NPFL123 Dialogue Systems

9. Voice Assistants & Question Answering

https://ufal.cz/npfl123

Ondřej Dušek, Simone Balloccu, Mateusz Lango, Kristýna Klesnilová & Jan Cuřín

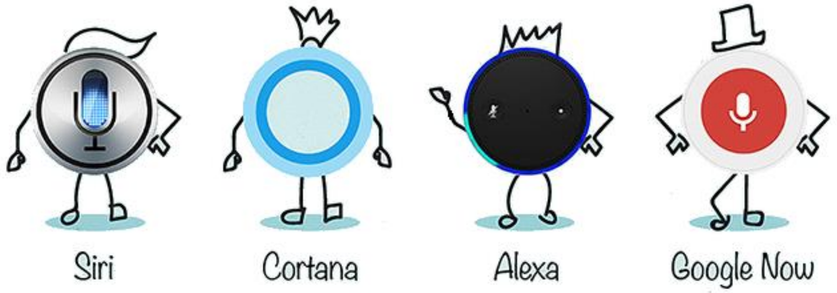
17.4.2024

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

CC BY SA
unless otherwise stated

Virtual Assistants (voice/smart/conversational assistants)

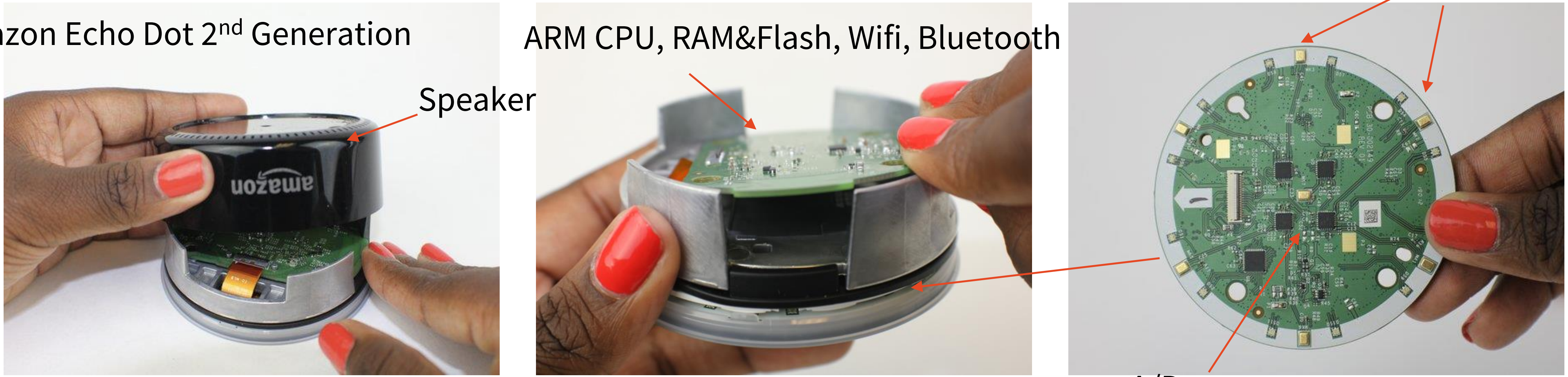
- “Definition”: voice-operated **software** (dialogue system) capable of **answering questions, performing tasks** & basic dialogue in **multiple domains**
- Apple Siri (2011) – question answering & iOS functions
- Every major IT company has/had them
  - Microsoft Cortana (2014-2023, now Copilot)
  - Amazon Alexa (2014)
  - Google Assistant (2016)
  - Samsung Bixby (2017)
  - Mycroft (now OpenVoiceOS), Rhasspy (open-source, 2018/2020)
  - Clova (Naver, 2017) – Korean & Japanese
  - Alice (Yandex, 2017) – Russian
  - DuerOS (Baidu, 2017), AliGenie (Alibaba, 2017) – Chinese



https://www.marketingdirecto.com/digital-general/digital/siri-cortana-alexa-origen-nombres-fruto-la-casualidad

Smart Speakers

- Internet-connected mic & speaker with a virtual assistant running
  - optionally video (display/camera)
  - ~ same functionality as virtual assistants in phones/computers
  - Amazon Echo (Alexa), Google Home (Assistant), Apple HomePod (Siri) […]
- Main point: multiple microphones – far-field ASR



https://www.ifixit.com/Device/Amazon_Echo_Dot_2nd_Generation

Capabilities

- Out of the box:
  - Question answering
  - Web search
  - News & Weather
  - Scheduling
  - Navigation
  - Local information
  - Shopping
  - Media playback
  - Home automation
- a lot of it through 3rd party APIs
- the domains are well connected

Google Nest

Amazon Echo

Apple HomePod

Demos

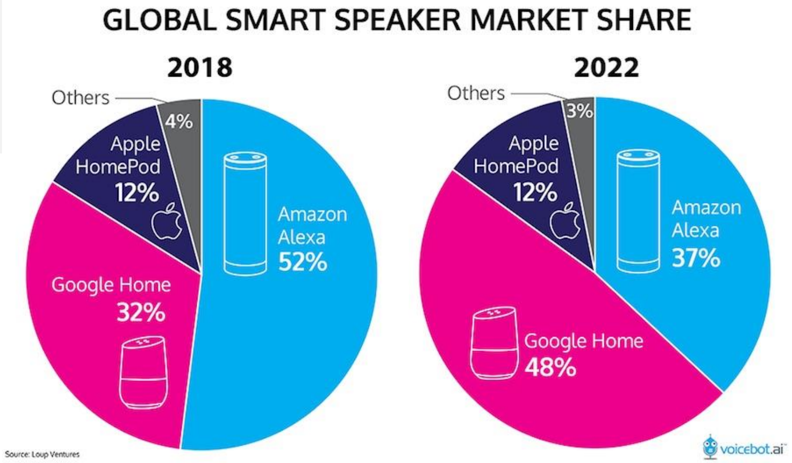
Raven H (powered by DuerOS, Baidu)

