

Natural Language Processing Applications



BITS Pilani
Pilani Campus

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



Session 1

These slides are prepared by the instructor, with grateful acknowledgement of James Allen and many others who made their course materials freely available online.

Session Content

- Objective of course
- What will we learn in this course?
- Text books and Reference books
- Evaluation Plan
- Application areas of Natural Language Processing



Objective of course

CO1	Identify and describe algorithms for real life NLP Applications
CO2	Demonstrate understanding of algorithms by using different NLP tools
CO3	Apply NLP techniques in state of art applications like Machine Translation, Information Extraction including Named entity recognition and Relation extraction
CO4	Evaluate different approaches of implementing NLP applications along with ethical considerations

What you will learn in this course

- **Grammar and spellcheckers**
- **Question Answering and Conversational AI**
- **Knowledge Graph Applications**
- **Machine Translation**
 - Statistical
 - Neural
 - Indic Languages
- **Information Extraction**
 - Named Entity Recognition
 - Relation Extraction
 - Extracting Events and Time
- **Sentiment Analysis**
 - Sentiment Analysis Methods
 - Neural Networks for Sentiment Analysis

All above interesting and important real world applications will be discussed with case study and implementations in the respective modules.

Text books and Reference books

T1	Speech and Language processing: An introduction to Natural Language Processing, Computational Linguistics and speech Recognition by Daniel Jurafsky and James H. Martin
----	---

R1	Manning and Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA
----	---

R2	Neural Machine Translation by Philipp Koehn
----	---

R3	Knowledge Graphs Methodology, Tools and Selected Use Cases by Dieter Fensel , Umutcan Şimşek, Kevin Angele, Elwin Huaman , Elias Kärle , Oleksandra Panasiuk , Ioan Toma, Jürgen Umbrich, and Alexander Wahler, Springer 2019
----	---

R4	Natural Language Toolkit. Bird and Loper, and other developers. Available for free at: – http://www.nltk.org/
----	--

Evaluation Plan

Name	Weight
Quiz (best 2 out of 3)	10%
Assignment 1 and 2	30%
Mid-term Exam	30%
End Semester Exam	30%

What is Natural Language Processing?

- Analyze, understand and generate human languages just like humans do
 - Applying computational techniques to language domain
 - To explain linguistic theories, to use the theories to build systems that can be of social use
 - Make computers learn our language rather than we learn theirs.
-

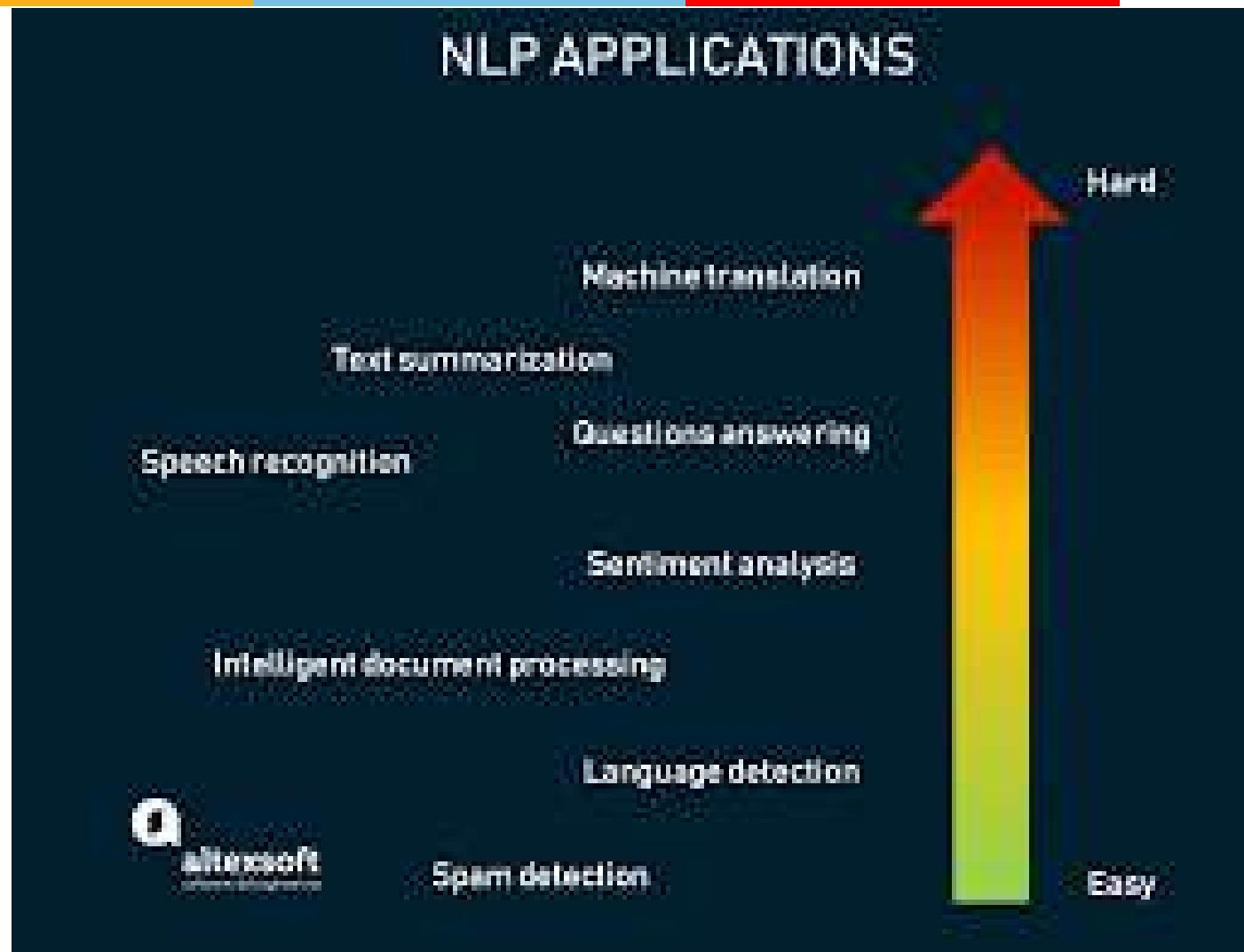
NLP Tasks and Applications

It's a big world out there



And everyone uses language

NLP Applications and Difficulty Level



NLP Applications

- Question answering
 - Who is the first Taiwanese president?
 - Text Categorization/Routing
 - e.g., customer e-mails.
 - Text Mining
 - Find everything that can be done with NLP
 - Machine (Assisted) Translation
 - Language Teaching/Learning
 - Usage checking
 - Spelling correction
 - Is that just dictionary lookup?
-

Application areas

- [Text-to-Speech & Speech recognition](#)
- [Natural Language Dialogue Interfaces to Databases](#)
- [Information Extraction](#)
- [Document Image Analysis](#)
- [Automatic Summarization](#) (<https://pypi.org/project/sumy/>)
- [Text Proof-reading – Spelling & Grammar](#)
- [Machine Translation](#)
- [Fake News](#) and [Cyberbullying Detection](#)
- [Monitoring Social Media](#)
- [Plagiarism detection](#)
- [Question Answering System](#) (<https://haystack.deepset.ai/>)
- Sentiment Analysis (<https://komprehend.io/sentiment-analysis>)

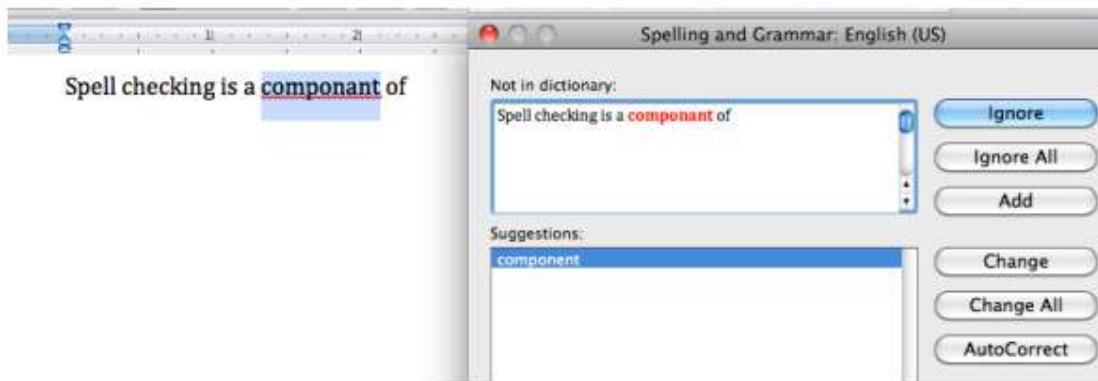
NLP Applications span across domains like Healthcare, Finance, Manufacturing and many more

Grammar and spellcheckers

Spell check

Applications for spelling correction

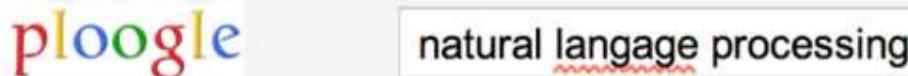
Word processing



Phones



Web search



2

Showing results for [natural language](#) processing
Search instead for [natural langage](#) processing

<https://languagetool.org/>

Spell check and Grammar check

Problem:

... and he fired presidential **aid/aide** Dick Morris after ...

⇒ *aid* or

⇒ *aide*

Training Data:

Spelling	Context
(1) aid " "	... and cut the foreign <i>aid/aide</i> budget in fiscal 1996 they offered federal <i>aid/aide</i> for flood-ravaged states ...
(2) aide " "	... fired presidential <i>aid/aide</i> Dick Morris after and said the chief <i>aid/aide</i> to Sen. Baker, Mr. John ...

Test Data:

Spelling	Context
???	... said the longtime <i>aid/aide</i> to the Mayor of St. ...
???	... will squander the <i>aid/aide</i> it receives from the ...

Case Study: AI-Powered



Communication Enhancement

Challenges

- Traditional tools often miss contextual errors (e.g., *your* vs. *you're*, *affect* vs. *effect*).
- Writers need more than just error correction; they need help communicating effectively and appropriately for their audience and context.

SOLUTION

- **Hybrid Approach:** Combines rule-based checks, statistical NLP (analyzing patterns in vast text data), and advanced DL/Transformers/Agentic AI
- **Contextual Understanding:** AI models analyze the entire sentence and surrounding text to detect subtle errors and suggest contextually appropriate words.
- **Beyond Grammar:** Employs NLP to analyze tone (e.g., confident, formal, friendly), suggest improvements for clarity and conciseness, check for plagiarism, and offer full-sentence rewrites.
- **Continuous Learning:** Models are constantly updated based on user interactions and evolving language patterns

Question Answering and Conversational AI

Case Study: AI-powered Question Answering (QA) system



- 24/7 Demand, High volume, Long wait times
- **Training:** The AI was trained to accurately handle the top 100 most frequent customer questions.
- **Integration:** The QA chatbot was launched, acting as the *first point of contact* for all support inquiries.
- **Handoff:** A seamless "handoff" protocol was created, allowing the bot to escalate complex or sensitive issues to a live human agent, along with the full chat history.

AI Assistants

Notification Assistant



Hi there - just a friendly reminder that your insurance policy expires in a month. Make sure to renew it in our member portal.

FAQ Assistant



I need to renew my renters insurance. How much will it be?



You can calculate your renewal price on our website here:
xyz.com/renew

Contextual Assistant



I need to renew my renters insurance. How much will it be?



I'd be happy to check for you. Firstly, are you still living in the same apartment?

Yes



Great - so just confirming it's 980 sq ft?

Yes



Thanks! Your new rate from September 1st onwards would be \$10 / month.



Would you like me to renew your policy for you right now?

Sure



Great. I've sent you a confirmation to your email.

AI Assistants

Personalized Assistant

- Assistant knows you much more in detail
- Quickly checks a few final things before giving you a quote tailored to your actual situation.



I can see your details are almost the same, except now you might want coverage for your new laptop. Additional coverage is only \$4 a month more for full coverage. Sound ok?

Sounds good!



Autonomous Organization of Assistants

- Group of AI assistants that know every customer personally
- Eventually run large parts of company operations—from lead generation over marketing, sales, HR, or finance





Case Study: Automating Customer Support with NLP Chatbots

- **Background**
 - HDFC Bank, one of India's largest private sector banks, was facing increasing customer service demands. Traditional support channels (phone/email) were overburdened, resulting in delayed responses and reduced customer satisfaction.
 - **Problem**
 - High customer query volume (especially repetitive questions)
 - Long response times
 - High operational costs for customer support
 - **Solution: EVA – HDFC Bank's AI Chatbot**
-

Case Study: Automating Customer Support with NLP Chatbots

- **Key NLP Features Used**
 - **Intent recognition:** Classifies queries into categories (e.g., account balance, card issues)
 - **Named Entity Recognition (NER):** Identifies dates, transaction amounts, account types
 - **Context management:** Maintains conversational state for multi-turn dialogues
 - **Multilingual support:** Handles queries in English and Hindi
- **Outcomes**
 - Handled over **2.7 million queries** in the first year
 - Resolved **85% of queries** without human intervention
 - **Reduced average query response time** to less than 0.4 seconds
 - Improved **customer satisfaction** and reduced cost-per-query
- **Challenges Faced**
 - Training the model on diverse and noisy real-world customer data
 - Maintaining user trust (transparency and security)
 - Escalating complex queries to human agents without losing context



Knowledge Graph Applications

Knowledge Graph

• Knowledge Graph Applications

Applications

- Semantic Search
- Question Answering
- Analytics
- Dashboards
- Knowledge Sharing
- Knowledge Management

Algorithms

- Inferencing
- Machine Learning
- Entity Recognition
- Disambiguation
- Text Understanding
- Recommendations

Knowledge Graph



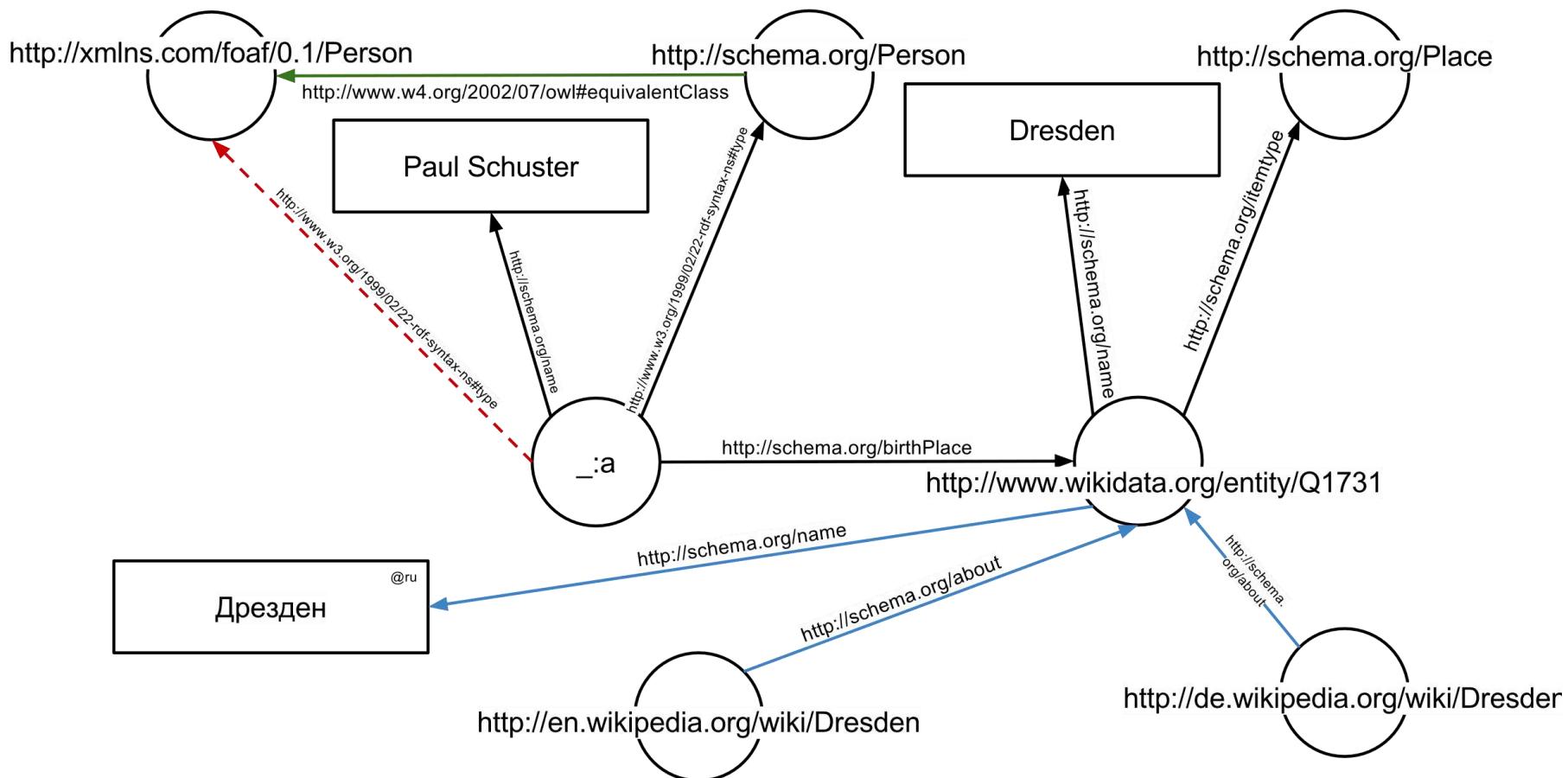
- Entities
- Relationships
- Semantic Descriptions

Data Sources

Data Transformation, Integration
Natural Language Processing



The Semantic Web



Case study – Knowledge Graph Applications in Agentic AI



Problem: Large tech company, faced challenges with internal information access. Key data about projects, teams, expertise, and documentation was scattered across various internal wikis, project management tools (like Jira), employee directories, and shared drives.

