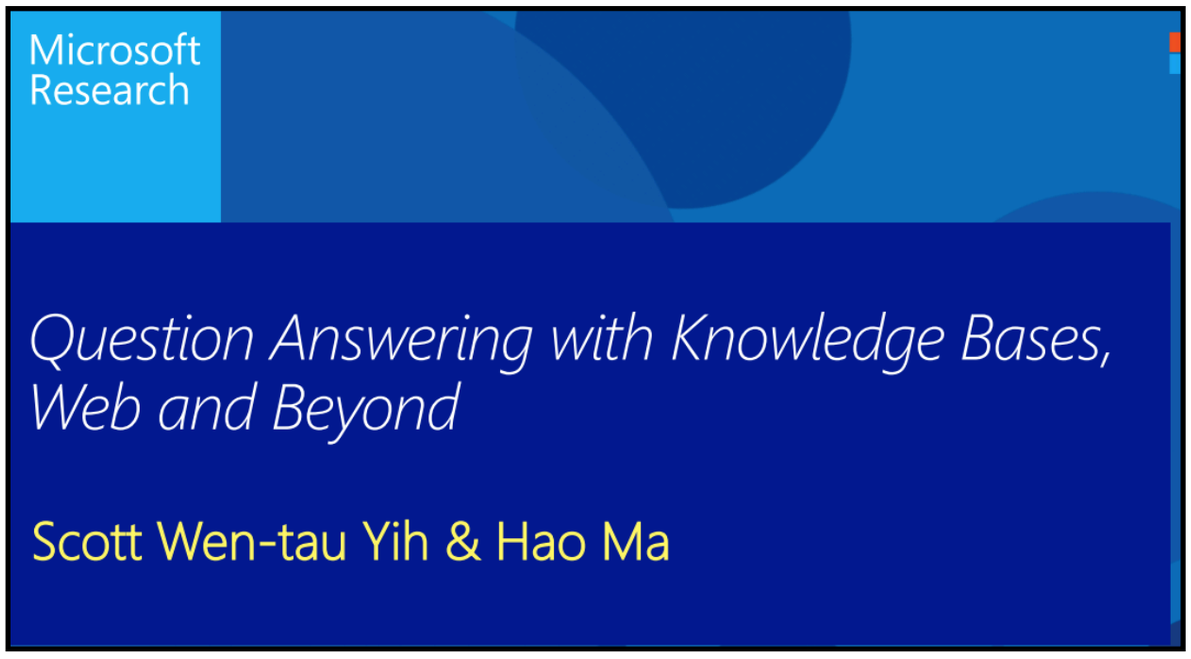
Natural Language Processing in Watson

Alfio M. Gliozzo, Aditya Kalyanpur, James Fan

IBM Watson system on Jeopardy questions

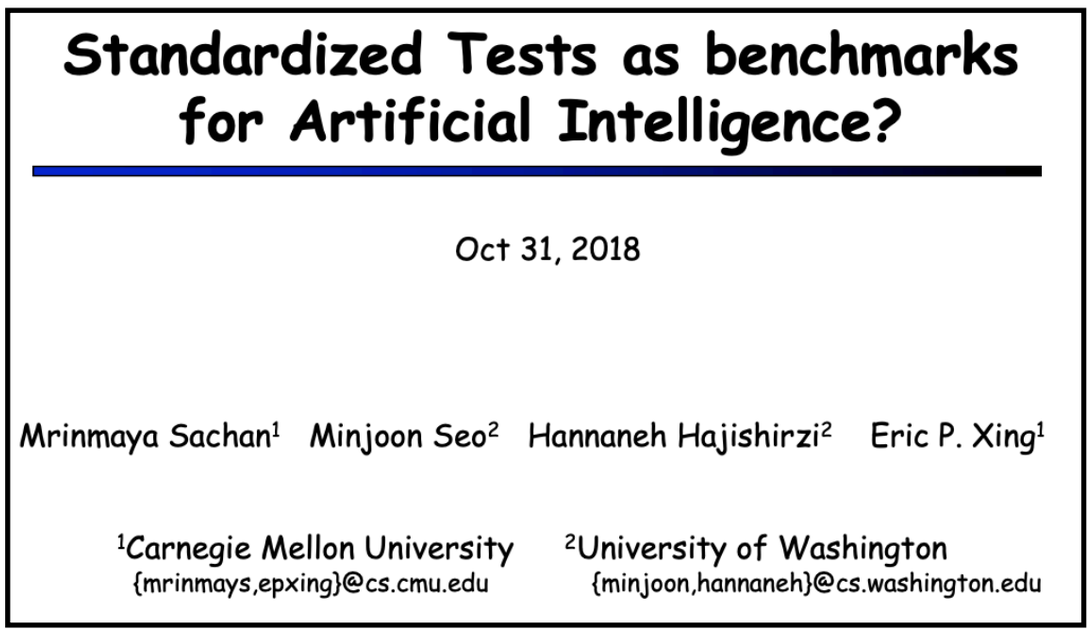
Why give this tutorial today?