https://www.youtube.com/watch?v=iqMjTNjFIMk

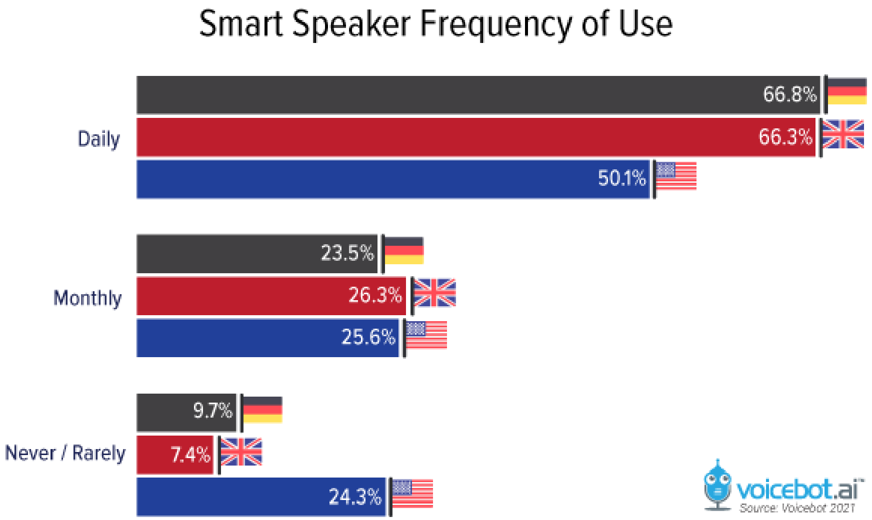
Google Assistant

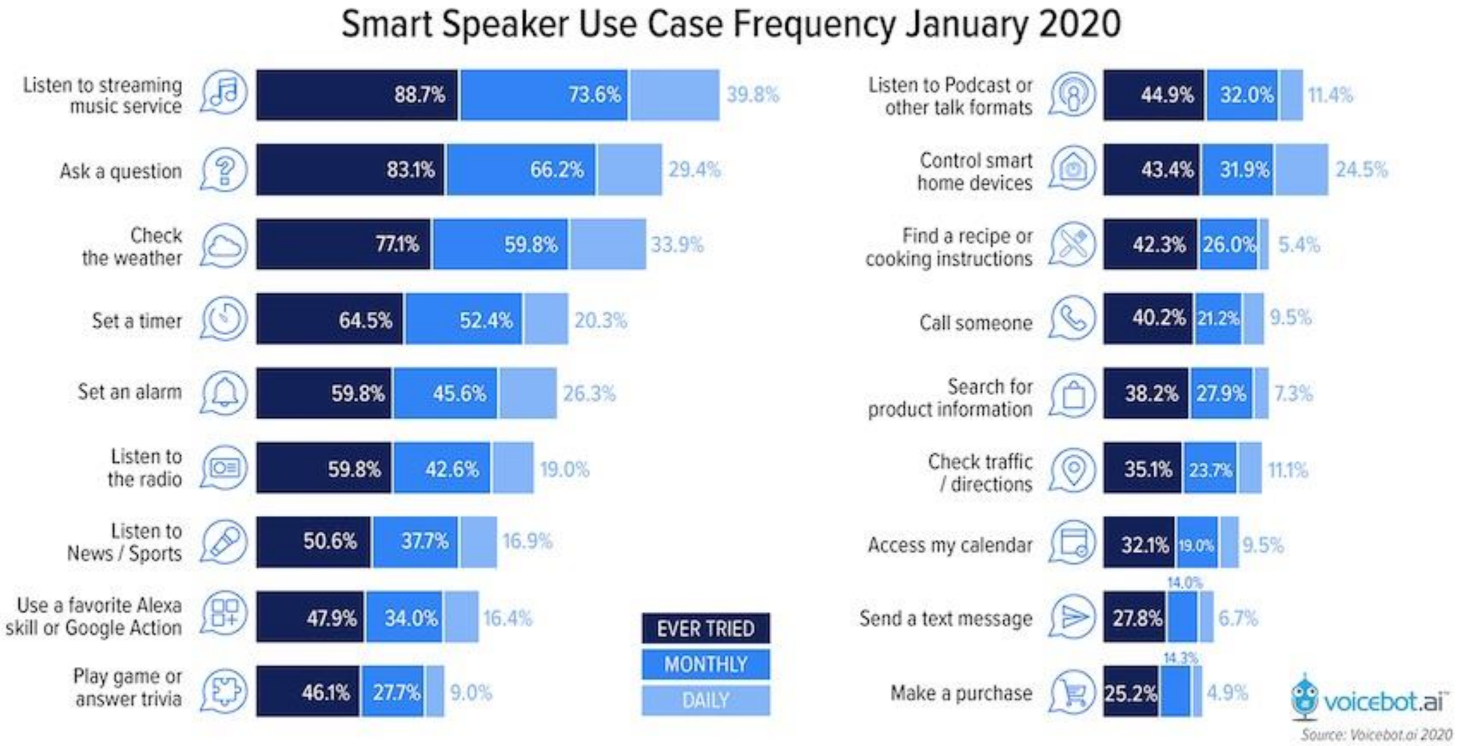
https://www.youtube.com/watch?v=JONGt32mfRY

Smart Speaker Adoption



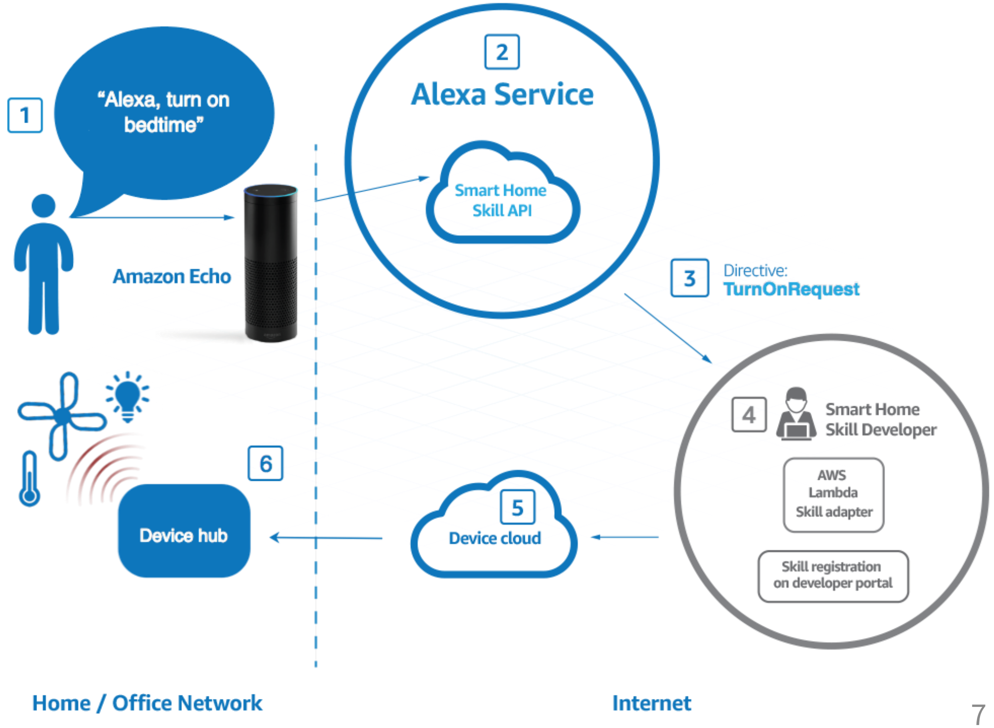
- >35% US/UK adults have a smart speaker
  - growth slowed, less adoption elsewhere (CZE – very low)
  - Amazon had an early lead, now it's more Google
- People really use them
  - early adopters – more intensively, correlated with phone assistant usage