Employees spent significant time searching for answers to complex questions like:

- "Who worked on the authentication module for Project X and what were the key technical documents?"
- "Which active projects are using Python and are related to our client Y?"
- "Find the design specification document for the UI redesign led by A's team."

Enterprise Knowledge Graph (EKG) Construction



Data Sources: Internal Wiki, Jira, HR Database, Code Repositories (metadata), Document Management System.

Entities: Employee, Project, Team, Document, Skill, Client, CodeModule.

Relationships: WORKS_ON (Employee -> Project), MEMBER_OF (Employee -> Team), AUTHORED (Employee -> Document), RELATED_TO (Project -> Document), HAS_SKILL (Employee -> Skill), SERVES_CLIENT (Project -> Client), IMPLEMENTS (CodeModule -> Project).

Population: NLP techniques (NER, Relation Extraction) were used to extract entities and relationships from unstructured text (wikis, documents). Structured data was mapped directly (HR DB, Jira)

Agentic AI System ("Ask Me")

Core: An LLM (like GPT-4 or a fine-tuned open-source model)

•Tools:

- query_EKG: Takes a natural language query part and translates it into a structured query (e.g., Cypher for Neo4j) to fetch data from the EKG.
- document_search: Performs keyword/semantic search over the document system for full-text retrieval when needed.

•Capabilities: Planning, Tool Use, Reasoning, Synthesis.

Example of Enterprise KG application

User Query: "Who led the backend development for Project X and what recent documents did they author?"

Agent's Plan:

1. Identify entities: "Project X" (Project), "backend development" (implicit Skill/Role), "documents" (Document).
2. Find the team/person associated with "Project X" and "backend development lead" role via EKG. -> Use query_EKG.
3. Once the person is identified (e.g., "A"), find recent documents AUTHORED by "A" via EKG. -> Use query_EKG.
4. Synthesize the results into a natural language answer.

Execution:

- Agent calls query_EKG ("Find lead backend developer for Project X"). EKG returns "John Smith".
- Agent calls query_EKG ("Find recent documents authored by John Smith"). EKG returns a list: ["API_Design_v2.pdf", "Deployment_Strategy.docx"].

Agent's Response: "A led the backend development for Project X. Recent documents he authored include 'API_Design_v2.pdf' and Deployment_Strategy.docx".



Machine Translation

Machine Translation

English to Hindi <http://anglahindi.iitk.ac.in>

Simple Sentences.

sarala vaakya .

Welcome to London.

landana men aapaka svaagata hai.

There are some cases which are still pending.

**NLP applications is an interesting course in MTech AIML
of BITS WILP**

एनएलपी एप्लिकेशन बिट्स WILP के एमटेक एआईएमएल में
एक दिलचस्प कोर्स है

Machine Translation - Challenges

Accent Restoration in Spanish & French

Problem:

Input: ... déjà travaille cote a cote ...



Output: ... déjà travaillé côte à côté ...

Examples:

... appeler l'autre **cote** de l'atlantique ...

⇒ *côté* (meaning side) or

⇒ *côte* (meaning coast)

... une famille des **pecheurs** ...

⇒ *pêcheurs* (meaning fishermen) or

⇒ *pécheurs* (meaning sinners)

Machine Translation

Accent Restoration in Spanish & French

Training Data:

Pattern	Context
(1) côté	... du laisser de <i>cote</i> faute de temps ...
” ”	... appeler l' autre <i>cote</i> de l' atlantique ...
” ”	... passe de notre <i>cote</i> de la frontiere ...
(2) côte	... vivre sur notre <i>cote</i> ouest toujours ...
” ”	... creer sur la <i>cote</i> du labrador des ...
” ”	travaillaient cote a <i>cote</i> , ils avaient ...

Test Data:

Pattern	Context
???	... passe de notre <i>cote</i> de la frontiere ...
???	... creer sur la <i>cote</i> du labrador des ...

Machine Translation - Challenges

Capitalization

Problem:

... FRIED CHICKEN, TURKEY SANDWICHES AND FROZEN ...

- ⇒ *turkey* (the *bird*) or
- ⇒ *Turkey* (the *country*)

Training Data:

Capitalization	Context
(1) turkey	... OF FRIED CHICKEN , TURKEY SANDWICHES AND FROZEN ...
” ”	... NTS A POUND , WHILE TURKEY PRICES ROSE 1.2 CENTS ...
” ”	... PLAY , REAL GRADE-A TURKEY , WHICH ONLY A PRICE ...
(2) Turkey	... INUNDATED EASTERN TURKEY AFTER THE EARLIER ...
” ”	... FEELINGS TOWARD TURKEY SURFACED WHEN GREECE ...
” ”	... THE CONTRACT WITH TURKEY WILL PROVIDE OPPORTU...

Test Data:

Capitalization	Context
???	... NECK LIKE THAT OF A TURKEY ON A CHOPPING BLOCK ...
???	... PROBLEM IS THAT TURKEY IS NOT A EUROPEAN ...

Machine Translation - Challenges

Word sense disambiguation

Problem:

... He wrote the last **sentence** two years later ...

⇒ *sentencia* (legal sentence) or

⇒ *frase* (grammatical sentence)

Training Data:

Translation	Context
(1) sentencia	... for a maximum <i>sentence</i> for a young offender ...
” ”	... of the minimum <i>sentence</i> of seven years in jail ...
” ”	... were under the <i>sentence</i> of death at that time ...
(2) frase	... read the second <i>sentence</i> because it is just as ...
” ”	... The next <i>sentence</i> is a very important ...
” ”	... It is the second <i>sentence</i> which I think is at ...

Test Data:

Translation	Context
???	... cannot criticize a <i>sentence</i> handed down by ...
???	... listen to this <i>sentence</i> uttered by a former ...

Machine Translation - Challenges

Text to speech

Problem:

... slightly elevated *lead* levels ...
 ⇒ *lɛd* (as in *lead mine*) or
 ⇒ *li:d* (as in *lead role*)

Training Data:

Pronunciation	Context
(1) lɛd	... it monitors the <i>lead</i> levels in drinking ...
	... conference on <i>lead</i> poisoning in ...
	... strontium and <i>lead</i> isotope zonation ...
(2) li:d	... maintained their <i>lead</i> Thursday over ...
	... to Boston and <i>lead</i> singer for Purple ...
	... Bush a 17-point <i>lead</i> in Texas , only 3 ...

Test Data:

Pronunciation	Context
???	... median blood <i>lead</i> concentration was ..
???	... his double-digit <i>lead</i> nationwide . The ...

Case Study: Scaling Global Communications

- **Custom Engines:** The system was trained on specific content to create custom models that understood its unique terminology and branding.
 - **Centralized Platform:** All translation requests were funneled through the MT platform, which was integrated into existing content systems.
 - **MT + Post-Editing:** The platform was used to instantly translate content. For high-visibility materials, human translators would then "post-edit" the AI's output, rather than translating from scratch.
-

Information Extraction

Information Extraction



As a task:

Filling slots in a database from sub-segments of text.

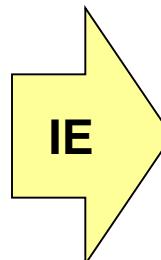
October 14, 2002, 4:00 a.m. PT

For years, [Microsoft Corporation](#) [CEO Bill Gates](#) railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said [Bill Veghte](#), a [Microsoft VP](#). "That's a super-important shift for us in terms of code access."

[Richard Stallman](#), [founder](#) of the [Free Software Foundation](#), countered saying...



NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

Phrase Types to Identify for IE



Closed set

U.S. states

He was born in Alabama...

The big Wyoming sky...

Regular set

U.S. phone numbers

Phone: (413) 545-1323

The CALD main office can be reached at 412-268-1299

Complex pattern

U.S. postal addresses

University of Arkansas
P.O. Box 140
Hope, AR 71802

Headquarters:
1128 Main Street, 4th Floor
Cincinnati, Ohio 45210

Ambiguous patterns, needing context and many sources of evidence

Person names

...was among the six houses sold by Hope Feldman that year.

Pawel Opalinski, Software Engineer at WhizBang Labs.

Deeper Information Extraction

1. Coreference resolution (within a document)
2. Entity linking (across documents)
3. Event extraction and linking
4. Knowledge base population (KBP)

Named Entity Recognition

CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by \$6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit AMR, immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL, said the increase took effect Thursday night and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Atlanta and Denver to San Francisco, Los Angeles and New York.

NE Types



Type	Tag	Sample Categories
People	PER	Individuals, fictional characters, small groups
Organization	ORG	Companies, agencies, political parties, religious groups, sports teams
Location	LOC	Physical extents, mountains, lakes, seas
Geo-Political Entity	GPE	Countries, states, provinces, counties
Facility	FAC	Bridges, buildings, airports
Vehicles	VEH	Planes, trains, and automobiles

Type	Example
People	<i>Turing</i> is often considered to be the father of modern computer science.
Organization	The <i>IPCC</i> said it is likely that future tropical cyclones will become more intense.
Location	The <i>Mt. Sanitas</i> loop hike begins at the base of <i>Sunshine Canyon</i> .
Geo-Political Entity	<i>Palo Alto</i> is looking at raising the fees for parking in the University Avenue district.
Facility	Drivers were advised to consider either the <i>Tappan Zee Bridge</i> or the <i>Lincoln Tunnel</i> .
Vehicles	The updated <i>Mini Cooper</i> retains its charm and agility.

Identifying phrases (chunking)

- Phrases that are useful for information extraction:
 - Named entities
 - As on previous slides
 - Relationship phrases
 - “said”, “according to”, ...
 - “was born in”, “hails from”, ...
 - “bought”, “hopes to acquire”, “formed a joint agreement with”, ...
 - Simple syntactic chunks (e.g., non-recursive NPs)
 - “Syntactic chunking” sometimes done before (or instead of) parsing
 - Also, “segmentation”: divide Chinese text into words (no spaces)
- So, how do we learn to mark phrases?
 - Earlier, we built an FST to mark dates by inserting brackets
 - But, it's common to set this up as a tagging problem ...

Reduce to a tagging problem ...

- The IOB encoding (Ramshaw & Marcus 1995):
 - B_X = “beginning” (first word of an X)
 - I_X = “inside” (non-first word of an X)
 - O = “outside” (not in any phrase)
 - Does not allow overlapping or recursive phrases

... **United Airlines** said Friday it has increased ...

B_ORG I_ORG O O O O
... the move , spokesman **Tim Wagner** said ...
O O O O B_PER I_PER O



What if this were tagged as B_ORG instead?

Example applications for IE

- Classified ads
- Restaurant reviews
- Bibliographic citations
- Appointment emails
- Legal opinions
- Papers describing clinical medical studies
- ...
- Adding facts to the semantic web

Sentiment Analysis

Sentiment classification



What features of the text could help predict # of stars?
(e.g., using a log-linear model) How to identify more?
Are the features hard to compute? (syntax? sarcasm?)

★★★★★ **An extremely versatile machine!**, November 22, 2006

By [**Dr. Nickolas E. Jorgensen "njorgens3"**](#)

This review is from: Cuisinart DGB-600BC Grind & Brew, Brushed Chrome (Kitchen)

This coffee-maker does so much! It makes weak, watery coffee! It grinds beans if you want it to! It inexplicably floods the entire counter with half-brewed coffee when you aren't looking! Perhaps it could be used to irrigate crops... It is time-consuming to clean, but in fairness I should also point out that the stainless-steel thermal carafe is a durable item that has withstood being hurled onto the floor in rage several times. And if all these features weren't enough, it's pretty expensive too. If faced with the choice between having a car door repeatedly slamming into my genitalia and buying this coffee-maker, I'd unhesitatingly choose the Cuisinart! The coffee would be lousy, but at least I could still have children...

Other text categorization tasks



- Is it **spam**? (see [features](#))
- What **grade**, as an answer to this essay question?
- Is it **interesting to this user**?
 - News filtering; helpdesk routing
- Is it **interesting to this NLP program**?
 - Skill classification for a digital assistant!
 - If it's **Spanish**, translate it from Spanish
 - If it's **subjective**, run the sentiment classifier
 - If it's an **appointment**, run information extraction
- Where should it be **filed**?
 - Which mail folder? (work, friends, junk, urgent ...)
 - Yahoo! / Open Directory / digital libraries

Case Study: AI-powered sentiment analysis platform



Data Aggregation:

- The tool pulled all reviews into a single dashboard.

Aspect-Based Sentiment:

- The AI didn't just label a review "Positive" or "Negative."
- It identified specific topics (aspects) and the sentiment for each.

Real-time Dashboard:

- Operations managers could instantly filter feedback by property, region, date, and topic.
-

Evaluation of NLP Applications

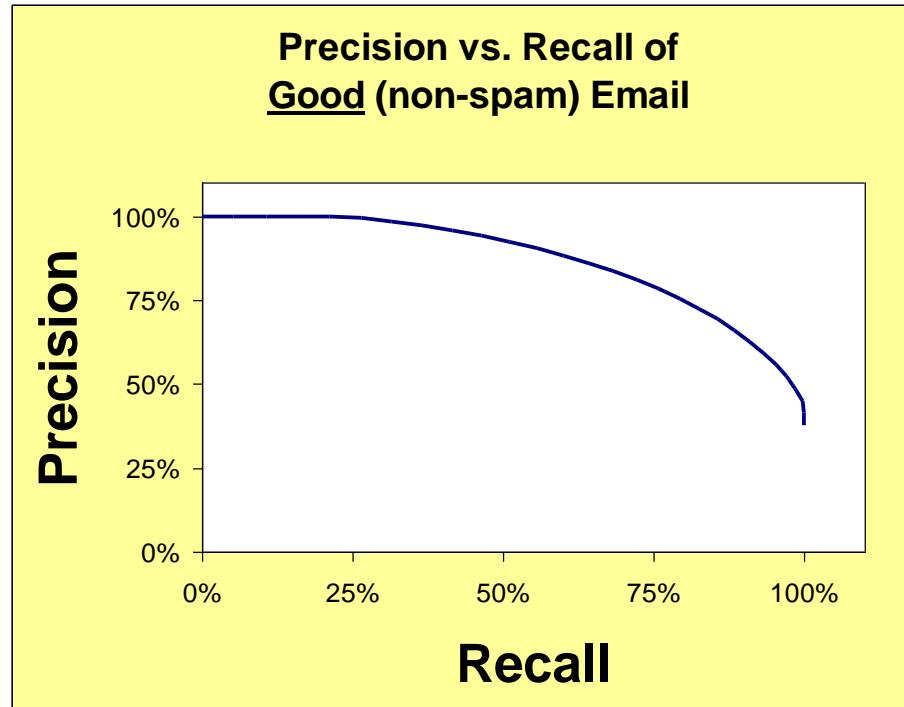
Measuring Performance

- Classification accuracy: What % of messages were classified correctly?
- Is this what we care about?

	Overall accuracy	Accuracy on spam	Accuracy on gen
System 1	95%	99.99%	90%
System 2	95%	90%	99.99%

- Which system do you prefer?