NAACL 2016



QA with KBs and tables

EMNLP 2018

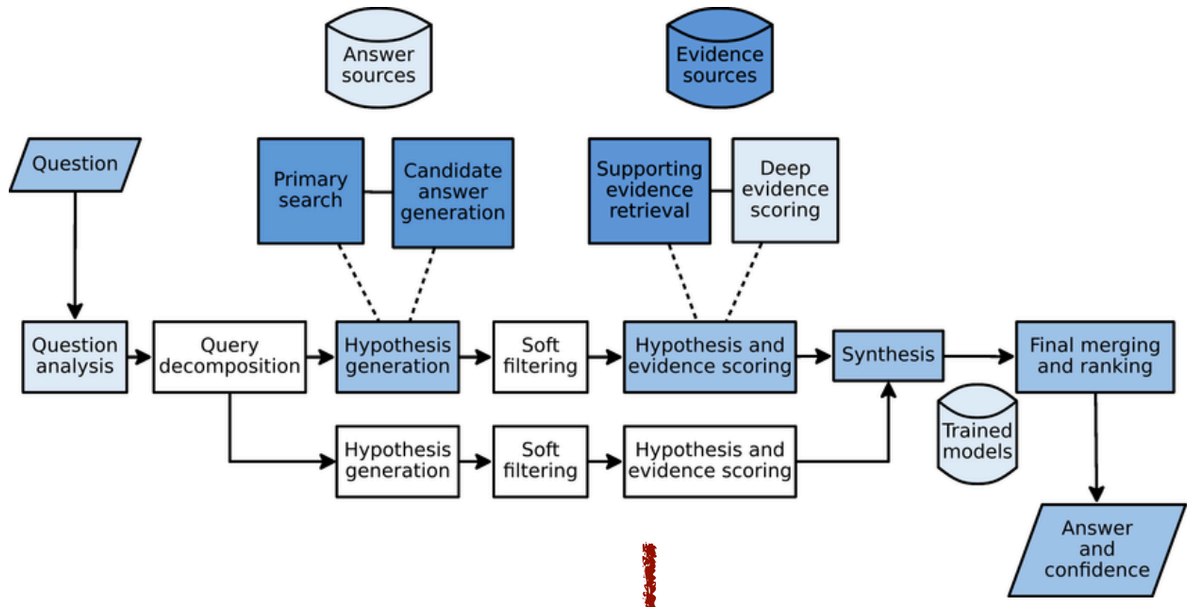


QA as standardized tests: reading comprehension and closed domain problems (science, geometry, algebra word etc)