How they work

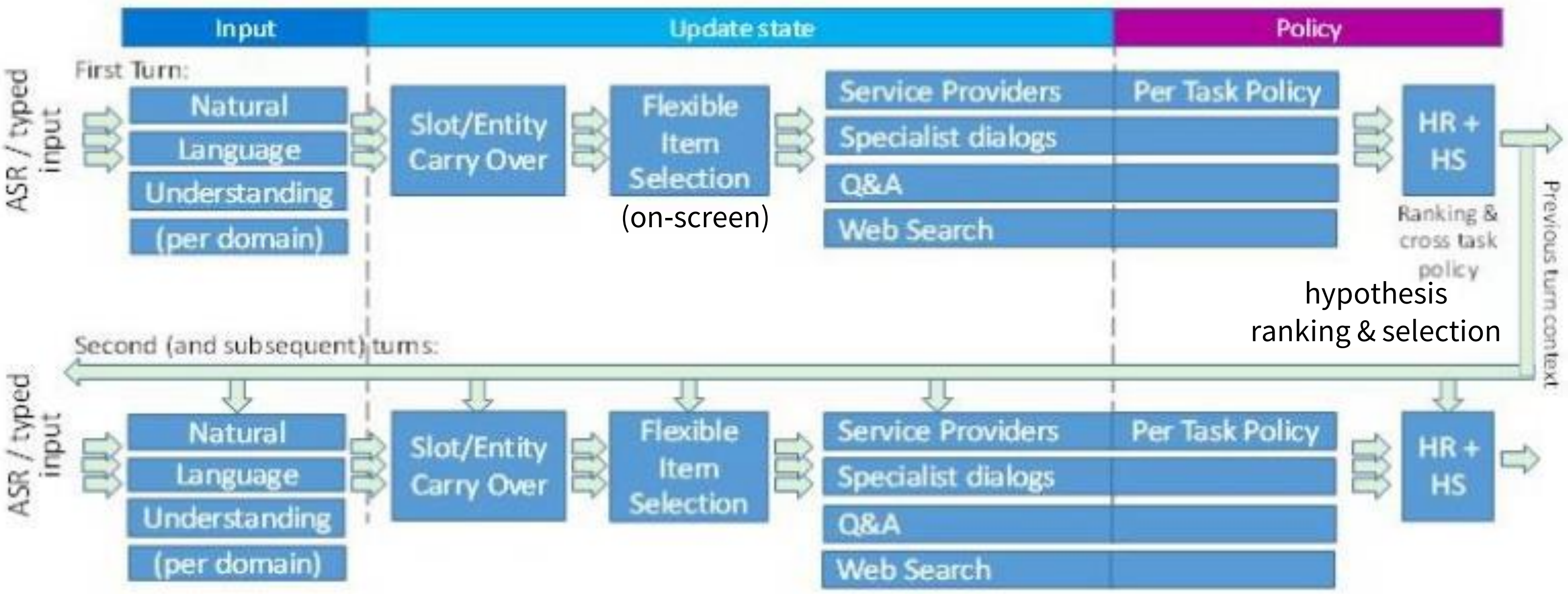
- Device listens for **wake word**
  - after the wake word, everything is processed in vendor's cloud service
  - raw **audio is sent to vendor**
  - follow-up mode – no wake word needed for follow-up questions (device listens for 5-10sec after replying)
  - privacy concerns
- **Intents** – designed for each domain
  - NLU trained on examples
  - DM + NLG handcrafted
  - extensible by 3rd parties (Skills/Apps)
- No incremental processing



https://developer.amazon.com/blogs/post/Tx38PSX7O9YKIK1/

How they work

- NLU includes **domain detection**
  - “web” domain as fallback
- Multiple NLU analyses (ambiguous domain)
  - resolved in context (hypothesis ranking)
- State tracker & coreference
  - Rules on top of machine learning
  - All per-domain



Cortana structure

(Sarikaya et al., 2016) https://ieeexplore.ieee.org/abstract/document/7846294

Why they are cool

- ASR actually impressive
  - NLU often compensates for problems
- Range of tasks is wide & useful
- 1st really large-scale
  dialogue system deployment ever
  - not just a novelty
  - actually boosted voice usage
    in other areas (phone, car etc.)

Assistants & Accents
https://youtu.be/gNx0huL9qsQ?t=41

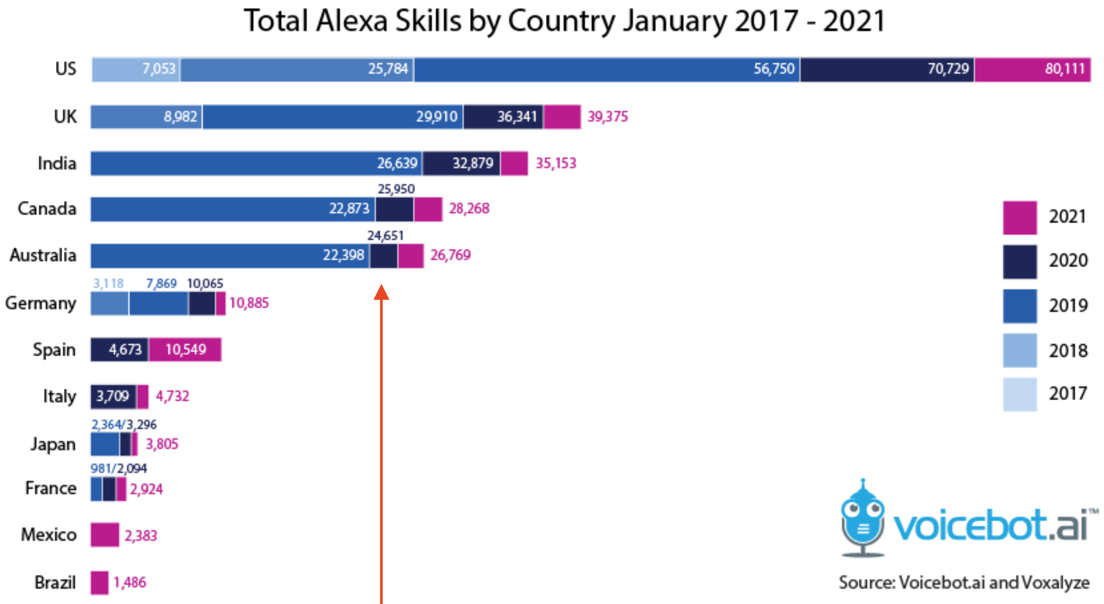
Why they are not so cool

- Still handcrafted to a large part
  - *conversational architects* are a thing now
- Not very dialogue-y
  - mostly just one turn, rarely more than a few
- Language limitations
  - only available in a few major languages (En, Zh, Jp, De, Es, Fr, Kr […])
- ASR still struggling sometimes
  - noise + accents + kids
  - not that far-field
  - helped a lot by NLU / domain knowledge