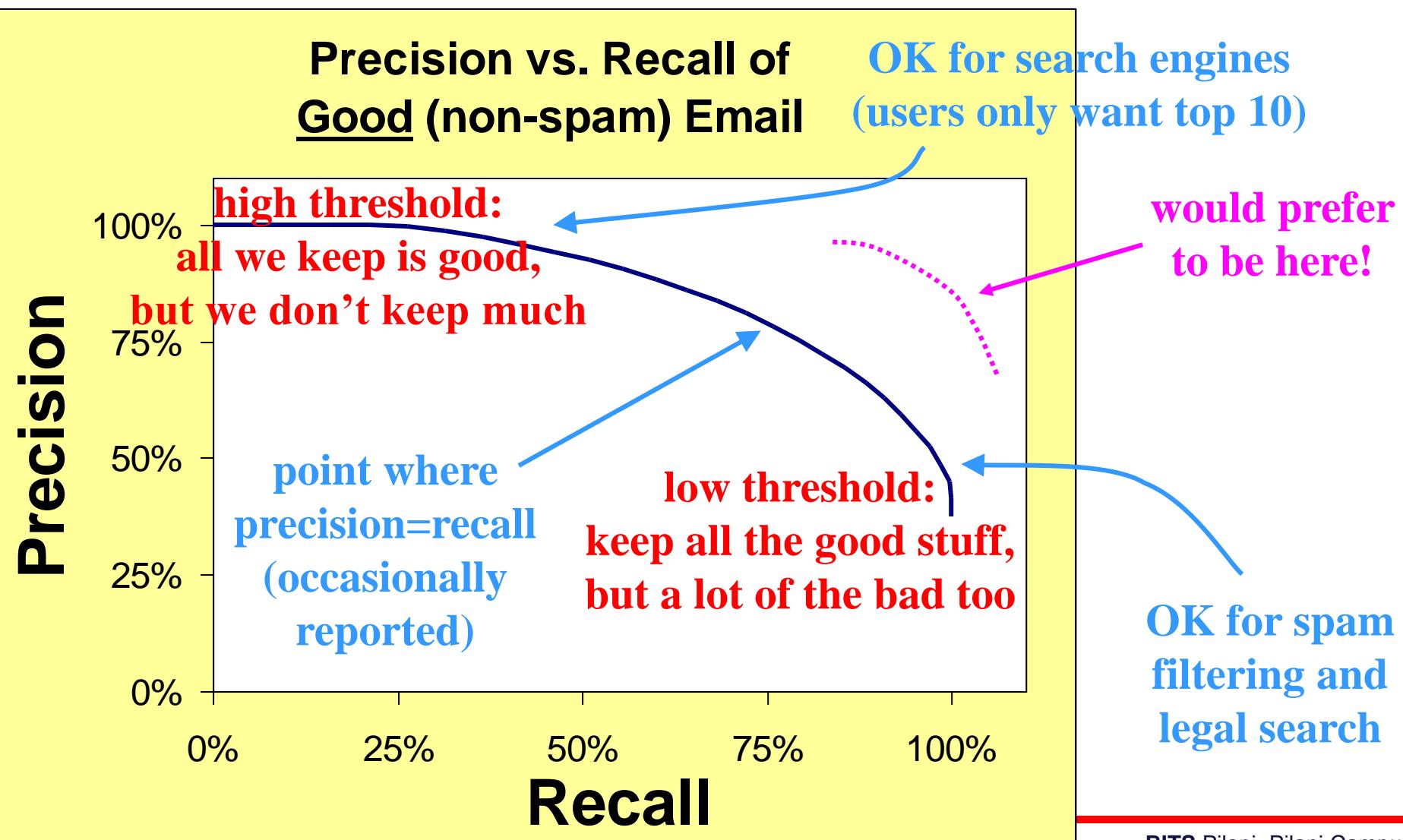
Measuring Performance



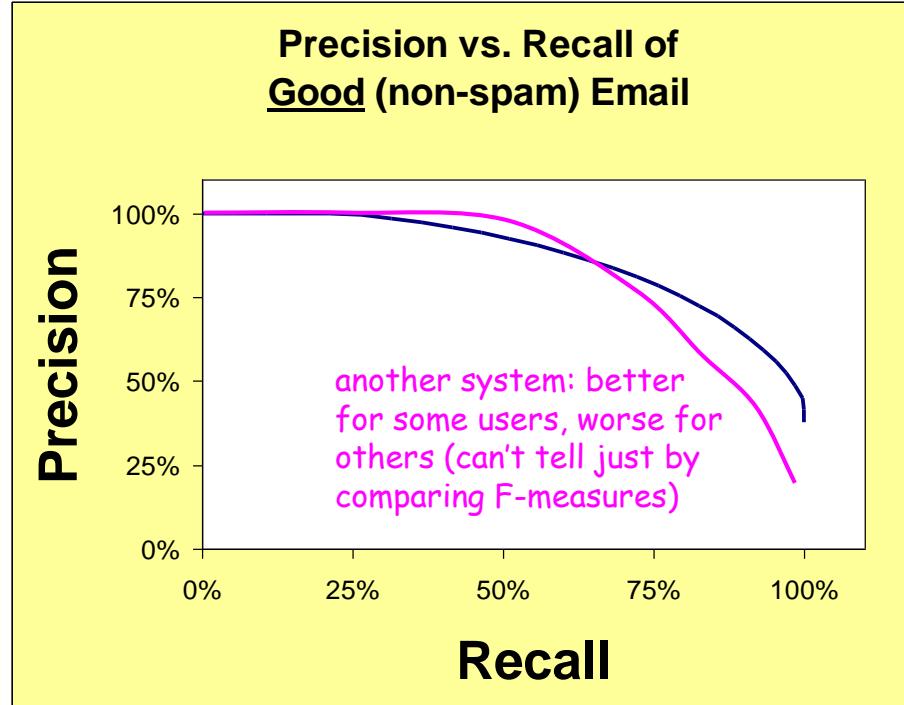
- **Precision** =
$$\frac{\text{good messages kept}}{\text{all messages kept}}$$
- **Recall** =
$$\frac{\text{good messages kept}}{\text{all good messages}}$$

Move from high precision to high recall by
deleting fewer messages (delete only if spamminess > high threshold)

Measuring Performance



Measuring Performance



- **Precision** = $\frac{\text{good messages kept}}{\text{all messages kept}}$
- **Recall** = $\frac{\text{good messages kept}}{\text{all good messages}}$
- **F-measure** =
$$\left(\frac{\text{precision}^{-1} + \text{recall}^{-1}}{2} \right)^{-1}$$

- Move from high precision to high recall by deleting fewer messages (raise threshold)
- Conventional to tune system and threshold to optimize F-measure on dev data
- But it's more informative to report the whole curve
 - Since in real life, the user should be able to pick a tradeoff point they like

More than 2 classes

- Report F-measure for each class
- Show a confusion matrix

		Predicted class			
		Lion	Camel	Deer	
True class	Lion				
	Camel	I			
	Deer	I	I		

56

correct

Generating new text

1. Speech recognition (transcribe as text)
 2. Machine translation
 3. Text generation from semantics
 4. Inflect, analyze, pronounce, or transliterate words
 5. Single- or multi-doc summarization
-

Multimodal interfaces or modeling

1. Sign languages
2. Speech + gestures
3. Images + captions
4. Brain recordings, human reaction times

Some Big Questions

- Neural nets are fluent at generating text, but do they really represent and reason about the world the text describes? Are their answers consistent? Can they explain them?
- How can models learn effectively through interaction with the world or with human teachers?
- What kinds of linguistic biases should we build in, and how? Huge Transformer LMs with enormous training sets work well, but can we find architectures that generalize like humans from much smaller datasets? (Or is that just pretraining + few-shot or fine-tuning?)

The NLP Research Community



- **Papers**
 - [ACL Anthology](#) has nearly everything, free!
 - Over 80,000 papers!
 - Free-text searchable
 - Great way to learn about current research on a topic
 - New search interfaces currently available in beta
 - » Find recent or highly cited work; follow citations
 - Used as a dataset by various projects
 - Analyzing the text of the papers (e.g., parsing it)
 - Extracting a graph of papers, authors, and institutions
(Who wrote what? Who works where? What cites what?)
 - [Google Scholar](#) to sort by citation count / track citations
 - [arXiv](#) papers

The NLP Research Community

- Institutions
 - Universities:
 - Several “big players” with many faculty
 - Companies:
 - Old days: AT&T Bell Labs, IBM
 - Now: Microsoft Research, Google Brain/DeepMind, FAIR, Amazon, startups ...
 - Nonprofits: AI2, HuggingFace, TTIC, ...
 - Many niche markets – online reviews, medical transcription, news summarization, legal search and discovery ...

The NLP Research Community



- **Software**

- Lots of people distribute code for these tasks
 - Search github – fun to download packages and play around!
 - Or you can email a paper's authors to ask for their code
 - [PapersWithCode.com](https://www.PapersWithCode.com)
 - Search for “awesome NLP” for some lists
- Toolkits and end-to-end pipelines for text analysis
 - [Hugging Face](https://huggingface.co) – > 1,143,998 models, > 248,002 datasets
 - Large pretrained models: pip install [transformers](#) ([quick tour](#))
 - Task-specific models: pip install [allennlp](#), etc.
 - [Allen NLP](#) (Python), [Spacy](#) (Cython), [UDPipe](#) (C++),
[Stanza](#) (Python), [CoreNLP](#) (Java), [NLTK](#) (Python)



Dr. Chetana is an Professor and Program lead in the CSIS department at Work Integrated Learning Programmes Division, BITS Pilani.

She has more than 27 years of teaching and industry experience. She did her PhD in Computer Science and Engineering from a joint programme of IIT Bombay and Monash University, Australia. She has been working extensively on different state of art research projects and has been awarded the “Best Industry Aligned Research” at the CSI TechNext India 2019 - Awards to Academia. She has published various papers and is also a reviewer at national and international level peer reviewed conferences and journals. Her areas of expertise include Machine Learning, Natural Language Processing, Semantic Web, Deep Learning, Text Mining, Big Data Analytics, Information Retrieval and Software Engineering.

Thank you!!

Good References

<https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html>

<https://web.stanford.edu/~jurafsky/sl3/>

<https://www.geeksforgeeks.org/top-7-applications-of-natural-language-processing/>

<https://emerj.com/partner-content/nlp-current-applications-and-future-possibilities/>

<https://www.nltk.org/book/>

<https://www.coursera.org/learn/python-text-mining/home/week/1>

<https://openai.com/api/>

<https://analyticssteps.com/blogs/top-nlp-tools>

<https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html>

https://www.cstr.ed.ac.uk/emasters/course/natural_lang.html

<https://web.stanford.edu/class/cs224u/2016/materials/cs224u-2016-intro.pdf>

<https://www.mygreatlearning.com/blog/trending-natural-language-processing-applications/>

<https://grammarchecker.io/page/plagiarism-checker>

<https://artificialanalysis.ai/leaderboards/models>

References

- Semantic Web stack of standards, W3C®
 - Linking Open Data cloud diagram by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net>
 - <https://www.geeksforgeeks.org/top-7-applications-of-natural-language-processing/>
 - <https://levity.ai/blog/11-nlp-real-life-examples>
 - <https://www.youtube.com/watch?v=otH29Uoo-HE>
 - <https://www.youtube.com/watch?v=UVfwBqcnbM>
 - <https://www.youtube.com/watch?v=BELFUhssjXE>
 - Overview of NLP Tasks and Featurization Frank Ferraro – ferraro@umbc.edu
 - Generative AI for everyone- coursera
 - Generative AI with LLM- courser
 - <https://www.assemblyai.com/blog/the-top-free-speech-to-text-apis-and-open-source-engines/>
 - <https://www.naturalreaders.com/online/>
 - <https://www.techtarget.com/whatis/feature/12-of-the-best-large-language-models>
 - <https://falconllm.tii.ae/>
-



BITS Pilani
Pilani Campus

Natural Language Processing Applications

Dr. Chetana Gavankar, Ph.D,

IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



Session 2:

These slides are prepared by the instructor, with grateful acknowledgement of Prof. Dan Jurafsky and many others who made their course materials freely available online.

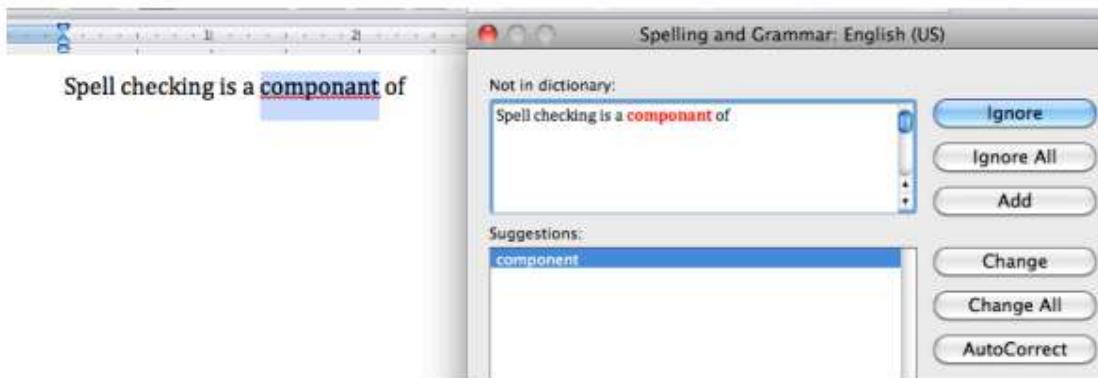
Session Content

- Spelling Correction and the Noisy Channel
 - Applications
 - Real World Spelling errors
 - Noisy channel
-

Spell check

Applications for spelling correction

Word processing



Phones



Web search

ploogle

natural langage processing

Showing results for [natural language](#) processing
Search instead for [natural langage](#) processing

Spelling Error Detection

- Spelling Error Correction:
- Autocorrect
 - the->the
- Suggest a correction
- Suggestion lists

Types of spelling errors

- **Non-word Errors**
 - graffe -> giraffe
 - acress
- **Real-word Errors**
 - Typographical errors
 - three -> there
- **Cognitive Errors (homophones)**
 - piece -> peace
 - too -> two
- **Short forms/Slang:** In this case may be u r just being kewl.

Rates of spelling errors

- 26%: Web Queries / LLM prompts
- 13%: Retyping, no backspace
- 7%: Words corrected retyping on phone-sized organizer
- 2%: Words uncorrected on organizer

Non-word spelling error

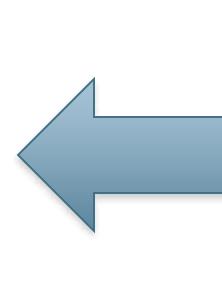
- Non-word spelling error detection:
 - Any word not in a dictionary is an error
 - The larger the dictionary the better
- Non-word spelling error correction:
 - Generate candidates: real words that are similar to error
 - Choose the one which is best:
 - Shortest weighted edit distance
 - Highest noisy channel probability

Real word spelling errors

- For each word w , generate candidate set:
 - Find candidate words with similar pronunciations
 - Find candidate words with similar spelling
 - Include w in candidate set
- Choose best candidate
 - Noisy Channel
 - Classifier

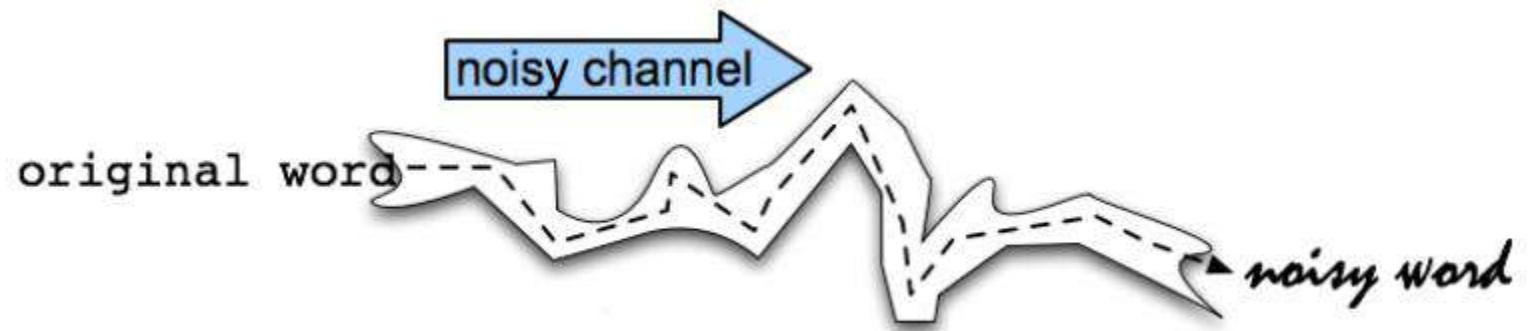
Terminology

- These are character bigrams:
 - *st, pr, an ...*
- These are word bigrams:
 - *palo alto, flying from, road repairs*
- In today's class, we will generally deal with *word bigrams*
- In the accompanying Coursera lecture, we mostly deal with *character bigrams* (because we cover stuff complementary to what we're discussing here)



Similarly trigrams,
k-grams etc

Noisy Channel Intuition



Noisy Channel

We see an observation x of a misspelled word
Find the correct word w

$$\begin{aligned}\hat{w} &= \operatorname{argmax}_{w \in V} P(w | x) \\ &= \operatorname{argmax}_{w \in V} \frac{P(x | w)P(w)}{P(x)} \\ &= \operatorname{argmax}_{w \in V} P(x | w)P(w)\end{aligned}$$

Non-word spelling error example

acress

Candidate generation

- Words with similar spelling
 - Small edit distance to error
- Words with similar pronunciation
 - Small edit distance of pronunciation to error

Damerau-Levenshtein edit distance

- Minimal edit distance between two strings, where edits are:
 - Insertion
 - Deletion
 - Substitution
 - Transposition of two adjacent letters

Words within 1 of acress

Error	Candidate Correction	Correct Letter	Error Letter	Type
acress	actress	t	-	deletion
acress	cress	-	a	insertion
acress	caress	ca	ac	transposition
acress	access	c	r	substitution
acress	across	o	e	substitution
acress	acres	-	s	insertion
acress	acres	-	s	insertion

Candidate generation

- 80% of errors are within edit distance 1
- Almost all errors within edit distance 2
- Also allow insertion of space or hyphen
 - thisidea -> this idea
 - inlaw -> in-law

How do you generate the candidates?

1. Run through dictionary, check edit distance with each word
2. Generate all words within edit distance $\leq k$ (e.g., $k = 1$ or 2) and then intersect them with dictionary
3. Use a character k -gram index and find dictionary words that share “most” k -grams with word (e.g., by Jaccard coefficient)
4. Compute them fast with a Levenshtein finite state transducer
5. Have a precomputed map of words to possible corrections

A paradigm ...