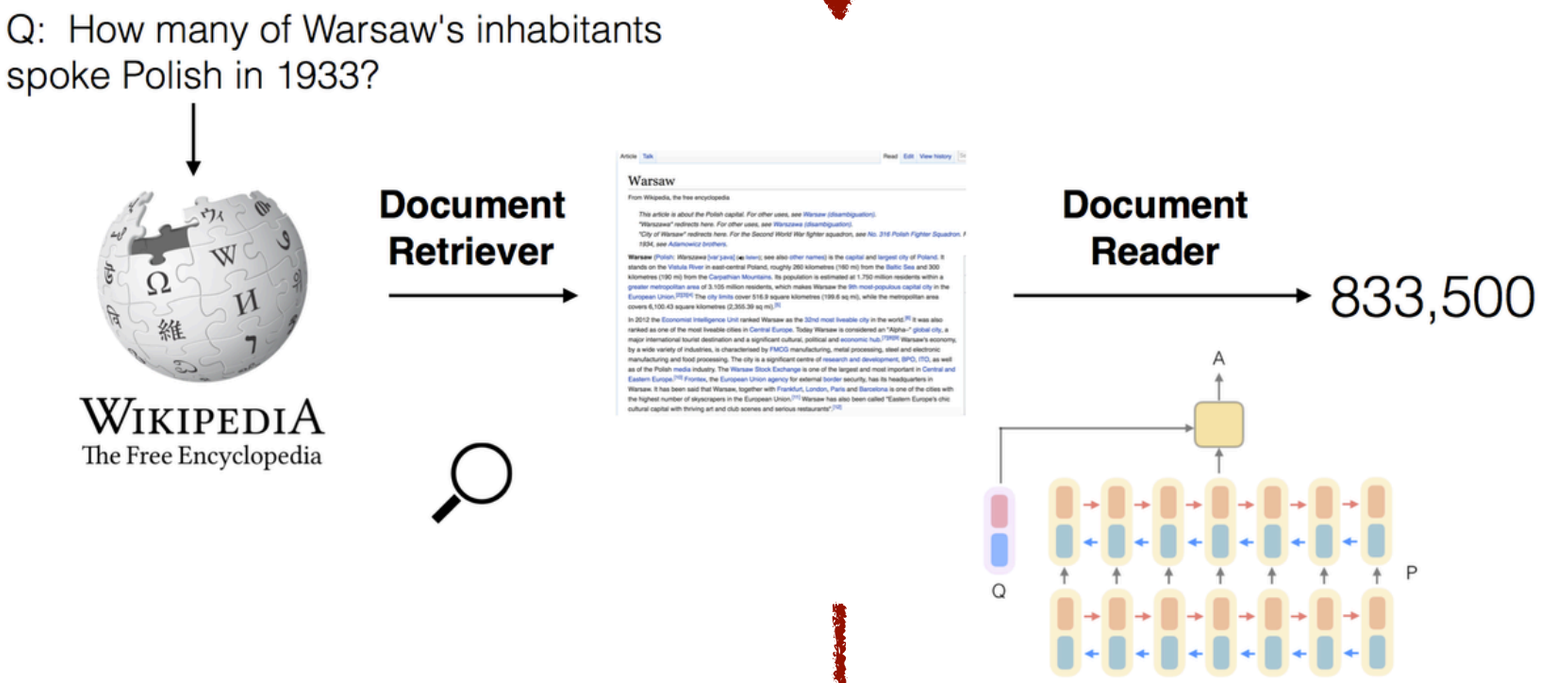
This tutorial: open-domain question answering over a large collection of unstructured documents; mostly the new generation and paradigm of the NLP technologies (2017-2020)

Why give this tutorial today?