https://youtu.be/CYvFxs32zvQ?t=65

Adding Skills/Apps

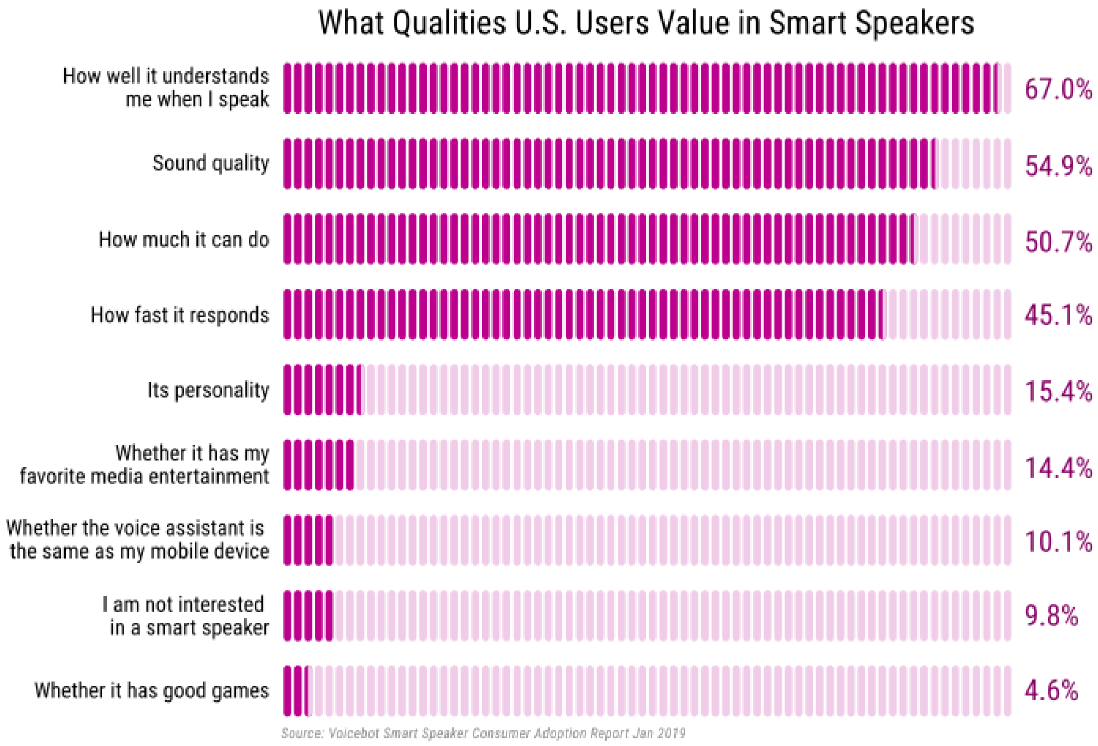
- Additional functionality by 3rd party developers
  - API/IDEs provided by vendors, enabled on demand (similar to installing phone apps)
- Not 1st-class citizens
  - need to be invoked specially
    - *Alexa, tell Pizza Hut to place an order*
    - *Alexa, ask Uber to get me a car*
  - much less used than the default ones
- There's thousands of them
  - many companies have a skill
  - many specific inventions
  - finance, fitness, food, games & trivia …
- Seem to go deprecated
  - few new skills, vendors dropping support



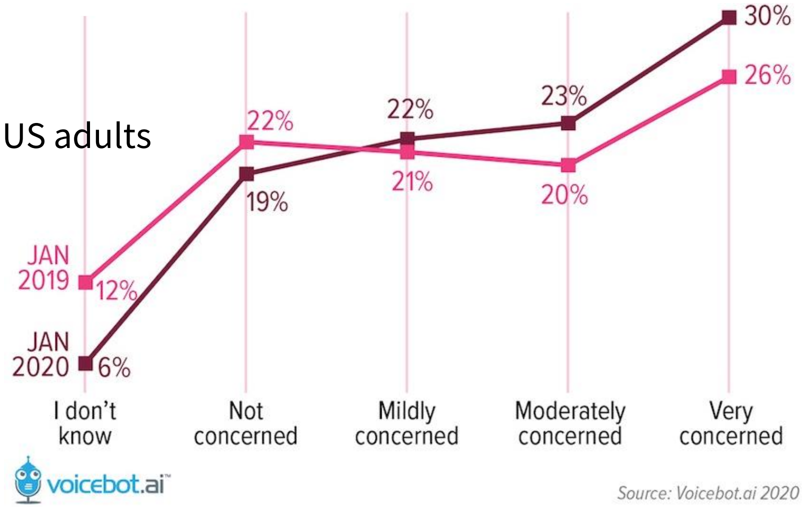
https://voicebot.ai/2021/01/14/alexa-skill-counts-surpass-80k-in-us-spain-adds-the-most-skills-new-skill-introduction-rate-continues-to-fall-across-countries/
https://arstechnica.com/gadgets/2024/04/amazon-virtually-kills-efforts-to-develop-alexa-skills-disappointing-dozens/ , https://9to5google.com/2022/06/13/google-assistant-voice-apps/

What people care about in smart speakers

- **Understanding, features, speed**
  - personality / dialogue not so much
  - 3rd party apps not so popular (should work out-of-the-box)
  - commerce not so popular, but growing
- QA: music, news, movies
- Privacy concerns don't stop people from buying/using smart speakers
  - privacy-conscious 16% less likely to own one



Privacy concerns in US adults



Question answering

- integral & important part of assistants
  - broadest domain available, apart from web search
- QA is not the same as web search
  - QA needs a specific, unambiguous answer, typically a (named) entity
    - person, object, location […]
    - ~ **factoid questions**
  - Needs to be within inference capabilities of the system

Who is the president of Germany? *How high is the Empire State Building?*	**x**	*Who is the best rapper?* *Who will become the next U.S. president?* *How much faster is a cheetah than an elephant?*