- We want the best spell corrections
- Instead of finding the very best, we
 - Find a subset of pretty good corrections
 - (say, edit distance at most 2)
 - Find the best amongst them
- *These may not be the actual best*
- This is a recurring paradigm in IR including finding the best docs for a query, best answers, best ads ...
 - Find a good candidate set
 - Find the top K amongst them and return them as the best

Let's say we've generated candidates: Now back to Bayes' Rule

- We see an observation x of a misspelled word
- Find the correct word \hat{w}

$$\hat{w} = \operatorname{argmax}_{w \in V} P(w | x)$$

$$= \operatorname{argmax}_{w \in V} \frac{P(x | w)P(w)}{P(x)}$$

$$= \operatorname{argmax}_{w \in V} P(x | w)P(w)$$

What's $P(w)$?

Language Model

- Take a big supply of words (your document collection with T tokens); let $C(w)$ = # occurrences of w

$$P(w) = \frac{C(w)}{T}$$

Language Model

- Use any of the language modeling algorithms we've learned
 - Unigram, bigram, trigram
 - Web-scale spelling correction
 - Stupid backoff

Unigram Prior probability

- Counts from 404,253,213 words in Corpus of Contemporary English (COCA)

word	Frequency of word	P(word)
actress	9,321	.0000230573
cress	220	.0000005442
caress	686	.0000016969
access	37,038	.0000916207
across	120,844	.0002989314
acres	12,874	.0000318463

Channel model probability

- Misspelled word $x = x_1, x_2, x_3 \dots x_m$
- Correct word $w = w_1, w_2, w_3, \dots, w_n$
- $P(x|w)$ = probability of the edit
 - (deletion/insertion/substitution/transposition)

Computing error probability: confusion matrix

```
del[x,y]:    count(xy typed as x)
ins[x,y]:    count(x typed as xy)
sub[x,y]:    count(x typed as y)
trans[x,y]:  count(xy typed as yx)
```

Insertion and deletion conditioned on previous character

Confusion matrix for spelling errors

X	sub[X, Y] = Substitution of X (incorrect) for Y (correct)																									
	Y (correct)																									
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	0	8	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

Nearby keys



Generating the confusion matrix

- [Peter Norvig's list of errors](#)
- [Peter Norvig's list of counts of single-edit errors](#)
 - All Peter Norvig's ngrams data links: <http://norvig.com/ngrams/>

Channel Model

$$P(x|w) = \begin{cases} \frac{\text{del}[w_{i-1}, w_i]}{\text{count}[w_{i-1} w_i]}, & \text{if deletion} \\ \frac{\text{ins}[w_{i-1}, x_i]}{\text{count}[w_{i-1}]}, & \text{if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, & \text{if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, & \text{if transposition} \end{cases}$$

Smoothing probabilities: Add-1 smoothing

- But if we use the confusion matrix example, unseen errors are impossible!
- They'll make the overall probability 0. That seems too harsh
 - e.g., in Kernighan's chart $q \rightarrow a$ and $a \rightarrow q$ are both 0, even though they're adjacent on the keyboard!
- A simple solution is to add 1 to all counts and then if there is a $|A|$ character alphabet, to normalize appropriately:

$$\text{If substitution, } P(x | w) = \frac{\text{sub}[x, w] + 1}{\text{count}[w] + A}$$

Channel model for acres

Candidate Correction	Correct Letter	Error Letter	x w	P(x word)
actress	t	-	c ct	.000117
cress	-	a	a #	.00000144
caress	ca	ac	ac ca	.00000164
access	c	r	r c	.000000209
across	o	e	e o	.0000093
acres	-	s	es e	.0000321
acres	-	s	ss s	.0000342

Noisy channel probability for acres

Candidate Correction	Correct Letter	Error Letter	x w	P(x word)	P(word)	$10^9 * P(x w)P(w)$
actress	t	-	c ct	.000117	.0000231	2.7
cress	-	a	a #	.00000144	.000000544	.00078
caress	ca	ac	ac ca	.00000164	.00000170	.0028
access	c	r	r c	.000000209	.0000916	.019
across	o	e	e o	.0000093	.000299	2.8
acres	-	s	es e	.0000321	.0000318	1.0
acres	-	s	ss s	.0000342	.0000318	1.0

Noisy channel probability for acres

Candidate Correction	Correct Letter	Error Letter	x w	P(x word)	P(word)	$10^9 * P(x w)P(w)$
actress	t	-	c ct	.000117	.0000231	2.7
cress	-	a	a #	.00000144	.000000544	.00078
caress	ca	ac	ac ca	.00000164	.00000170	.0028
access	c	r	r c	.000000209	.0000916	.019
across	o	e	e o	.0000093	.000299	2.8
acres	-	s	es e	.0000321	.0000318	1.0
acres	-	s	ss s	.0000342	.0000318	1.0

Incorporating context words: Context-sensitive spelling correction

- Determining whether **actress** or **across** is appropriate will require looking at the context of use
- We can do this with a better **language model**
- A **bigram language model** conditions the probability of a word on (just) the previous word

$$P(w_1 \dots w_n) = P(w_1)P(w_2 | w_1) \dots P(w_n | w_{n-1})$$

Incorporating context words

- For unigram counts, $P(w)$ is always non-zero
 - if our dictionary is derived from the document collection
- This won't be true of $P(w_k | w_{k-1})$. We need to **smooth**
- We could use add-1 smoothing on this conditional distribution
- But here's a better way – interpolate a unigram and a bigram:

$$P_{\text{li}}(w_k | w_{k-1}) = \lambda P_{\text{uni}}(w_k) + (1-\lambda)P_{\text{bi}}(w_k | w_{k-1})$$

- $P_{\text{bi}}(w_k | w_{k-1}) = C(w_{k-1}, w_k) / C(w_{k-1})$

Using a bigram language model

- “*a stellar and versatile **acress** whose combination of sass and glamour...*”
- Counts from the Corpus of Contemporary American English with add-1 smoothing
 - $P(\text{actress}|\text{versatile}) = .000021$
 - $P(\text{across}|\text{versatile}) = .000021$
 - $P(\text{whose}|\text{actress}) = .0010$
 - $P(\text{whose}|\text{across}) = .000006$
- $P(\text{"versatile actress whose"}) = .000021 * .0010 = 210 \times 10^{-10}$
- $P(\text{"versatile across whose"}) = .000021 * .000006 = 1 \times 10^{-10}$

Using a bigram language model

- “**a stellar and versatile **acress** whose combination of sass and glamour...**”
- Counts from the Corpus of Contemporary American English with add-1 smoothing
 - $P(\text{actress}|\text{versatile}) = .000021$ $P(\text{whose}|\text{actress}) = .0010$
 - $P(\text{across}|\text{versatile}) = .000021$ $P(\text{whose}|\text{across}) = .000006$
- $P(\text{"versatile actress whose"}) = .000021 * .0010 = 210 \times 10^{-10}$
- $P(\text{"versatile across whose"}) = .000021 * .000006 = 1 \times 10^{-10}$

Evaluation

- Some spelling error test sets
 - [Wikipedia's list of common English misspelling](#)
 - [Aspell filtered version of that list](#)
 - [Birkbeck spelling error corpus](#)
 - [Peter Norvig's list of errors \(includes Wikipedia and Birkbeck, for training or testing\)](#)

Real word spelling errors

- ...leaving in about fifteen **minuets** to go to her house.
- The design **an** construction of the system...
- Can they **lave** him my messages?
- The study was conducted mainly **be** John Black.
- 25-40% of spelling errors are real words Kukich 1992

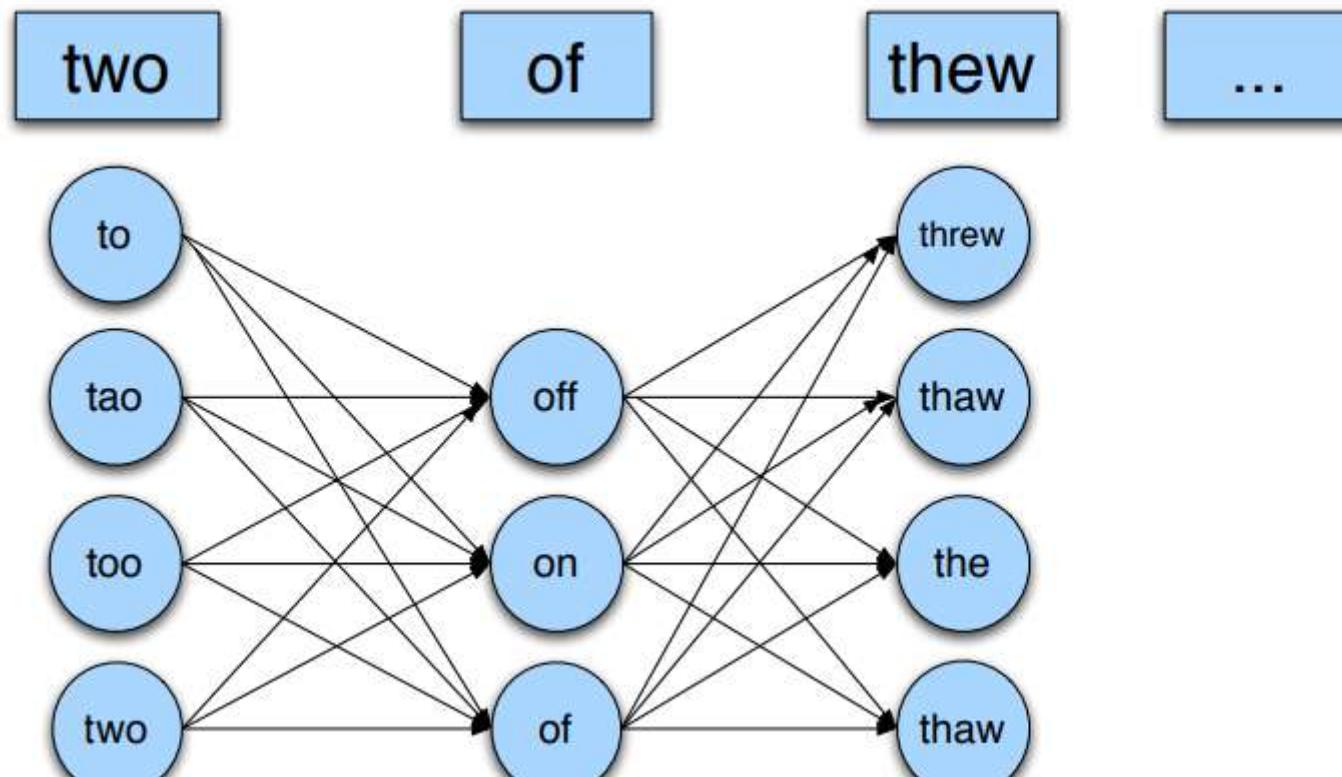
Solving real-word spelling errors

- For each word in sentence
 - Generate candidate set
 - the word itself
 - all single-letter edits that are English words
 - words that are homophones
 - Choose best candidates
 - Noisy channel model
 - Task-specific classifier

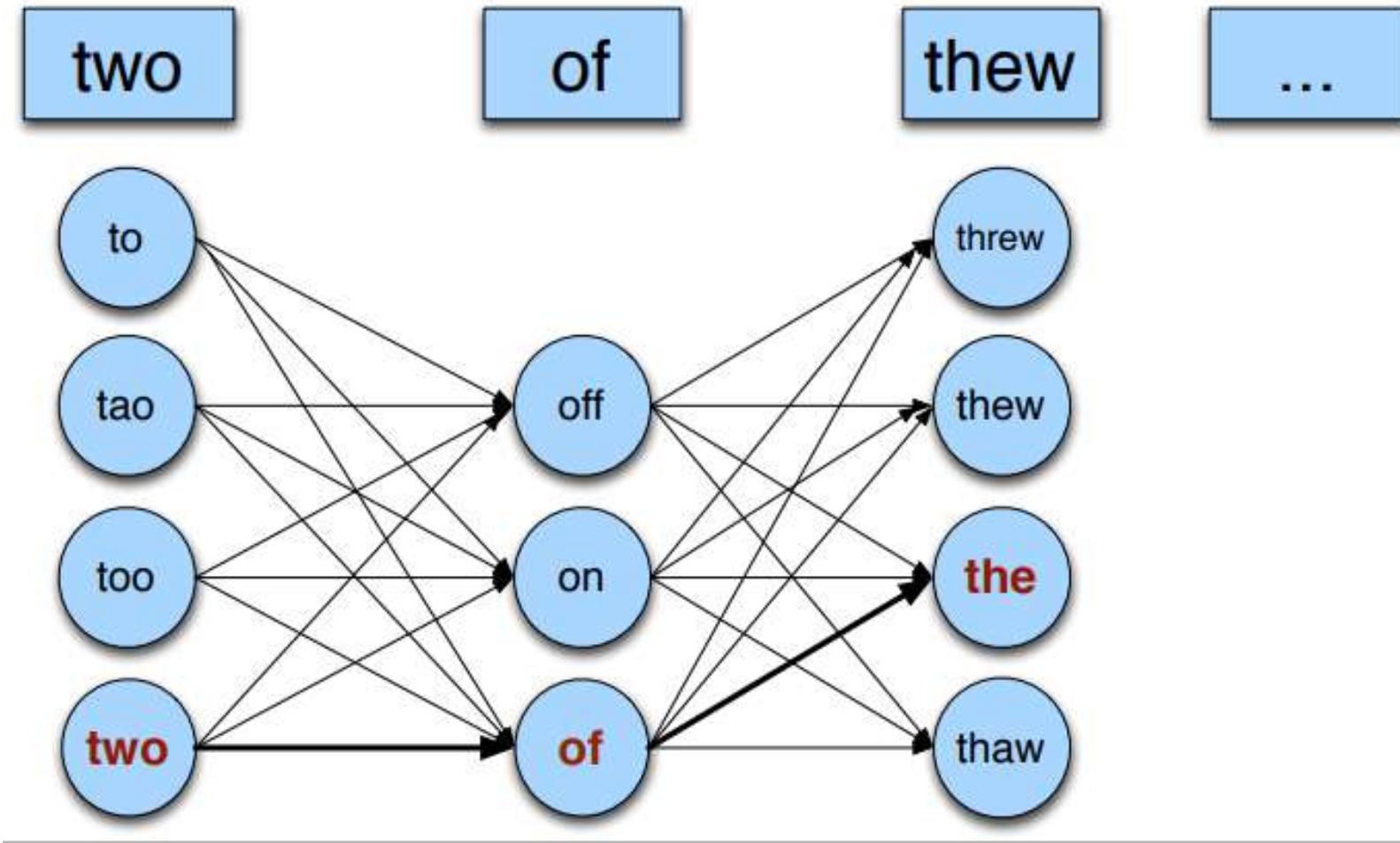
Noisy channel for real-word spell correction

- Given a sentence $w_1, w_2, w_3, \dots, w_n$
- Generate a set of candidates for each word w_i
 - Candidate(w_1) = $\{w_1, w'_1, w''_1, w'''_1, \dots\}$
 - Candidate(w_2) = $\{w_2, w'_2, w''_2, w'''_2, \dots\}$
 - Candidate(w_n) = $\{w_n, w'_n, w''_n, w'''_n, \dots\}$
- Choose the sequence W that maximizes $P(W)$

Noisy channel for real-word spell correction



Noisy channel for real-word spell correction



Simplification: One error per sentence

- Out of all possible sentences with one word replaced
 - w_1, w''_2, w_3, w_4 two off thew
 - w_1, w_2, w'_3, w_4 two of the
 - w'''_1, w_2, w_3, w_4 too of thew
 - ...
- Choose the sequence W that maximizes $P(W)$

Where to get the probabilities

- Language model
 - Unigram
 - Bigram Etc
- Channel model
 - Same as for non-word spelling correction
 - Plus need probability for no error, $P(w|w)$

Probability of no error

- What is the channel probability for a correctly typed word?
 - $P(\text{"the"} | \text{"the"})$
- Obviously this depends on the application
 - .90 (1 error in 10 words)
 - .95 (1 error in 20 words)
 - .99 (1 error in 100 words)
 - .995 (1 error in 200 words)

Peter Norvig's “thew” example

x	w	x w	P(x w)	P(w)	$10^9 P(x w)P(w)$
thew	the	ew e	0.000007	0.02	144
thew	thew		0.95	0.00000009	90
thew	thaw	e a	0.001	0.0000007	0.7
thew	threw	h hr	0.000008	0.000004	0.03
thew	thwe	ew we	0.000003	0.00000004	0.0001

HCI issues in spelling

- If very confident in correction
 - Autocorrect
- Less confident
 - Give the best correction
- Less confident
 - Give a correction list
- Unconfident
 - Just flag as an error

Improvements to channel model

- Allow richer edits (Brill and Moore 2000)
 - Ent -> ant
 - ph -> f
 - le -> al
- Incorporate pronunciation into channel (Toutanova and Moore 2002

Channel model

- Factors that could influence $p(\text{misspelling} \mid \text{word})$
 - The source letter
 - The target letter
 - Surrounding letters
 - The position in the word
 - Nearby keys on the keyboard
 - Homology on the keyboard
 - Pronunciations
 - Likely morpheme transformations

Open source Spellcheck tools based on Rule based and Statistical methods

- **pyspellchecker**: This is a pure Python library and a direct implementation of Peter Norvig's famous statistical spell-checking algorithm. It's simple, easy to use, and self-contained.
- **textblob**: This popular NLP library includes a statistical spellchecker (also based on Norvig's algorithm) as one of its many features.
- **Symspellpy**: is a high-speed Python library for spelling correction and fuzzy searching based on the Symmetric Delete algorithm
- **LanguageTool**: Powerful rule-based spellchecker.
 - combines these rules with statistical (n-gram) models to catch errors that simple dictionary lookups would miss.
- **Hunspell**: Famous and powerful open-source rule-based spellcheck engine.
It is the spellchecking engine used by Google Chrome, Mozilla Firefox, LibreOffice & OpenOffice, macOS (for many applications)

Open source spellcheck tools

Tool	Type	Primary Use	Context-Aware?
LanguageTool	Hybrid (Rules + ML)	Full Grammar & Style API	Yes (statistical)
Hunspell	Engine / Library	Fast, accurate spellchecking	No
SymSpell	Library	Extreme speed (e.g., search)	No
pyspellchecker	Library (Statistical)	Simple Python spellcheck	No

Open source Spellcheck online tools

- <https://languagetool.org/>
- <https://spellcheck24.net/>
- <https://www.reverso.net/text-translation>
- <https://www.scribens.com/>

Classifier-based methods for real-word spelling correction

- Instead of just channel model and language model
- Use many features in a classifier
- Build a classifier for a specific pair like: whether/weather
 - “cloudy” within +- 10 words
 - ____ to VERB
 - ____ or not

Rule-based methods

- Rule-based methods rely on predefined rules and dictionaries to identify and correct spelling errors.
- For example, a rule-based spell checker might use a list of common words, a set of spelling rules, and a set of exceptions to detect and suggest corrections.
- Rule-based methods are easy to implement, fast, and transparent.
- Microsoft Word uses a combination of rule-based and dictionary-based spell checkers to ensure accurate text correction
- However, they also have some limitations.
 - They cannot handle new words, slang, proper nouns, or variations in spelling.
 - They also require a lot of manual work to create and maintain the rules and dictionaries.
 - Moreover, they may not be able to handle complex or ambiguous errors, such as homophones, context-dependent errors, or errors that affect the meaning of the sentence.