Classical QA pipeline

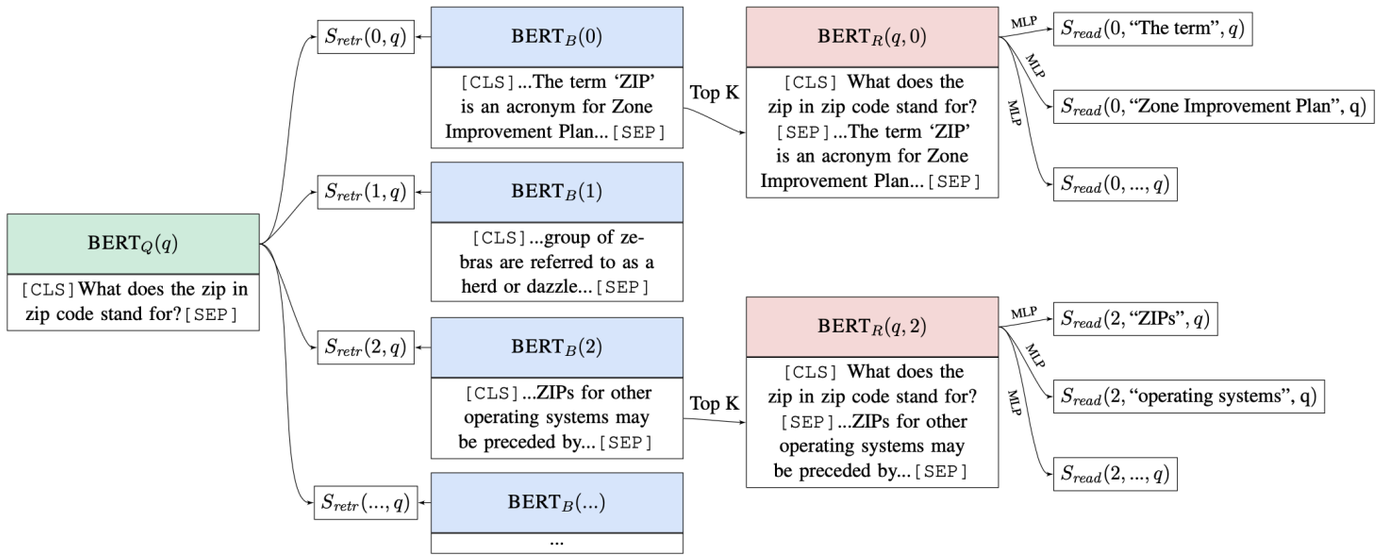


Two-stage Retriever-reader approaches

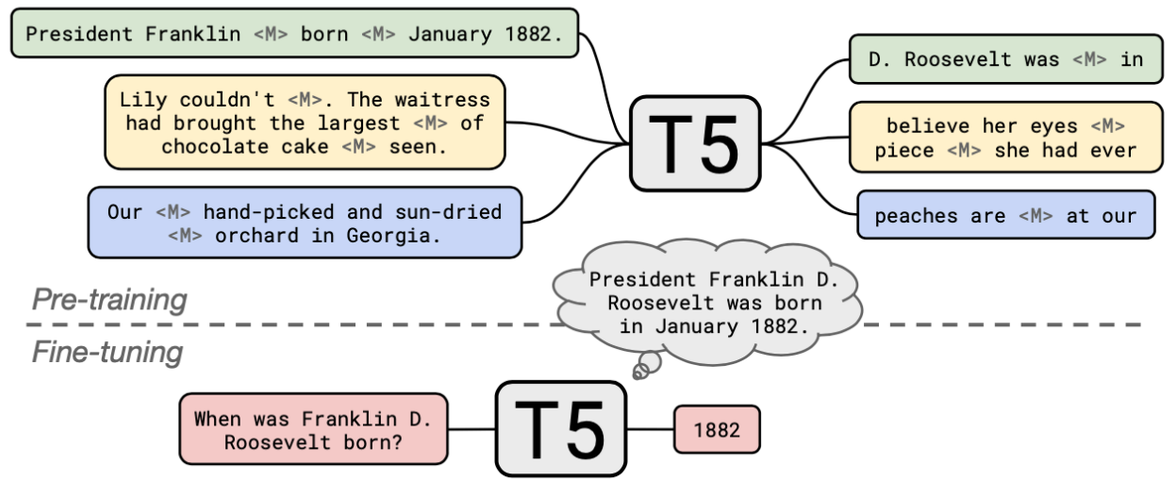


Why give this tutorial today?

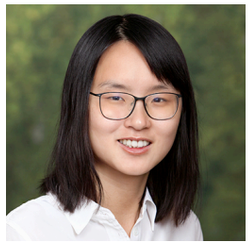
End-to-end learning



Retrieval-free models

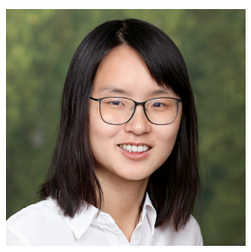


Outline



- Part 1. Introduction <- We are here!

- Part 2. A history of open-domain QA
- Part 3. Datasets & evaluation



- Part 4. Two-stage retriever-reader approaches

30min coffee break

- Part 5. Dense retriever and end-to-end training
- Part 6. Retrieval-free approaches

- Part 7. Open-domain QA using KBs and text
- Part 8. Open problems and future directions