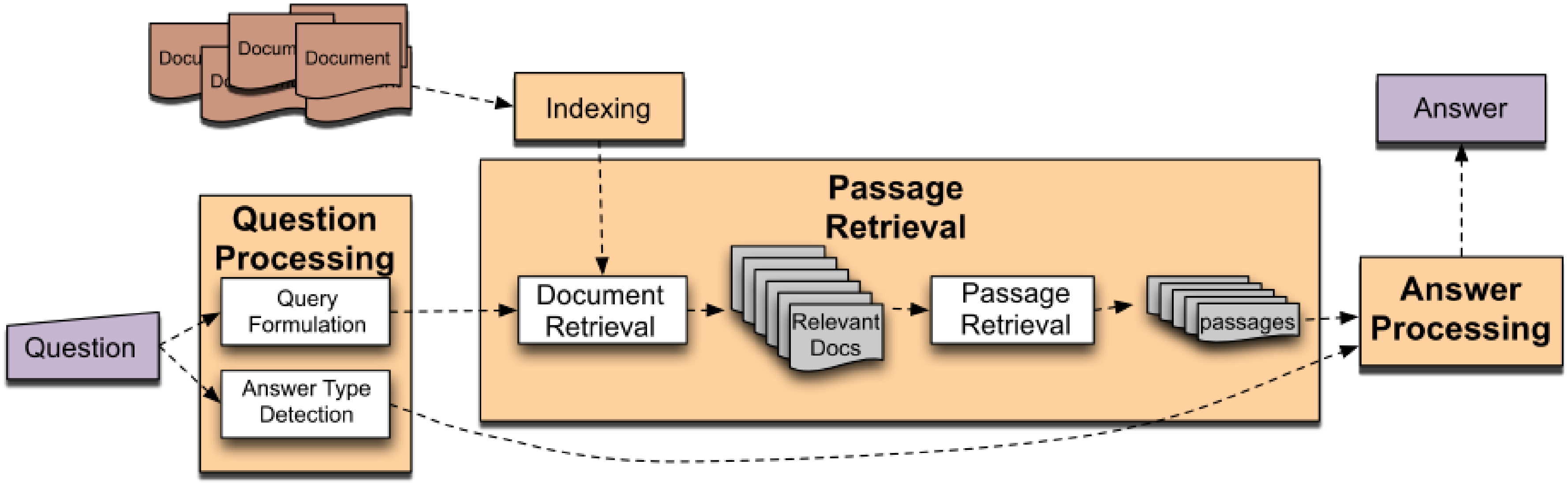
Web search

- Given a query, find best-matching **documents**
  - Over unstructured/semi-structured data (e.g. HTML)
- Basic search
  - Candidates: find matching word occurrences in index
  - Reranking: many features
    - Location of words (body, title, links)
    - Frequency of words (TF-IDF →)
    - Word proximity
    - PageRank – weighing links to documents/webpages (how many, from where)
  - 2^{nd} level: personalized reranking
- Query reformulation & suggestion

QA approaches

- **Information Retrieval**
  - Basically improved web search
  - IR + phrase extraction
    - getting not just relevant documents, but specific phrases within them
- **Knowledge Graphs**
  - KGs – storage of *structured* information
  1) Semantic parsing of the query
  2) Mapping to KG(s)
- **Hybrid** (IBM Watson, probably most other commercial systems)
  - candidates from IR
  - reranking using KGs/semantic information

IR-based QA Pipeline

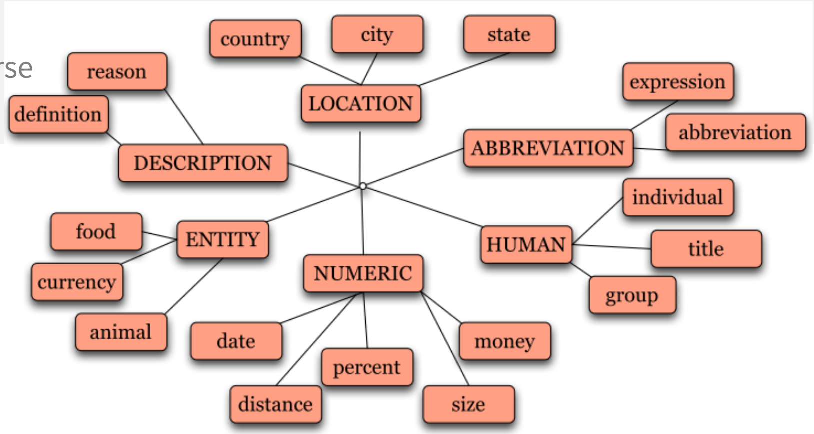


like web search | additional

from Jurafsky & Manning
QA slides, Coursera NLP course

Question Processing

Jurafsky & Manning
QA slides, Coursera NLP course



- **Answer type detection**
  - what kind of entity are we looking for?
  - rules / machine learning (with rules as features)
  - rules: regexes
    - headword = word right after wh-word

Who is the […] composer/football player […]
Which city is the largest […]

- **Named entity recognition**
- **IR Query formulation** – keyword selection
  - ignore stop words (*the, a, in*)
  - prioritize important words (named entities)
  - stemming (remove inflection)
- **Question type classification** – definition, math…
- **Focus detection** – question words to replace with answer
- **Relation extraction** – relations between entities in question
  - more for KGs, but can be used for ranking here

IR Document Retrieval

- Candidates – find matching words in index (same as web search)
- Weighting
  - Frequency: **TF-IDF (term frequency-inverse document frequency)**
    - TF – document more relevant if term is frequent in it
    - IDF – document more relevant if term only appears in few other documents

$$\mathrm{TFIDF} = \underbrace{(1 + \log \underbrace{f_{t,d}}_{\text{\# times } t \text{ appears in } d})}_{\text{TF (log-scaled)}} \cdot \underbrace{\log \frac{N}{n_t}}_{\text{IDF}}$$

total # of documents

# of documents containing *t*

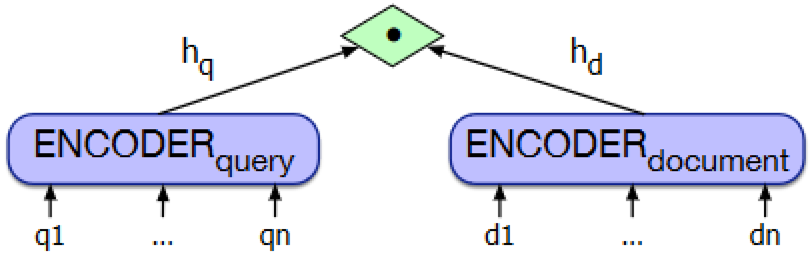
    - this is just one of many variants

  - Other metrics – **BM25** – more advanced smoothing, heeds document length
  - Proximity: also using n-grams in place of words