Statistical methods

- Statistical methods use probabilistic models and machine learning to learn from large corpora of text and generate corrections based on the likelihood of occurrence.
- For example, a statistical spell checker might use a language model to estimate the probability of a word given its previous and next words, and a noisy channel model to estimate the probability of a word given its misspelled version.
- Statistical methods are more flexible, adaptive, and scalable than rule-based methods.
- They can handle new words, variations in spelling, and different domains and languages.
- However, they also have some drawbacks.
 - They require a lot of data and computational resources to train and run the models.
 - They may also produce inaccurate or inappropriate corrections, especially if the data is noisy, biased, or sparse.
 - Furthermore, they may not be able to explain why they made a certain correction or how confident they are about it.

DL/ Transformer based Spell checkers

- Neural spell checker might use an encoder-decoder architecture to encode the misspelled word and its context into a vector representation, and then decode it into a corrected word or sequence.
- They can handle complex and multiple errors, capture the meaning and structure of the text, and adapt to different domains and languages.
- However, they also have some challenges.
 - They require a lot of data and computational resources to train and run the networks.
 - They may also suffer from overfitting, underfitting, or hallucination, meaning that they may memorize, ignore, or invent words that are not in the data.
 - Moreover, they may not be able to explain why they made a certain correction or how confident they are about it.

DL/ Transformer based Spell checkers

- **Sequence-to-Sequence (Seq2Seq) Models:**
- Based on recurrent neural networks (RNNs) or transformer architectures,
- These models are trained on pairs of input and output sequences (e.g., correct sentences and their corresponding erroneous versions) and learn to predict corrections by generating output sequences given the input.
- **LSTM (Long Short-Term Memory) Networks:**
- type of recurrent neural network, is effective in learning long-range dependencies in sequential data, making it useful for spellchecking and grammar correction tasks.
- LSTMs can handle variable-length sequences and retain information over longer periods, which is beneficial in understanding context within sentences.

Bi-directional LSTM

- *The bidirectional Long Short Term Memory networks (BLSTM) that can be trained using all available input information in the past and future of a specific time frame.*
- *For example,* Let us take an example of missing word generation in the *I am ___ student.*
- Unidirectional LSTMs will use only ‘I am’ to generate next word and based on the example it has seen during training it will generate a new word (it may be ‘a’, ‘very’ etc.).
- But bidirectional LSTMs have information of the past (I am) and future (student), so it can easily see that here it has to be **a**.

DL/ Transformer based Spell checkers

- **Attention Mechanisms:**
- Commonly integrated into transformer-based models like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), or RoBERTa, enable the model to focus on specific parts of the input sequence while making corrections.
- This attention mechanism enhances the model's ability to capture relevant context for accurate spell and grammar checks.
- **BERT-Based Models:**
- BERT, a transformer-based language representation model, has been fine-tuned for various natural language understanding tasks, including grammar and spellcheck.
- Its bidirectional architecture allows it to understand context more effectively, aiding in better error detection and correction.

LLM used in spell check and grammar check

- **Language Models for Contextual Understanding:**
- Language models trained on large corpora of text data, such as GPT are adept at understanding context and can be fine-tuned for grammar and spellcheck tasks.
- Generate corrections based on the context provided in a given sentence.
- **Ensemble Models:**
- Ensemble methods leverage the strengths of various models to improve performance.

DL/ Transformer based Open Source Spell checker Tools

- Pre-trained Models
 - Hugging Face Transformers like T5 that has been fine-tuned on the task of "translating" incorrect text into correct text.
- LanguageTool
 - uses a **hybrid approach**, which is what makes it so robust.
 - **Rule-Based:** A massive set of hand-crafted rules for catching common grammar and spelling errors.
 - **Machine Learning:** Statistical (n-gram) models to find contextual errors (like "I went **two** the store") that the rules miss.
 - **DL/ Transformer based** (used in premium version)
- NeuSpell
 - Implementations of different DL models (BERT-based, LSTM-based, and Seq2Seq models) all pre-trained for spellchecking.

Combining Neural network and statistical methods

- **Language Models as Prior Knowledge:**
 - Statistical language models, such as n-grams or Hidden Markov Models (HMMs), can provide probabilities of word sequences or character sequences, which neural networks can leverage to refine their predictions or to provide additional context.
- **Error Probability Estimation:**
 - Statistical methods can help estimate the likelihood of a word being misspelled or a sequence of words containing grammatical errors.
 - These probabilities can serve as features or signals for the neural network model to consider during its learning process, enabling it to weigh the predictions accordingly.

Combining Neural network and statistical methods

- **Feature Engineering:**
 - Features derived from statistical analyses, such as character n-grams, syntactic features, or probabilities based on language rules, can be fed into the neural network as additional input features, enhancing its ability to detect and correct spelling or grammar errors.
- **Combining Outputs:**
 - ensemble the outputs of multiple neural network models.
 - For instance, by leveraging statistical techniques like voting or weighted averaging of predictions from different neural network architectures, the overall accuracy and robustness of the spellcheck system can be improved.

Combining Rule based, Statistical and Neural network methods

- **Rule-based Preprocessing:**
- Utilize rule-based systems to perform initial preprocessing steps such as basic spelling checks, identifying common grammatical errors, and applying language-specific rules.
- Create rules to detect and correct known errors or patterns that are easily identifiable, such as basic misspellings or common grammatical mistakes.

Combining Rule based, Statistical and Neural network methods

- **Statistical Methods for Error Detection:**
- Employ statistical models, such as n-grams, Hidden Markov Models (HMMs), or language models, to calculate probabilities of word sequences or character sequences.
- Use statistical analysis to identify anomalies or deviations in text that might indicate spelling or grammar errors.
- Estimate error probabilities for words or sequences and use these probabilities as features in the subsequent stages.

Combining Rule based, Statistical and Neural network methods

- **Neural Network Integration:**
- Train neural network architectures, such as LSTM networks, Transformers (e.g., BERT), or sequence-to-sequence models, to learn from labeled data containing correct and incorrect text pairs.
- Use neural networks to capture complex patterns, context, and semantic understanding in the text data.
- Incorporate statistical features or error probabilities derived from the statistical methods as additional input features for the neural network models.

Combining Rule based, Statistical and Neural network methods

- **Hybrid Model Ensemble:**
- Ensemble different models (rule-based, statistical, neural network) to combine their outputs for final error detection and correction.
- Develop a mechanism to combine predictions from each approach, either through voting mechanisms, weighted averaging, or stacking techniques.
- Apply a post-processing step that merges the strengths of each model's predictions to provide the most accurate and contextually appropriate corrections.

Recent Research Trends

- **Multilingual grammar correction**
- **Low-resource grammar correction** using synthetic error generation
- **Unsupervised and semi-supervised GEC**
- **Reinforcement Learning** for fluency-aware correction
- **Explainable GEC models** (interpretable edits)

References

- Speech and Language processing: An introduction to Natural Language Processing, Computational Linguistics and speech Recognition by Daniel Jurafsky and James H. Martin[3rd edition] – Appendix B

Dan Jurafsky Course :

- https://www.youtube.com/watch?v=dAf_vcs7DuE

Pawan Goyal Course:

- <https://www.youtube.com/watch?v=Hmc0lVdXVpE>
- <https://sanketp.medium.com/language-models-spellchecking-and-autocorrection-dd10f739443c>
- <https://www.youtube.com/watch?v=4htP-hmlBvc>
- <https://nlp.stanford.edu/IR-book/html/htmledition/spelling-correction-1.html>
- <https://www.geeksforgeeks.org/spelling-checker-in-python/>
- <https://www.slideshare.net/AmirShokri2/personalized-spell-checking-using-neural-networks>
- https://www.youtube.com/watch?v=2RU5egfeb_g

References

- <https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html>
- <https://web.stanford.edu/~jurafsky/slp3/>
- <https://www.youtube.com/watch?v=3a7QEkrDko>
- <https://www.youtube.com/watch?v=sQVpMzNXdco>
- <https://towardsdatascience.com/textblob-spelling-correction-46321fc7f8b8>
- DL models for Spellcheck
- <https://bhashkarkunal.medium.com/spelling-correction-using-deep-learning-how-bi-directional-lstm-with-attention-flow-works-in-366fabcc7a2f#:~:text=In%20Symmetric%20delete%20spelling%20correction,during%20a%20pre%20calculation%20step.>
- <https://medium.com/@agusnavce/a-quick-overview-of-the-implementation-of-a-fast-spelling-correction-algorithm-39a483a81ddc>



BITS Pilani
Pilani Campus

Natural Language Processing Applications

Dr. Chetana Gavankar, Ph.D,

IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



Session 3:

These slides are prepared by the instructor, with grateful acknowledgement of Prof. Dan Jurafsky and many others who made their course materials freely available online.

Session Content

-
- Grammatical Errors
 - Types of Grammatical Errors
 - Corpora and Tasks
 - Methodologies and Systems
 - Grammar Correction Approaches
 - Evaluation
 - Current and Future Trends
-

What is a grammatical error?

Syntax:

“Each language has its own systematic ways through which words and sentences are assembled to convey meaning.” Fraser & Hodson (1978)

Syntax errors are rule-driven (e.g. subj-verb agreement) thus easier to learn

Usage: Conventional usage habits

A wrong preposition or missing determiner

– do not break rules of syntax but of usage.

Usage errors are most common for learners

– greater reliance on memory than rules

Syntax

- **Syntax** (of natural languages) describe how words are strung together to form components of sentences, and how those components are strung together to form sentences.
- In the core of the description of the syntax of a natural language, we use context-free grammars (CFGs).
- Groups of words may behave as a single unit or phrases, called as **constituent**.
 - noun phrase,
 - verb phrase
- CFGs will allow us to model these constituency facts.

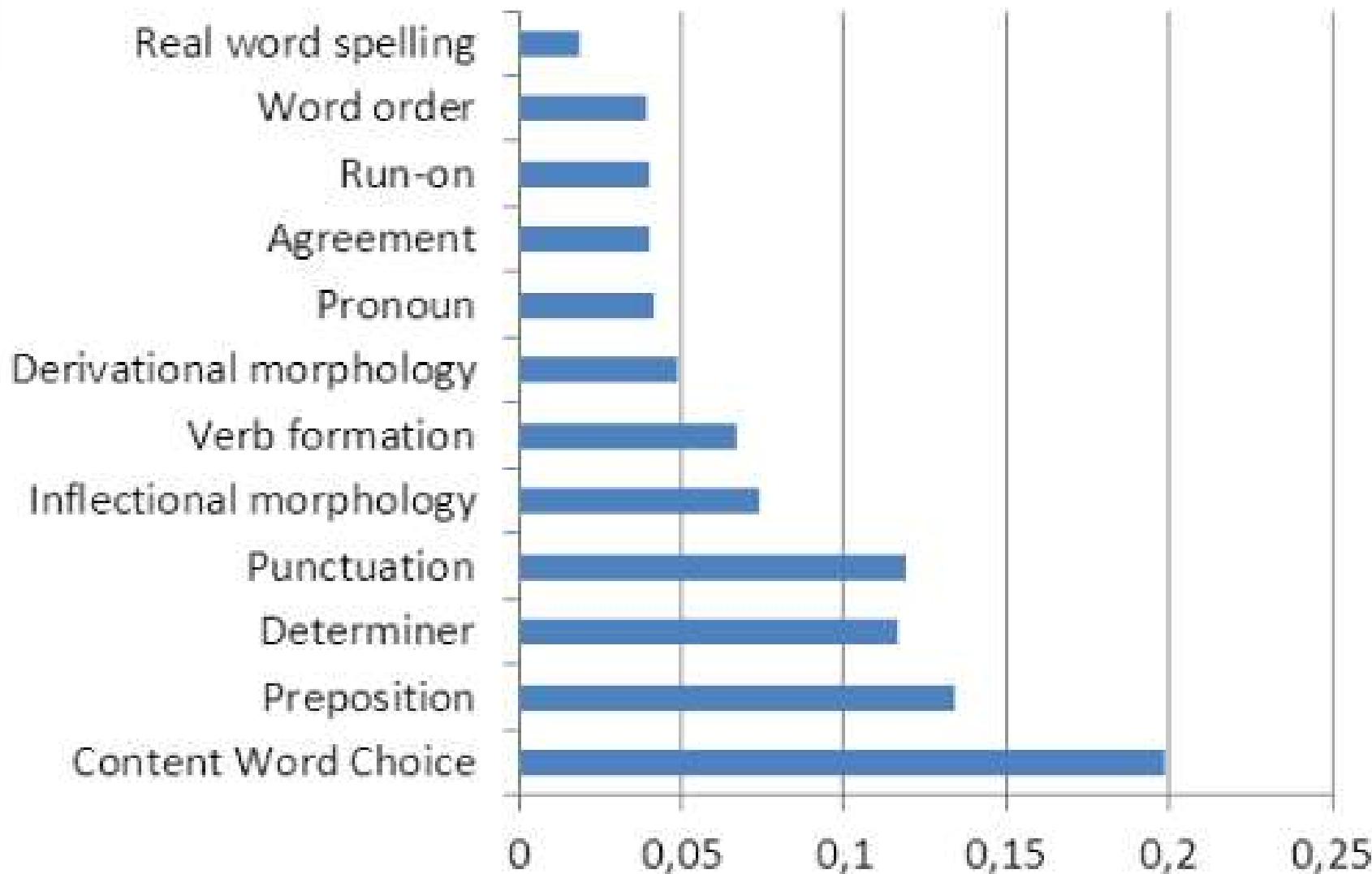
Applications of Grammar Check

- Question Answering
- Information Extraction
- Generation
- Translation
- Understanding

Focus on English only for practical reasons

- Over a billion people speak English as a second or foreign language worldwide
- By 2025, estimated that English language learners will make up 25% of the US public school population
- 725,000 international students at US universities
- 627 million people have taken the TOEFL
- English language has most resources

Learner errors



Prepositions Presence and Choice: 13%

- Prepositions are problematic because they perform so many complex roles
- Preposition choice in an adjunct is constrained by its object (“leave on Friday”, “leave at noon”)
- Prepositions are used to mark the arguments of a predicate (“fond of beer.”)
- Phrasal Verbs (“give in to their demands.”)
- “give in” -> “acquiesce”, “surrender”



Preposition choice

- Multiple prepositions can appear in the same context:

“When the plant is horizontal, the force of the gravity causes the sap to move __ the underside of the stem.”

Choices	Source
<ul style="list-style-type: none">• to• on• toward• onto	<ul style="list-style-type: none">• Writer• System• Annotator 1• Annotator 2

Determiner Presence and Choice: 12%

- English Article system: a, an, the
 - levels of countability from a car to *an equipment
 - Syntactic properties: have a knowledge vs a knowledge of English
 - Discourse factors – previous mention
 - Idioms: kick the/a bucket
 - World Knowledge
 - the moon (on earth)



Punctuation Conventions

- Apostrophe (1%):
 - Possessives
 - Contractions
- Comma (10%)
 - Missing after introductory clause
- Hyphenation (1%)
 - when used adjectively



Verbal Morphology and Tense: 14%

- Over-regularization of irregular verbs
 - The women *weared/wore long dresses.
- Ill-formed tense, participle, infinitive, modal & auxiliary
 - I look forward to *see/seeing you.
 - People would *said/say
 - It can *do/be harmful.
- Can be dependent on discourse
 - I will clean my room yesterday



Derivational Morphology: 5%

- Confusion of adjectival, nominal, verbal, adverbial forms
 - I have already made the *arranged/arrangements.
 - There was a wonderful women volleyball match between Chinese team and *Cuba/Cuban team.
 - I *admiration/admire my teacher.