IR Passage Retrieval

- Passage **segmentation** – split document into ~paragraphs
  - anything short enough will do
- Passage **ranking** – typically machine learning based on:
  - named entities & their type (matching answer type?)
  - # query words contained
  - query words proximity
  - rank of the document containing passage
- **Neural ranking**: 2x Transformer LM (BERT/SBERT) + dot product
  - or cosine similarity (~+normalization)
  - no need for specific features
  - alt: 1 transformer, feed both & classify

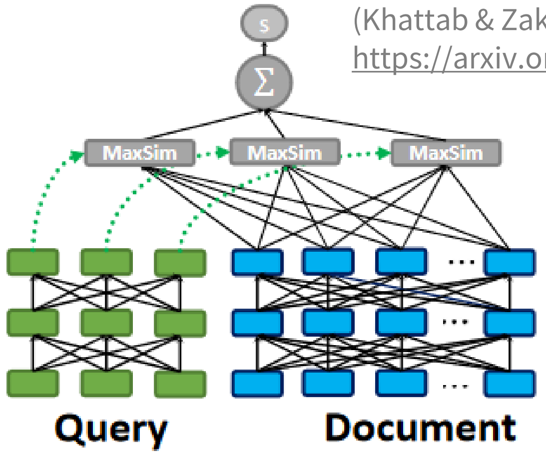
(Reimers & Gurevych, 2019)
https://aclanthology.org/D19-1410/



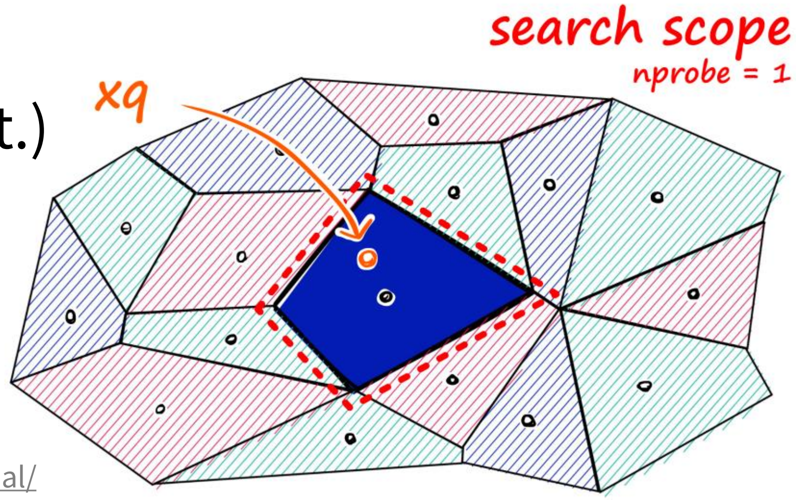
(Jurafsky & Martin, 2023)
https://web.stanford.edu/~jurafsky/slp3/14.pdf

Dense Retrieval

- Working with a **neural-ranking-like approach on the whole data**
  - less focus on words, more on semantics/embeddings
- **Precompute** & store all document embeddings
  - compare via cosine similarity to query embeddings
- Less accurate than full (S)BERT finetuning
  - but that wouldn't be viable over large data
  - **ColBERT**: compromise
    – token embeddings & compute + aggregate similarities
- Larger-scale: clustering (**Faiss**)
  - cluster embeddings into Voronoi cells (centroids & L2 dist.)
  - only search in the closest cell
  - & some other efficiency tricks (e.g. quantization)



(Khattab & Zakharia, 2020)
https://arxiv.org/abs/2004.12832



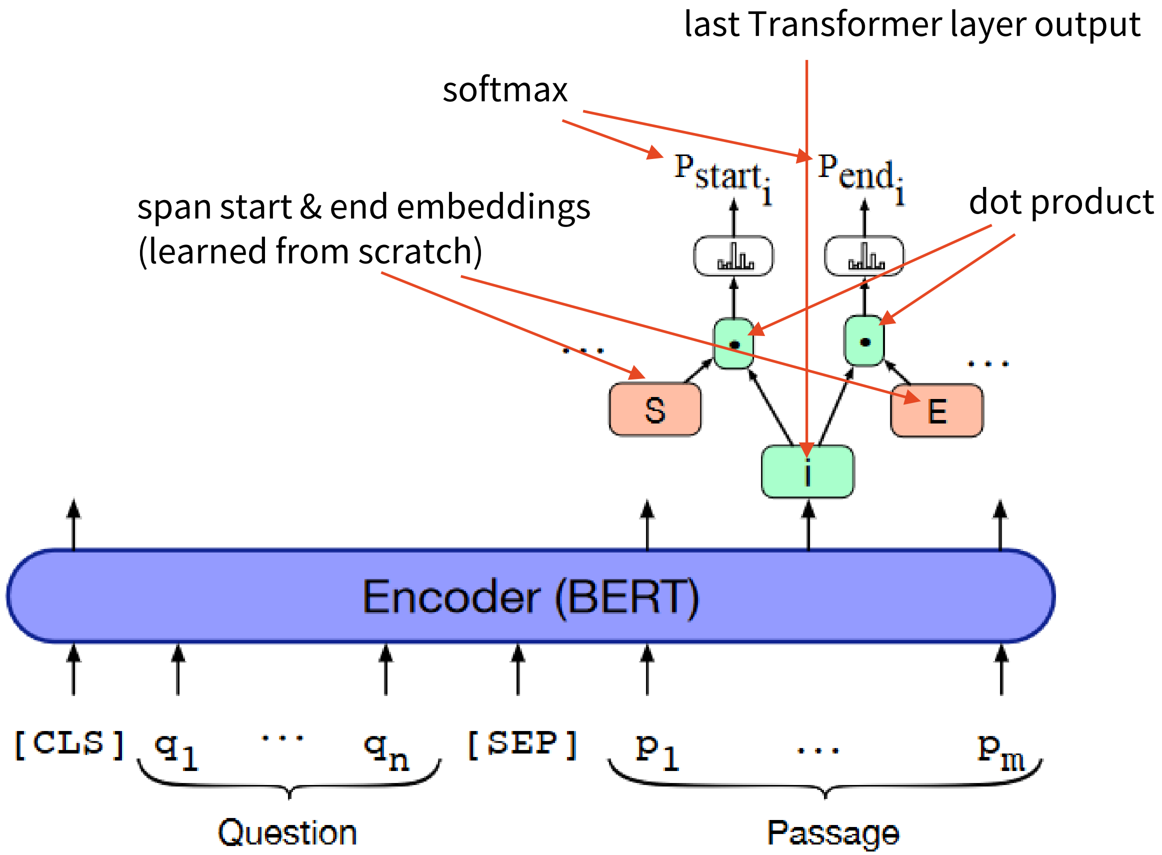
https://github.com/facebookresearch/faiss
https://www.pinecone.io/learn/series/faiss/faiss-tutorial/