Pronoun Error: 4%

- **Use of wrong case Wrong gender**
 - *Him/He went to the store.
- Wrong gender
 - I met Jane and he showed me where to go.
- Vague pronoun reference
 - I'll position the target, and when I nod my head, shoot at it.

Agreement Error: 4%

- These can be long distance Subject-verb agreement:
 - Three new texts which deal with this problem *has/have been written last year.
- Subject-verb agreement:
 - I *were/was in my house.
- Noun-number agreement
 - I am reading *these/this book.
 - Conversion always takes a lot of *efforts/effort.

Run-on Sentences: 4%

- Two independent clauses not connected by appropriate punctuation or conjunction:
 - They deliver documents to them they provide fast service.
 - It is nearly half past five, we cannot reach town before dark.



Word Order (4%)

- **Idiomatic Ordering of adjectives & nominal compounds**
 - tried and true vs true and tried
 - Ordering of adjectives & nominal compounds
 - A pop British band called “Spice Girl”.
 - English word order: subject verb object (SVO) Eat kids free (VSO)

Real Word Spelling Errors (2%)

- Homophones
 - there, their, they're
 - to, too, two
- Near Homophones
 - affect, effect
 - lose, loose



Content Word Choice: 20%

- Most common & least understood.
- Cover a wide range of errors & not fall into a pattern
- Collocation:
- strong / *powerful tea*strong / powerful computer
- Confusion of similar looking or sounding words:
 - Deliver the merchandise on a daily *base/basis



Influence of the Native Language

- L1 has no close equivalent construction
 - leading to difficulty in learning
 - Chinese and Russian have no equivalent of articles
- L1 has close equivalent construction
 - Easier to learn.
 - German article system similar to English
- Two languages closely related
 - transfer problems where they differ
- Unrelated languages
 - no transfer but will make more errors due to difficulty of complex English structures
 - Chinese/Russians need to learn the article rules

Goal of Grammatical Error Correction

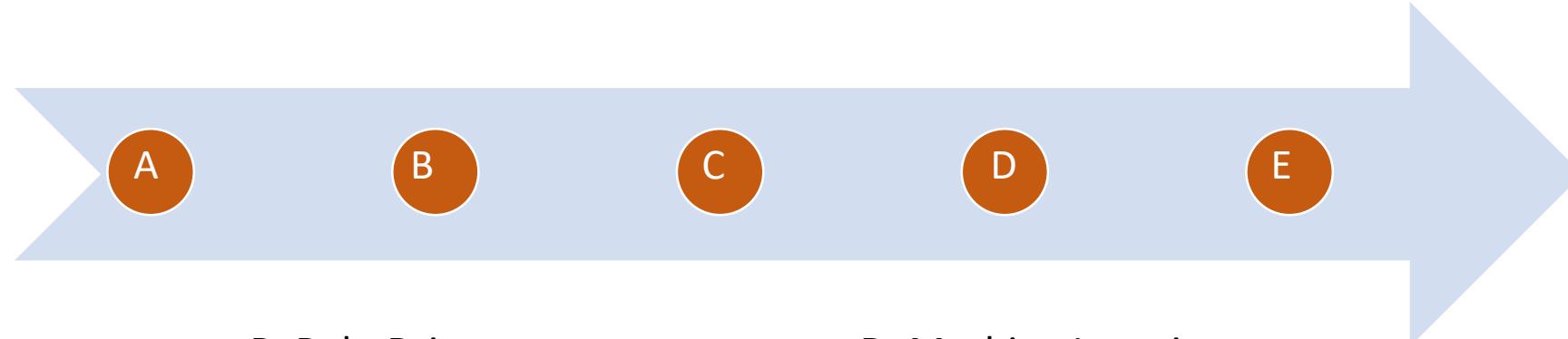
- Grammatical error correction systems, like Microsoft Word, cover error types made by native speakers.
- They rarely identify article or preposition errors.
- Need systems that focus on those problems made by Language Learners: eg, articles, prepositions, verb formation, collocations, content word choice ...

Different Approaches

A: Rule-Driven:
No Context Needed

C: Parsing: Require syntactic
structure,
in sentence and beyond

E: Whole Sentence
Correction



B: Rule-Driven:
Local Context Needed

D: Machine Learning
methods

A: No Context Needed: Simple as a Regular Expression

- Regular expressions for many verb errors:

- Infinitive formation

/to(RB)* VB[DNGZ]/ → /to(RB)* talk/

to talking → to talk

- Modal verb + have + past participle

/MD of VBD/ → /MD have VBD/

would of liked → would have liked

- Word lists

- Over-regularized morphology: I eated/ate an omelet



Background: NLP Tools

- The chickens are crossing the road.

Token	The	chickens	are	crossing	the	road	.
Lemma	the	chicken	be	cross	the	road	.
POS1	DET	NOUN	VERB	VERB	DET	NOUN	PUNCT
POS2	DT	NNS	VBP	VBG	DT	NN	.
Parse	det	nsubj	aux	ROOT	det	dobj	punct

1. Rule-Based Methods

✍ Subject-verb agreement



The chicken crosses the road .

The chicken cross the road .

The chickens crosses the road .

The chickens cross the road .

1. Find subject of verb: *nsubj*

2. If NN VBZ No error

NN VBP Error

NNS VBZ Error

NNS VBP No error

3. Generate correction from lemma + number

1. Rule-Based Methods

❖ Complications

The **chicken cross** the road.

The **chicken are crossing** the road.

The **chicken and the duck is** crossing the road.

The **chicken looks both ways and then cross** the road.

The **chicken which hate** cars crosses the road.

Crossing the road **are** dangerous for a chicken.

The **police is** outside.

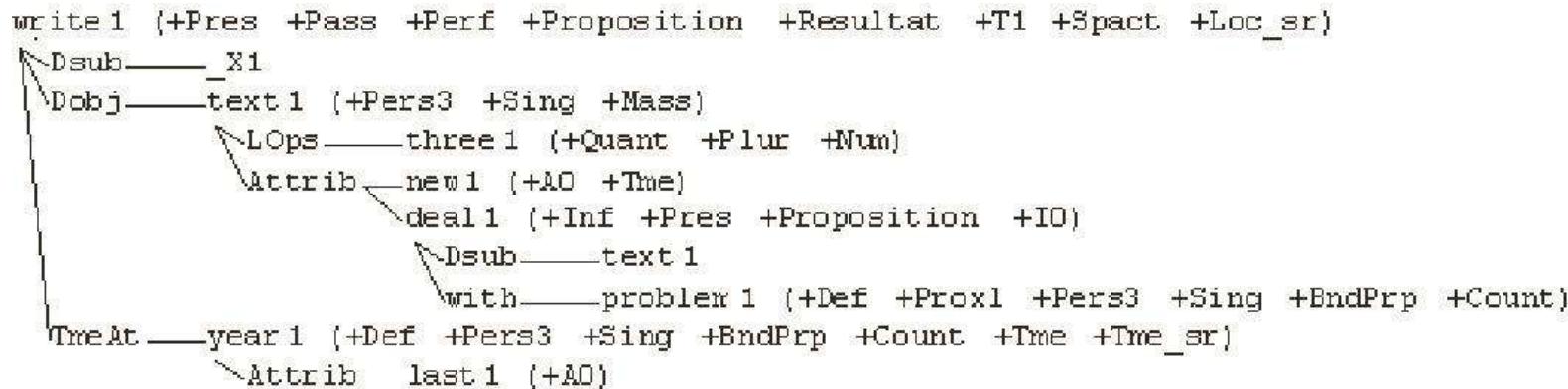
Five **pounds are** a lot of money.

C: Parsing/Logical Form

- 1980's: Before statistical parsers, modified parsers to recognize targeted errors
 - Allow parse trees that violate constraints – increment counter. Best solution has smallest index.
 - Add weights for specific violations
 - Mal-rules: Write rules to detect specific errors
 - Relax constraints on feature unification & use violations to produce feedback
- None allow for analysis of arbitrarily ungrammatical text

C: Microsoft Word

Complex hand-crafted phrase structure rules that read off of a logical form.



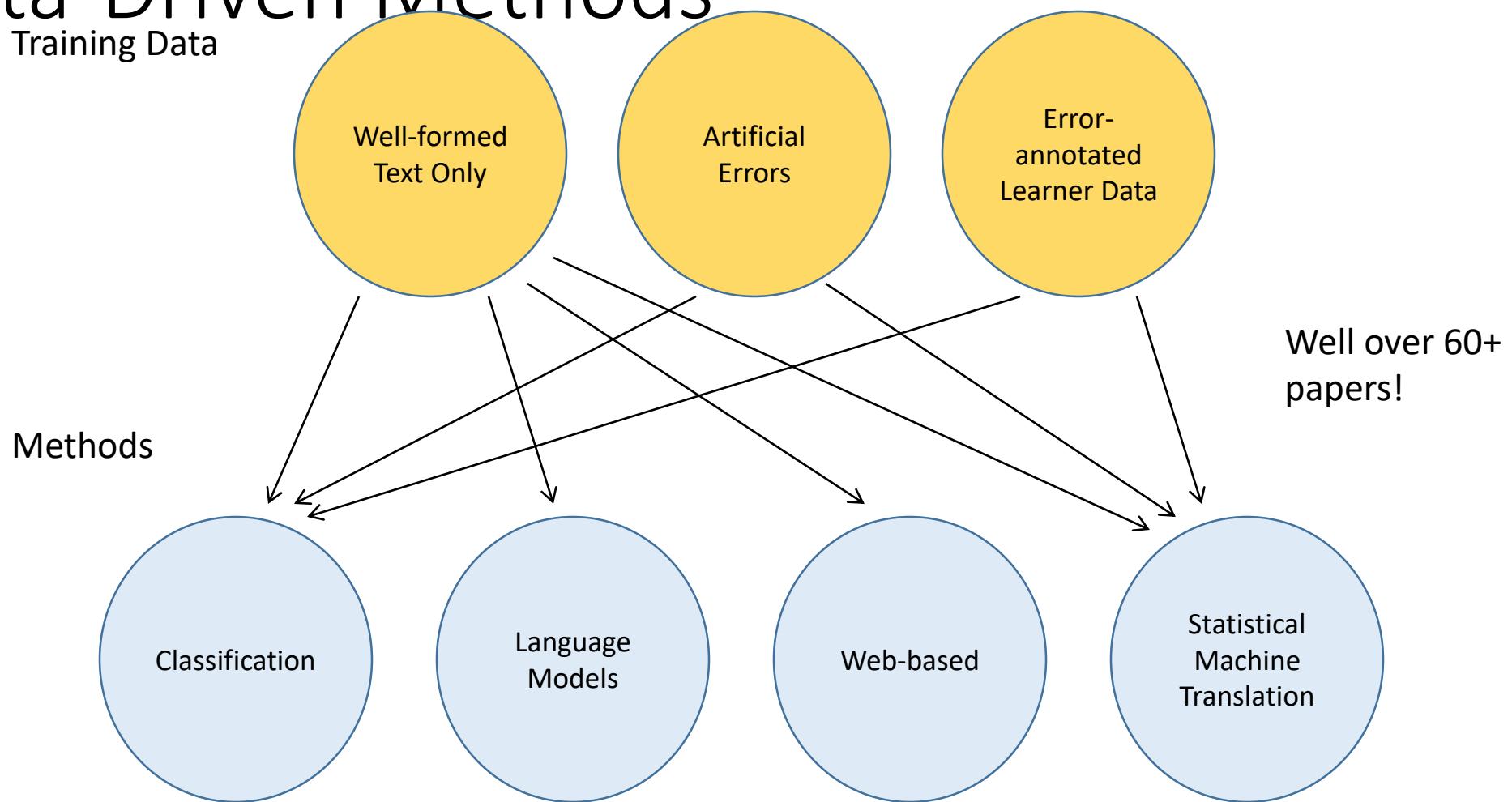
- Parse: Three new text which deal with the problem has been written.
- Pl quantifier and sg head noun. Suggest: text → texts
- Parse: Three new text which deal with the problem has been written.
- Detect subject-verb disagreement. Suggest: has → have
- Parse: no error detected

D: Error types that Require Data-Driven Methods

- Articles (*a, an, the*): presence and choice
- Prepositions (10 – 27): presence and choice
- Auxiliary verbs (*be, do, have*): presence and choice
 - A fire will break out and it can do/*be harm to people
 - A fire will break out and it can *do/be harmful to people.
- Gerund/Infinitive Confusion
 - On Saturday, I with my classmate went *eating/to eat.
 - Money is important in improving/*improve people's spirit.
- All verb errors – Lee & Seneff (2008), Rozovskaya et al (2014)

Data-Driven Methods

Training Data



Well over 60+
papers!

Methods

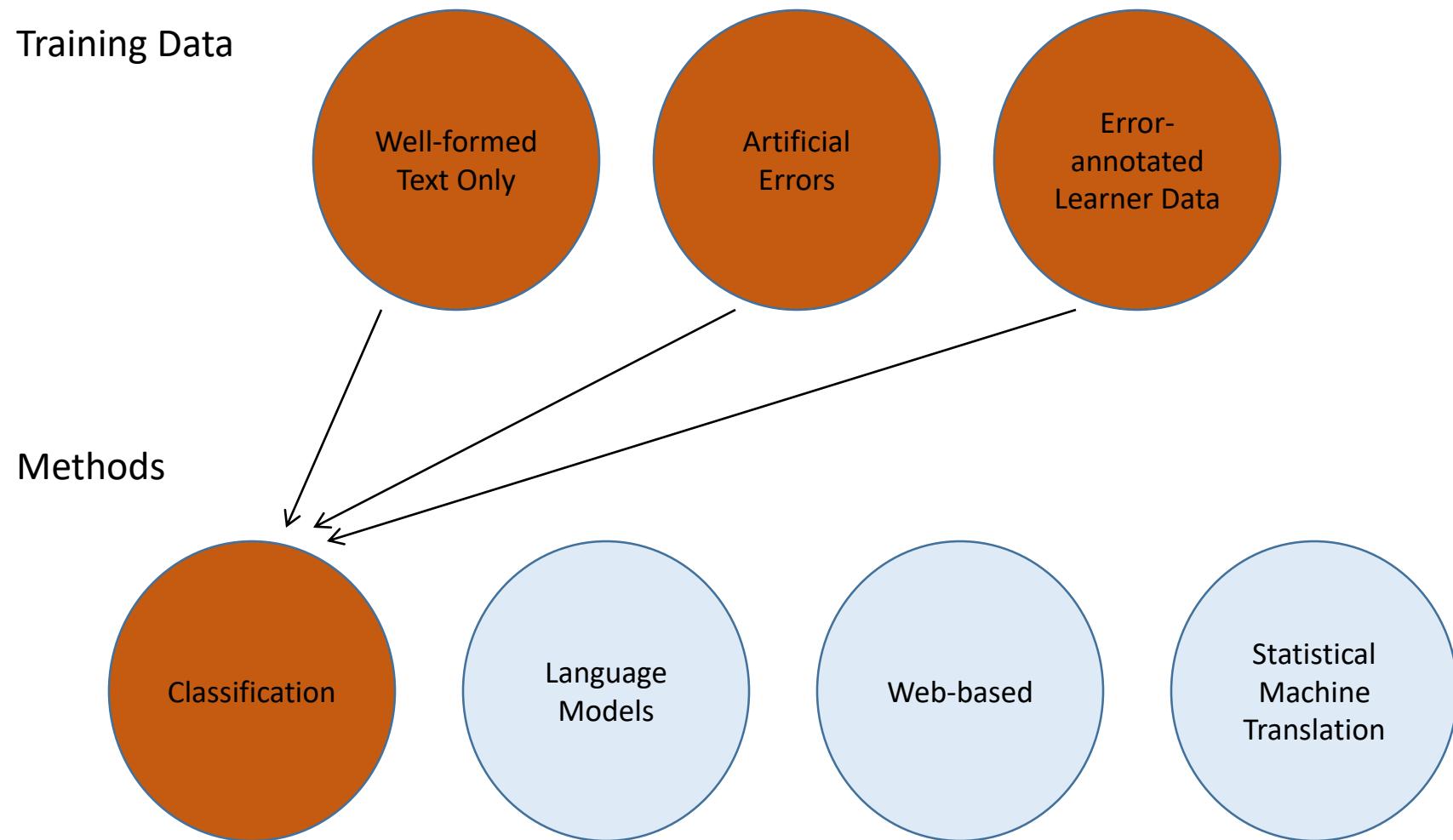
Classification

Language
Models

Web-based

Statistical
Machine
Translation

Data-Driven Methods



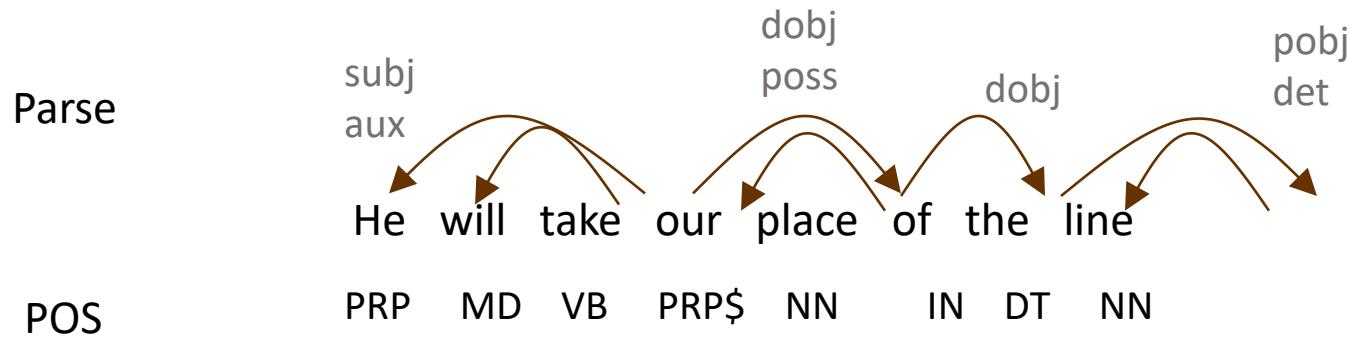
D: Data-Driven Methods

- *Supervised classification* requires:
 - Machine learning classifier (MaxEnt, SVM, Average Perceptron, etc.)
 - Data with labels for each training example

Label	Example
Correct	He will take our place in the line.
Error	He will take our place of the line.

Also need features!

Typical Features



WordNet
VerbNet
NER taggers
Semantic Role Labelers

Semantic

1-gram: place, the
2-gram: our-place, place-of, of-the, the-line
3-gram: our-place-of, place-of-the, of-the-line

Source

Writer's word(s) selection
L1 of writer
Genre of writing

N-grams

Types of Training Data

1. Training on examples of correct usage *only*
2. Training on examples of correct usage and *artificially generated errors*
3. Training on examples of correct usage and *real learner errors*

Choice of training data largely determined by availability of data

1. Training on Correct Usage

- Prior to 2010, very few error-annotated corpora to get enough examples of errors for ML
- Solution: train on examples of correct usage only
 - [Han et al., 2006; Tetreault and Chodorow, 2008; Gamon et al., 2008; Felice and Pulman, 2009]
- Advantages: plenty of well-formed text available
 - Google n-gram corpus to build language models
 - Large corpora such as news, Wikipedia, etc. to derive features from
- Challenges:
 - Best to match genre of learner writing, so need lots of well-formed student essays
 - Does not exploit any information of when or how errors tend to appear

2. Artificial Errors



- Training only on examples of correct usage has performance limitations
- Approximate learner writing by introducing artificial errors into a corpus of well-formed text
- Training instances
 - “Positive”: well-formed text
 - “Negative”: artificial errors
- Add a feature to capture transformation from erroneous choice to correct choice
- Challenge: determining the best way to approximate the errors

3. Error-Annotated Corpora

- Most common approach in shared tasks now that there are some labeled corpora available
- Use writer's word choice as a feature
- Some key works:
 - Han et al. (2010): showed that having a large corpus of annotated essays significantly outperformed positive-examples-only training on prepositions
 - Dahlmeier & Ng (2011): showed that Alternating Optimization Techniques worked well with error-annotated data for prepositions
 - Most CoNLL 2014 shared task systems

Classification

- They **were eaten** ice-cream when I **arrived**.
- Goal: Predict the correct form of a verb.

Verb Forms	Part-Of-Speech	Example
Infinitive	VB	eat
Past Tense	VBD	ate
Gerund or Present Participle	VBG	eating
Past Participle	VBN	eaten
Non-3 rd Person Singular Present	VBP	eat
3 rd Person Singular Present	VBZ	eats

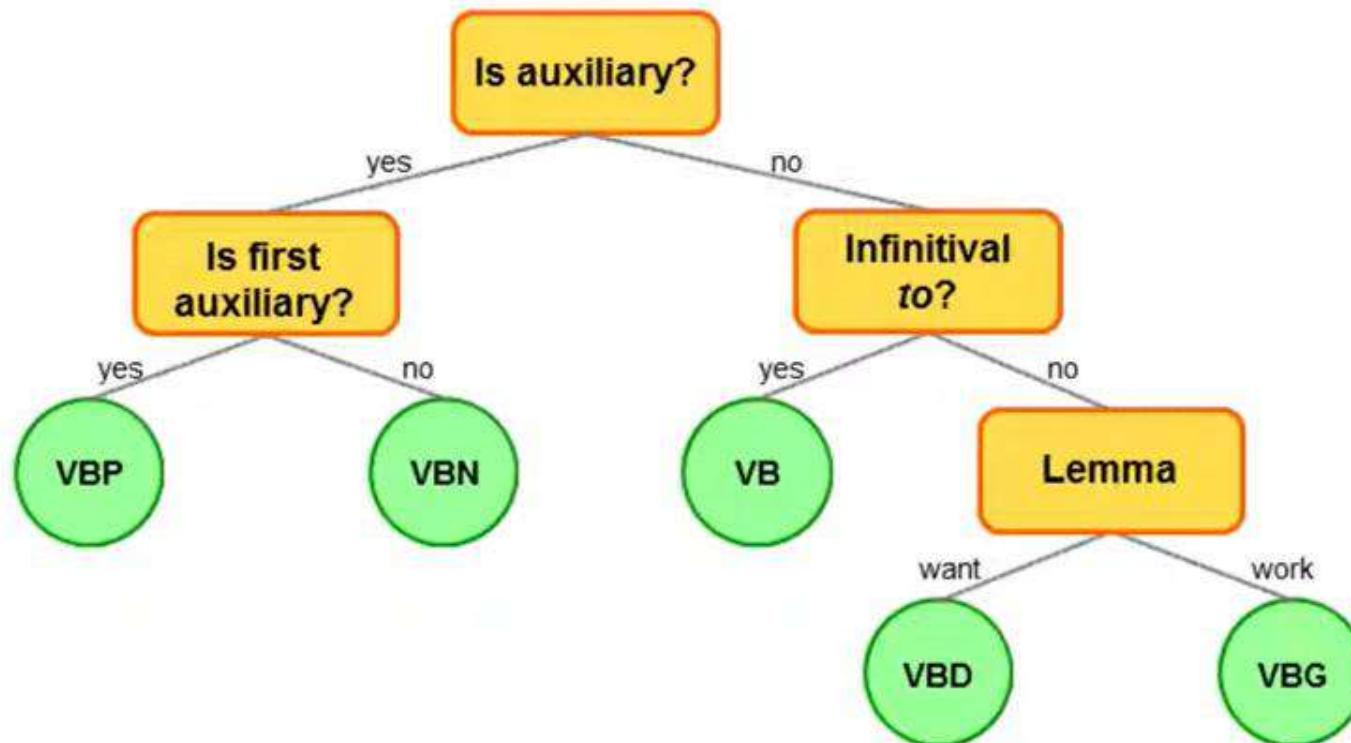
Classification - Features

- They **were eaten** ice-cream when I **arrived**.
- Instead of rules, define features!

Sample features	Values	Example: “were”
Is auxiliary?	Yes/No	Yes
Is first auxiliary?	Yes/No	Yes
Infinitival to?	Yes/No	No
Is ROOT verb?	Yes/No	No
Lemma	Lemma	be
N-gram	N words	They were eaten
Time expressions?	today, yesterday, ...	when ...
Target	VB, VBZ, VBP, ...	VBD

Classification - training

- ✍ Use machine learning to weight each feature.
 - Logistic Regression, Decision Trees, Support Vector Machines, ...



Classification



Advantages

- More flexible than rules.
- Incorporates human intuition.
- Only requires native data.



Disadvantages

- Feature engineering can be complicated.
- Better for smaller confusion sets.
- Only targets single error types.

Language Models

- ✍ Which is “best”?
 - Why did the chicken cross the **the**?
 - Why did the chicken cross the **line**?
 - Why did the chicken cross the **street**?
 - Why did the chicken cross the **road**?
- ✍ In a given context, some words more probable than others.
- ✍ Use this property to flag improbable sequences as errors.

Language Models

- ✍ N-gram models

1	Why	did	the	...
2	Why did	did the	the chicken	...
3	Why did the	did the chicken	the chicken cross	...
4	Why did the chicken	did the chicken cross	the chicken cross the	...

- ✍ Low N: High frequency, small context

- ✍ High N: Low frequency, large context

- ✍ Bigram (n=2) sentence probability

$$\rightarrow P(\text{Why}|\langle s \rangle) \times P(\text{did}|\text{Why}) \times P(\text{the}|\text{did}) \times P(\text{chicken}|\text{the}) \times \dots$$

Language Models

- ✍ Train a model on native text; e.g. Wikipedia
- ✍ Generate a confusion set
 - in, at, from, on, with, ...
- ✍ Score alternatives to find the best
 - I often work **in** home. |
 - I often work **at** home. ━━
 - I often work **from** home. ━━━
 - I often work **on** home. |
 - I often work **with** home. |
 - ...

Language Models

- Targeted Approach: can use LM scores over phrase or sentence for correction and detection

at	0.1
by	0.2
for	0.1
He will take our place in the line.	→ 0.3
from	0.0
to	0.1
with	0.1

- Similar to Web-based approach though one has more control of the data
- Nearly half of the HOO2012 systems used LMs

Language Models - Advantages

- ☞ Only require (lots of) native data
- ☞ Fairly easy to implement
- ☞ Can detect all error types
- ☞ Collocation errors
 - I have a **strong/powerful** computer.
 - This **effected/affected** me deeply.
- ☞ Versatile

Language Models - Disadvantages

- ☞ Probability is not grammaticality.
 - Student: I is ...
 - Teacher: No, you must always say “I am”.
 - Student: Ok. I am the ninth letter of the alphabet.

- I am at home. 
- I was at home. 

- ☞ Rare/unseen words: paraklausithyron, covfefe
- ☞ Long range dependencies
 - The house on the corner with the blue windows is new.

Methods: Web-Based Methods

- Language learners will typically look at counts returned by search engine to figure out best word to use
- What happens when we use this simple methodology?
 - Select “target word” and search for alternatives
 - Select alternative with top web count

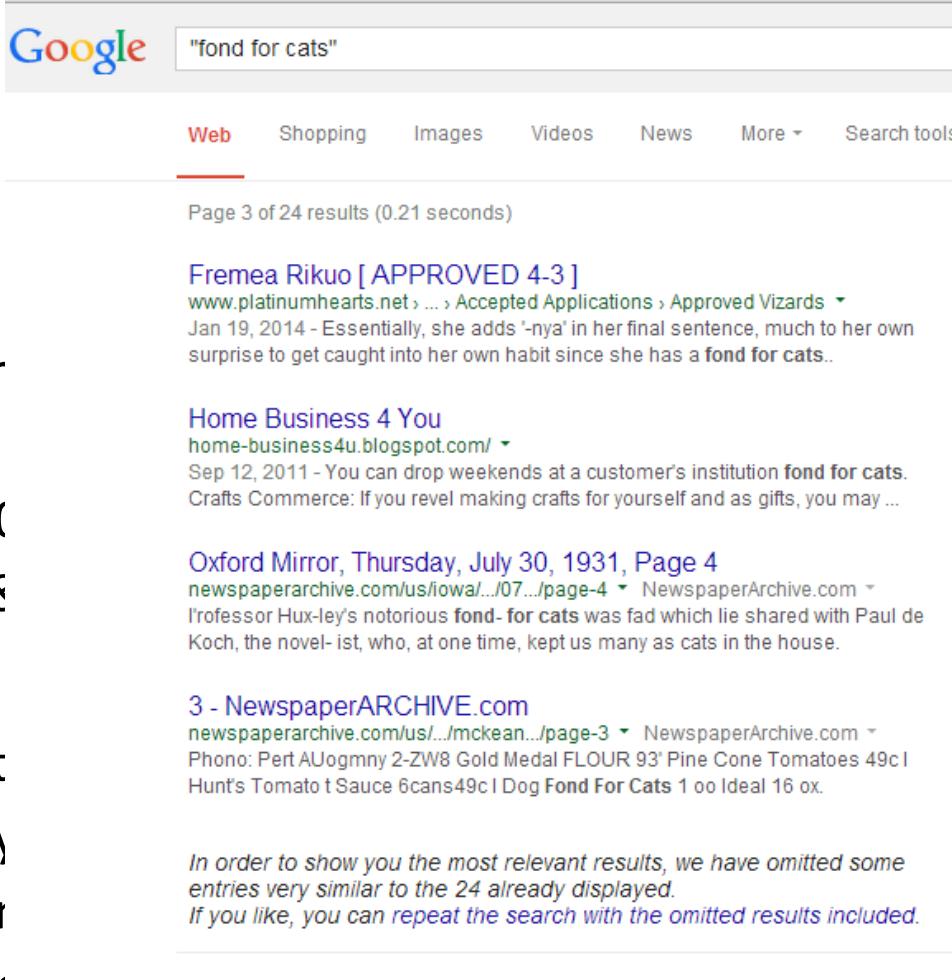
Web-Based Methods

Phrase	Google Count	Bing Count
“fond of cats”	638,000	42,800
“fond for cats”	178	2
“fond by cats”	0	0
“fond to cats”	269	5
“fond with cats”	13,300	10



Method

- Prior work shows approaches
 - Yi et al. (2011)
 - Tetreault & Smith (2012)
- Issues:
 1. No POS tagging
 2. Search syntax
 3. Constraint
 4. Search counts are for pages not instances
 5. Search engines behave differently



A screenshot of a Google search results page for the query "fond for cats". The search bar at the top contains the query. Below it, the "Web" tab is selected. The results show four entries:

- Fremea Rikuo [APPROVED 4-3]**
www.platinumhearts.net › ... › Accepted Applications › Approved Wizards ›
Jan 19, 2014 - Essentially, she adds '-nya' in her final sentence, much to her own surprise to get caught into her own habit since she has a fond for cats..
- Home Business 4 You**
home-business4u.blogspot.com/ ›
Sep 12, 2011 - You can drop weekends at a customer's institution fond for cats. Crafts Commerce: If you revel making crafts for yourself and as gifts, you may ...
- Oxford Mirror, Thursday, July 30, 1931, Page 4**
newspaperarchive.com/us/iowa/.../07.../page-4 › NewspaperArchive.com ›
Professor Huxley's notorious fond-for cats was fad which he shared with Paul de Koch, the novelist, who, at one time, kept us many as cats in the house.
- 3 - NewspaperARCHIVE.com**
newspaperarchive.com/us/.../mckean.../page-3 › NewspaperArchive.com ›
Phono: Pert AUogmny 2-ZW8 Gold Medal FLOUR 93' Pine Cone Tomatoes 49c | Hunt's Tomato t Sauce 6cans49c | Dog Fond For Cats 1 oo Ideal 16 ox.

In the bottom right corner of the search results, there is a note: *In order to show you the most relevant results, we have omitted some entries very similar to the 24 already displayed. If you like, you can repeat the search with the omitted results included.*

ods

ot over classification
tion of queries
nine L1 specific errors

Statistical Machine Translation

✍ Greek to English

Άφησα το βιβλίο μου στο σπίτι .
I left my book at home .

✍ “Bad” English to “Good” English

I leaved my book in the home .
I left my book at home .

Statistical Machine Translation

- ✍ Requires parallel data

Name	Type	# Words	Error Types?	Availability
CLC	Essays	~29 million	Yes (80)	Private
Lang 8	Web Forum	~12 million	No	Public
NUCLE	Essays	~1.2 million	Yes (28)	Public

- ✍ Issues

- >10x smaller than translation corpora
- Annotation quality
- Proficiency levels
- Genre

Error-Annotated Corpora

NUCLE

- **National University of Singapore Corpus of English**
- 1,450 essays by Singapore college students
- Used in CoNLL shared tasks
- Publically available

FCE

- 1,244 essays from **First Certificate in English** exam (CLC subset)
- Used in HOO 2012 task
- Includes score, error annotation and demographics
- Publically available

HOO2011

- Hand corrected papers from **ACL Anthology**
- 38 conference papers
- Publically available

CLEC

- **Chinese Learners of English Corpus**
- 1M words
- Five proficiency levels
- Inexpensive

Other Learner Corpora

TOEFL11

- **ETS Corpus of Non-Native English**
- 12,100 essays (1,100 essays each for 11 different L1s)
- Includes proficiency information
- Available through LDC

ICLE

- **International Corpus of Learner English**
- 3.7 M from over 16 different L1s
- Partially error-annotated
- Needs to be licensed.

Lang-8

- **Language Learner Social Community Website**
- Nearly 200,000 Learner journal entries with community corrections
- Need a script to extract data (Mizumoto et al., 2011)



mestre

Jul 26, 2014 12:47

- ✍ I'm going to an afternoon mini-concert that's going to take place at the small stage in the shopping mall.
- ✓ I'm going to an afternoon mini-concert that's going to take place at a small stage in the shopping mall.

"the" may be correct here, but since we're not familiar with the shopping mall in question, "a" feels more natural.

1 people think this correction is good.

- ✍ My daughter in law is going to play the sax in it.
- ✓ My daughter in law will be playing the sax.

More of a suggestion than anything.

1 people think this correction is good.