IR Answer Extraction

- **NER on passages** – looking for the right answer type
- 1 entity found → done
- More entities present → needs **another ranking**, based on:
  - answer type match
  - distance from query keywords in passage
  - novelty factor – not contained in query
  - position in sentence
  - semantic parse / relation
  - passage source rank/reliability

Neural answer extraction

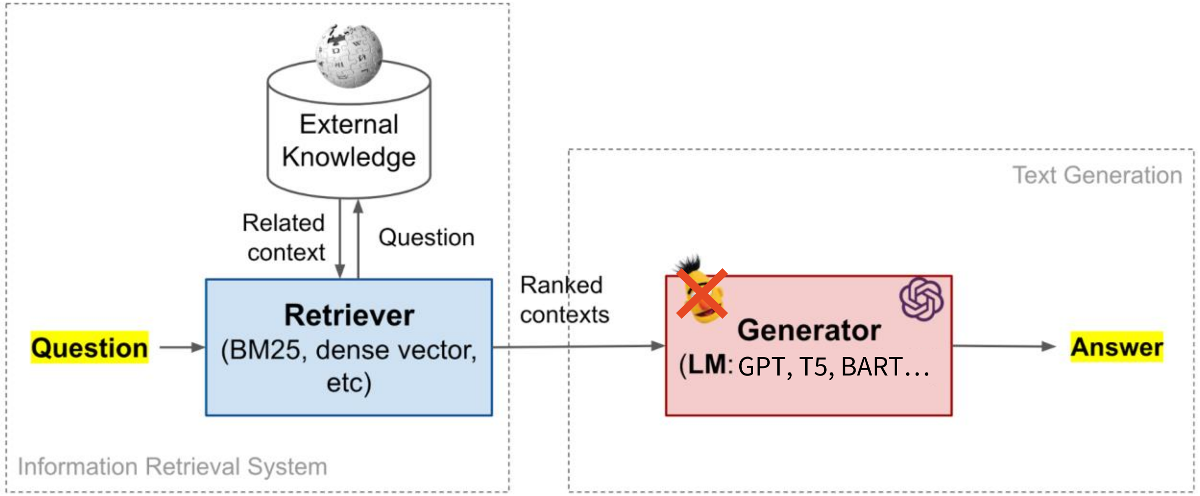
- Feed in question + extracted passage(s) to a Transformer model
  - typically a pretrained LM (e.g. BERT)
- 2 classifiers: start + end of answer span
  - softmax over passage(s) tokens
- NB: LLMs (ChatGPT) do no retrieval!
  - just generate reply from scratch
  - doesn't work well, not designed for QA
- alternative: generative QA
  - feed in passage
  - generate reply word-by-word (see NLG)



(Jurafsky & Martin, 2023)
https://web.stanford.edu/~jurafsky/slp3/14.pdf

Retrieval-augmented Generation QA

- Not just extraction, but full-sentence answer formulation
- Transformer generative (L)LMs
  - decoder models
  - input: retrieved passage
  - output: full-sentence response
- Train/prompt to provide reply
  - avoid hallucination
  - avoid copying everything verbatim

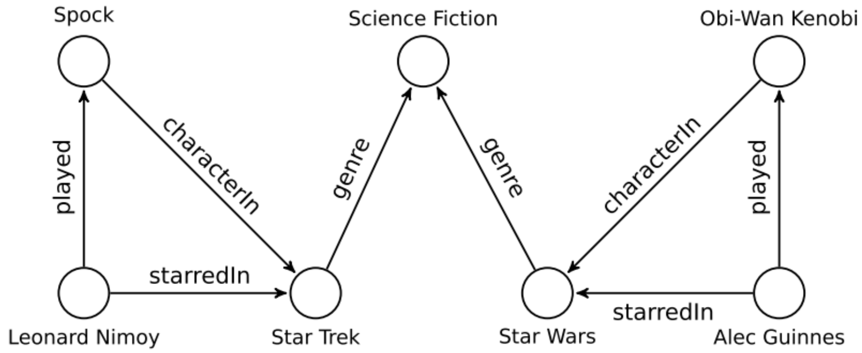


https://lilianweng.github.io/posts/2020-10-29-odqa/

- Retriever & generator can be trained jointly (Lewis et al., 2020) https://arxiv.org/abs/2005.11401
- Option: ask LM if the retrieved is relevant, then generate (Wang et al., 2023) https://arxiv.org/abs/2309.02233
- Option: ask LM to link to sources (Chen et al., 2023) https://arxiv.org/abs/2310.12150

Knowledge Graphs

- Large repositories of **structured, linked** information
  - **entities** (nodes) + **relations** (edges)
  - typed (for both)
  - entity/relation types form an **ontology** (itself a similar graph)



from Jens Lehman's QA keynote

- Open KGs (millions of entities, billions of relations)
  - Freebase (freely editable, many sources, bought by Google & shut down)
  - DBPedia (based on Wikipedia)
  - Wikidata (part of Wikipedia project, freely editable)
  - Yago (Wikipedia + WordNet + GeoNames)
  - NELL (learning from raw texts)
- Commercial KGs: Google KG, Microsoft Satori, Facebook Entity Graph
  - domain specific: Amazon products, Domino's pizza […]

RDF Representation

- RDF = Resource Description Framework
  - Most popular KG representation
  - Wikidata – different format but accessible as RDF
- **Triples**: <subject, predicate, object>
  - predicate = relation
  - subject, object = entities
  - can also include relation confidence (if extracted automatically)
- Entities & relations typically represented by URI (not always)
  - objects can also be constants (string, number)

subject: *Leonard Nimoy*
predicate: *played*
object: *Spock*
[confidence: *0.993*]

SPARQL

- Query language over RDF databases
  - relatively efficient
  - can query multiple connected triples (via ?variables)
- can be used directly
  - if you know the domain/application
- QA – need to map user question to this
  - or use IR-based methods instead

Wikidata: largest cities with female mayors
https://query.wikidata.org/

```
SELECT DISTINCT ?city ?cityLabel ?mayor ?mayorLabel
WHERE
{
  BIND(wd:Q6581072 AS ?sex)
  BIND(wd:Q515 AS ?c)

    ?city wdt:P31/wdt:P279* ?c .  # find instances of subclasses of city
    ?city p:P6 ?statement .            # with a P6 (head of goverment) statement
    ?statement ps:P6 ?mayor .          # ... that has the value ?mayor
    ?mayor wdt:P21 ?sex .       # ... where the ?mayor has P21 (sex or gender) female
    FILTER NOT EXISTS { ?statement pq:P582 ?x }  # ... but the statement has no P582 (end date) qualifier

    # Now select the population value of the ?city
    # (wdt: properties use only statements of "preferred" rank if any, usually meaning "current population")
    ?city wdt:P1082 ?population .
    # Optionally, find English labels for city and mayor:
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en" .
    }
}
ORDER BY DESC(?population)
LIMIT 10
```

KG Retrieval

- Problem: **synonymy** – many ways to ask the same question
  - RDF relations have a specific surface form (not just *wd:1234*)
  - needs normalization/lexical mapping/usage of synonyms
    - WordNet expansion
    - stemming/lemmatization
    - multiple labels for entities/relations
    - string similarity/word embeddings

How fast do jaguars run?
What is a top speed of a jaguar?

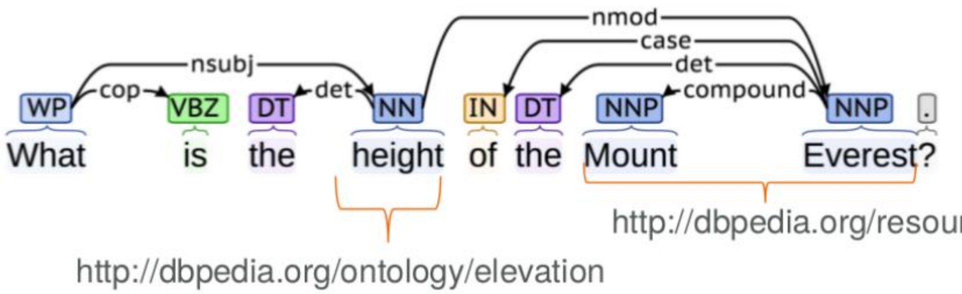
- Problem: **ambiguity**
  - needs entity/relation disambiguation/grounding/linking (to KG-compatible URIs)
  - context used to disambiguate
    (neighbour words, syntax, parts-of-speech)
  - KG itself used – closest/semantically related entities

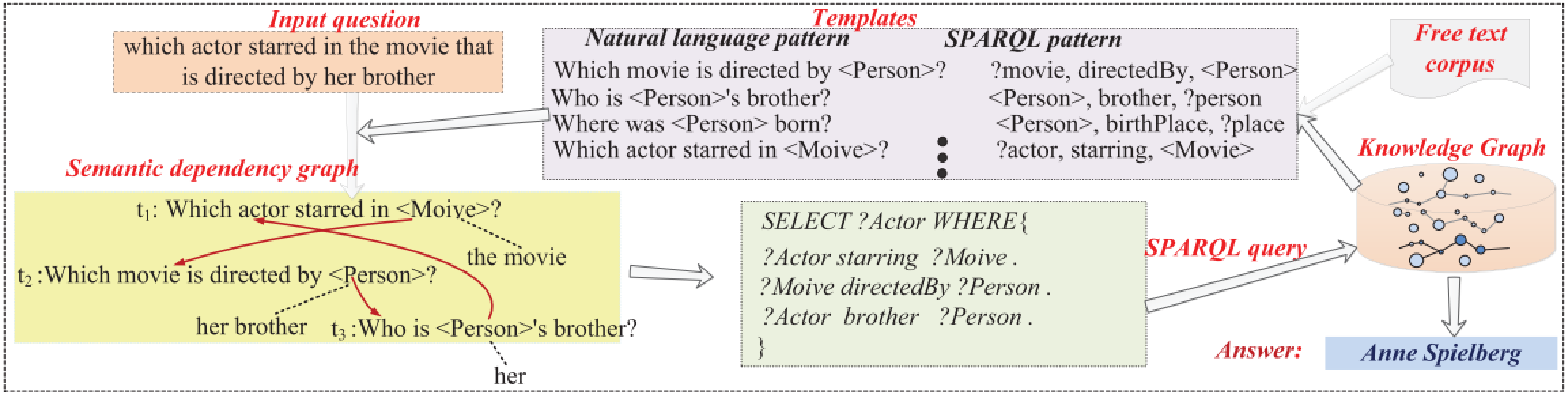
How fast is a Jaguar [I-Pace]?

KG Retrieval

- **Semantic parsing** can be used for query normalization
- Dependencies help decompose complex questions
  - Doesn't have to be syntactic dependencies
  - Template mapping:
    map simple question patterns
    that have SPARQL equivalents

from Jens Lehmann's QA keynote





(Zheng et al., 2018)
http://www.vldb.org/pvldb/vol11/p1373-zheng.pdf

KG Maintenance

- Information needs to be up-to-date
- Deduplication
- Ontology changes
  - need to version ontologies (and data)
    (for new/split/merged entity & relation types)
- Integrating multiple KGs
  - larger world knowledge coverage
  - company suppliers, mergers
  - → ontology bridging/mapping needed

"Basically, we're all trying to say the same thing."

http://dit.unitn.it/~accord/RelatedWork/Matching/Noy-MappingAlignment-SSSW-05.pdf

from Alex Marin's KG QA slides

Ontology mapping

- Mismatch types
  - different labels (easiest)
  - same term, different thing & vice-versa
  - different modelling approaches (e.g. subclass or property?)
  - different granularity (more/less subclasses)
- Mappings
  - handcrafted (best results, but expensive)
  - rule-based – map into a common ontology
    - string distances, WordNet
  - graph-based – compare ontology structure
  - machine learning

Summary

- Virtual assistants/smart speakers are booming
  - large variety of tasks, interconnected
  - most part of the processing happens online
  - impressive ASR, typically handcrafted dialogue policy, NLG
- Question answering – **factoids**
  - **IR approaches**: word-based document retrieval, passage extraction, ranking
    - **TF-IDF** & co. for retrieval, answer type selection
    - **dense retrieval** using vector representation & similarity
    - **ranking** with word features or NNs
    - **generative QA** – retrieve passages & compose reply with LM
  - **KG approach**: semantic parsing & mapping to SPARQL queries
    - **RDF** triple representations

Thanks

Contact us:

https://ufaldsg.slack.com/
{odusek,schmidtova,hudecek}@ufal.mff.cuni.cz
Skype/Meet/Zoom (by agreement)

Labs in 10 mins

Get the slides here:

http://ufal.cz/npfl123

References/Further:

- Dan Jurafsky & Chris Manning's slides at Stanford/Coursera: https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html
- Alex Marin's slides at Uni Washington: https://hao-fang.github.io/ee596_spr2018/
- Anton Leuski's slides at UCSC: http://projects.ict.usc.edu/nld/cs599s13/
- VoiceBot smart speaker report: https://voicebot.ai/smart-speaker-consumer-adoption-report-2019/
- Jens Lehmann's keynote: http://jens-lehmann.org/files/2017/fqas_keynote.pdf
- Wikipedia pages of the individual KGs, assistants + Smart_speaker, Okapi_BM25, TF-IDF