- ✍ She's cool, active, and she's well into the last trimester of pregnancy, and proud.
- ✓ She's cool, active, and she's well into the last trimester of pregnancy, and proud.

This sentence sounds rather unnatural with the 'and' both here and at the end portion, so I removed this bit. Adding "of it" after "proud" will make that portion stronger and more specific, but may not be your intention.

1 people think this correction is good.

2
Entries Written

2
Entries Written

2
Entries Written

Latest entries

● A New Trend of University English Entrance Examination (9)

● A Mini-Concert (18)

[Read more](#)

Latest comments

● 07:03 いしぐろ Ishiguro○_○

● 05:22 Kalyak

● 04:50 tony

● 03:40 student113

● 00:38 建

Statistical Machine Translation

- ☛ Input: Parallel sentence pairs
- ☛ Phrase-based SMT learns the mapping between phrases.
 - [το βιβλίο μου → my book]
 - [discuss about → discuss]
 - [informations → information]
 - [as result → as a result]
 - [the → the]
- ☛ The mappings are stored in a *phrase table*.
- ☛ Translate using the phrase table and a language model.

Statistical Machine Translation

Goal: Find the best translation

Let 's discuss about this informations .

Let 's discuss about this informations .

Lets talk over the information ?

Let 's discuss the information !

talk about this information

these informations

Let 's discuss this information .

Two Classes of GEC / SMT

1. Noisy Channel Model

- View error correction as the process of *translating* from learner English to fluent English

2. Round Trip Machine Translation

- View SMT as a “black box” and use MT engine to generate possible corrections

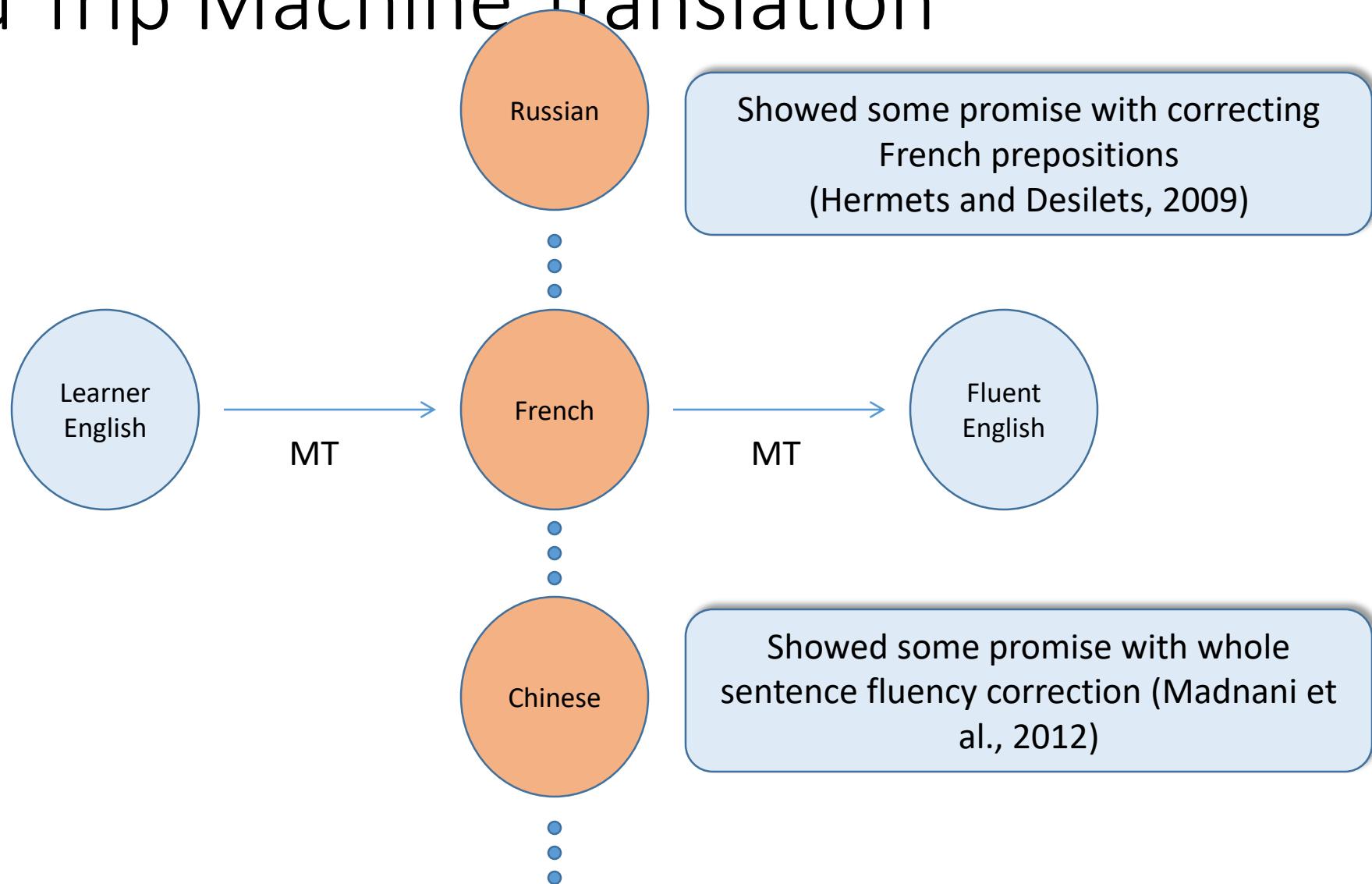
Noisy Channel Model

- Re-train MT system with examples of error phrases (or sentences) and their corrections
- Dependent on having enough error-annotated data
- Some examples:
 - Brocket et al. (2006): use artificial errors to train SMT to correct mass noun errors
 - Park & Levy (2011): use technique with FSTs

Round Trip Machine Translation

- Use pre-existing MT system to translate a sentence into another language and translate back into English
 - Thus does not use learner data
- Preliminary pilot studies with this method show some potential

Round Trip Machine Translation



Statistical Machine Translation



Advantages

- Can detect and correct all error types simultaneously
- Handles interacting errors
- No feature engineering or expert knowledge
- State-of-the-art



Disadvantages

- Long training time: several days
- Require lots of parallel training data
- Out-Of-Vocabulary words (OOV)
- Hard to customise

Commercial Grammar Check tools

Tool	Primary Approach	How It Works
Grammarly	Agentic AI & Generative AI	Deploys a team of "AI agents" (e.g., Proofreader, Paraphraser, Tone Adjuster) that proactively analyze context, intent, and style to offer holistic writing suggestions.
QuillBot	Generative AI (LLM)	Its core function is paraphrasing, which is powered by a generative LLM. Its grammar check and other tools are also built with this AI-first approach to rewrite and correct text.
ProWritingAid	Hybrid (Analytical AI + Generative AI)	Uses AI to run 25+ distinct analytical reports (e.g., "Sticky Sentences," "Pacing"). It <i>separately</i> uses generative AI for specific tasks like "Sparks" (rephrasing) and "Critiques."

Open Source Grammar Check tools

Language Tool	Hybrid (Rule-Based + AI)	The free, open-source core is a powerful rule-based engine (using XML/Java rules). The premium version adds a proprietary AI layer for advanced style and rephrasing.
After the Deadline	Statistical NLP (N-grams)	A classic NLP model that uses statistical analysis (n-grams) and neural networks to find "real word errors" by checking the probability of word sequences against a large corpus.
Gramformer	Machine Learning (Transformers)	A Python framework (not a full app) built on Transformer models (like those in LLMs). It's an ML model fine-tuned specifically to detect and correct grammatical errors.



Notable Tools and Models

- **GECToR**: Transformer-based token-level corrector (very fast and accurate)
- **Errant**: Tool for evaluating GEC systems
- **OpenNMT / FairSeq**: Frameworks for building GEC models

System/Model	Approach	Strengths	Weaknesses	Used By
GECToR	Token-level Transformer (edit tagging)	Fast, interpretable, state-of-the-art for GEC	Needs annotated data, limited fluency handling	Open-source, academic
T5 / BART	Seq2Seq transformer (text-to-text)	Powerful fluency correction, few-shot learning	Slow, may hallucinate corrections	General-purpose (used in some GEC research)
BERT / RoBERTa Classifiers	Error detection, often paired with correctors	High accuracy detection	Not standalone for full correction	Grammarly (likely part of stack), research
mT5 / XLM-R	Multilingual Transformers	Supports many languages	Computationally heavy	Google, multilingual GEC tasks
Rule-Based Systems	Hand-crafted grammar and style rules	Transparent, fast	Rigid, low coverage	LanguageTool, old Microsoft Word
Grammarly	Hybrid: deep learning + rules + ensembles	Fluent, context-aware, real-world tool	Proprietary, limited transparency	Grammarly
Google Docs Grammar Tool	Transformer-based, real-time	Good latency, seamless user experience	Mostly grammar, not style	Google Workspace

Agentic AI and Spellcheck/Grammarcheck

- **Preprocessing Step** – Agentic AI often includes a spellcheck and grammar-correction module to clean user input before analysis.
- **Improved Understanding** – Corrected text ensures better intent recognition and semantic parsing for downstream reasoning.
- **Embedded or External Tools** – Systems may use internal NLP models, APIs (like Grammarly or LanguageTool), or LLM-based self-correction.
- **Pipeline Integration** – Grammar/spell check sits in the preprocessing stage, feeding into semantic analysis → reasoning → action/output loop.

NLP Evaluation Metrics

- True Positive (TP) “hit”
 - Flags *I am going for walk this afternoon.*
- False Positive (FP)
 - Flags *I am going for a walk this afternoon.*
- True Negative (TN)
 - Not flag *I am going for a walk this afternoon.*
- False Negative (FN) “miss”
 - Not flag *I am going for walk this afternoon.*



NLP Evaluation Metrics

$$Precision = \frac{TP_s}{TP_s + FP_s}$$

$$Recall = \frac{TP_s}{TP_s + FN_s}$$

$$F\text{-}score = \frac{Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP_s + TN_s}{TP_s + TN_s + FP_s + FN_s}$$

Grammar Checking Approaches

1. Rule-Based Approaches:

- These methods employ predefined grammar rules and patterns to identify and correct errors in text.
- Grammar rules are formulated based on linguistic principles and syntax.
- For instance, identifying subject-verb agreement errors, incorrect verb tense usage, punctuation errors, and sentence fragments.

2. Statistical Approaches:

- Statistical methods leverage machine learning algorithms to identify grammatical errors by analyzing large amounts of text data.
- These models learn patterns and probabilities of correct language usage based on annotated corpora.
- They can detect errors based on deviations from statistically common linguistic structures.

3. Hybrid Approaches:

- Combining rule-based and statistical methods can enhance the accuracy of grammar checking.
- Hybrid systems utilize both rule-based grammar rules and statistical models to detect and correct errors, leveraging the strengths of each approach.

Deep Learning Approaches

Deep Learning Approaches:

- Neural network-based models, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and transformers, have been employed for grammar checking.
- These models can learn complex linguistic patterns and relationships, aiding in detecting errors and suggesting corrections.

Error-Correcting Models:

- Some NLP models are specifically trained to detect and correct errors.
- For instance, sequence-to-sequence models are trained on pairs of correct and incorrect sentences, learning to generate corrections for the given erroneous sentences.

Language Models for Contextual Understanding:

- Recent advancements in large language models, like BERT, GPT, and their variants, have improved grammar checking by incorporating contextual understanding.
- These models can comprehend the context of sentences and suggest corrections based on the surrounding words.

Grammar Checking Approaches

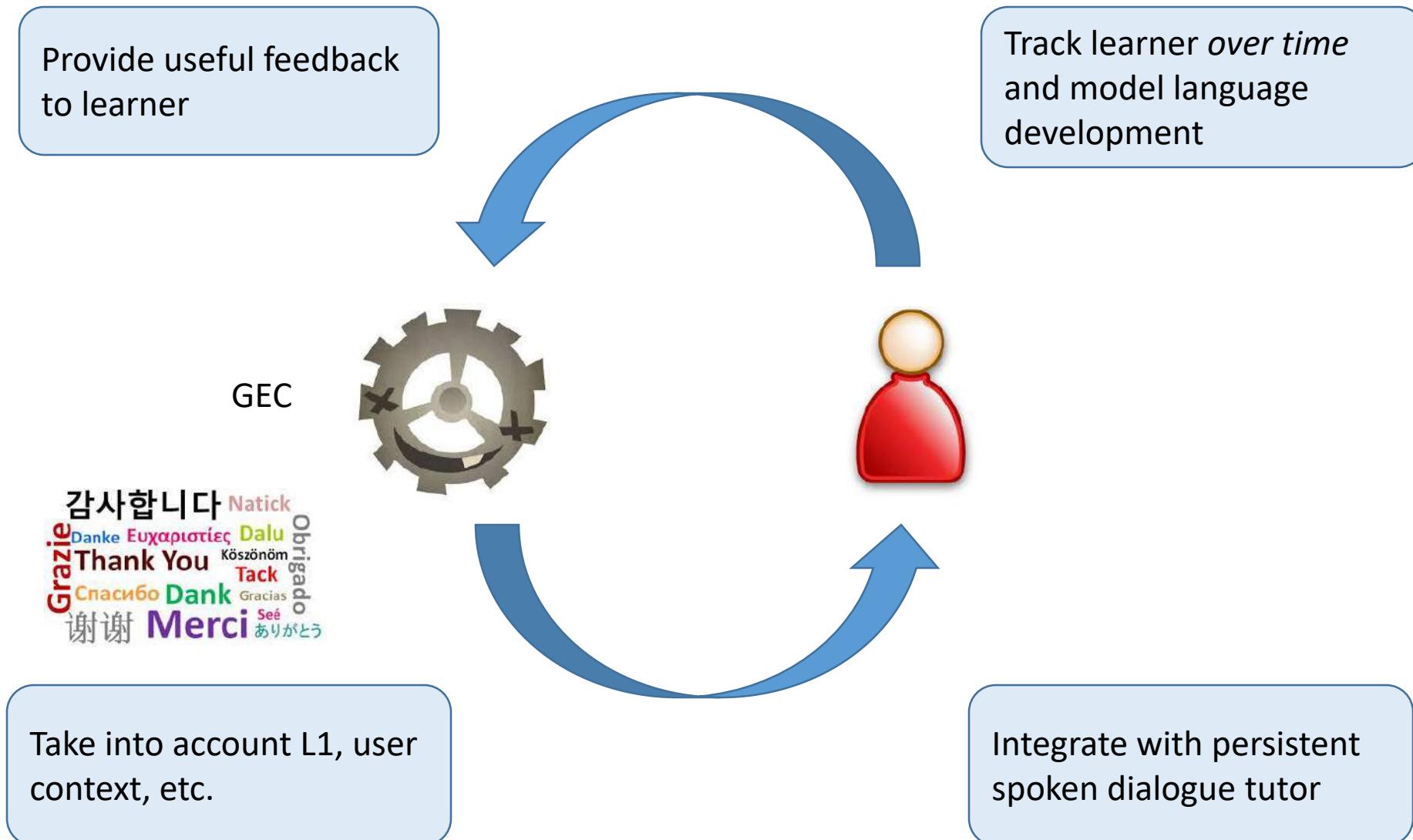
User Feedback and Machine Learning Iteration:

- Some grammar checkers improve over time by incorporating user feedback.
- Machine learning algorithms learn from user corrections to enhance their accuracy in identifying and correcting errors.

Fine-tuning and Transfer Learning:

- Pre-trained language models can be fine-tuned on specific grammar checking tasks by training them on annotated datasets.
- Transfer learning allows models to adapt their knowledge from general language understanding to the specific task of grammar checking.
- Transfer learning, used in machine learning, is the reuse of a pre-trained model on a new problem. In transfer learning, a machine exploits the knowledge gained from a previous task to improve generalization about another.

What is the future of GEC?



What is the future of GEC?

- A system which can automatically transform one noisy sentence to a fluent sentence...without a change in meaning

Having discuss all this I must say that I must rather prefer to be a leader than just a member.



GEC

After discussing all this I must say that I'd prefer to be a leader than a follower.

What is the future of GEC?

- System need not simply be a text to text transformation, could also take into account:
 - Other sentences in document
 - Context of document (writer's intention)
 - Register
 - Who the document is for
 - Prior sentences writer has produced (personalization)

Scope of research in Spellcheck and Grammarcheck

No.	Research Area	Focus	Example / Challenge
1	Context-aware models	Semantic understanding	“Their” vs “there”
2	Multilingual & code-mixed text	Cross-language adaptation	Hindi-English text
3	Privacy & on-device NLP	Local model deployment	Federated learning
4	User/domain adaptation	Personalized writing feedback	Legal or academic writing
5	Evaluation & explainability	Human-centered metrics	Transparent suggestions

Applications of GEC

- Automated Essay Scoring
- Native Language Identification
- MT Quality Estimation
- Noisy data processing
 - Social Media / normalization
 - MT post-processing
 - Assistive Tech: GEC of automatic closed captions

Summary

- Provided a history of GEC
- Described popular methodologies for correcting language learner errors
- Grammatical Error Correction one of the oldest fields and applications of NLP
- Still much work to be done as performance is still low!

References

- https://nlp.johnsnowlabs.com/analyze_spelling_grammar
- <https://direct.mit.edu/coli/article/49/3/643/115846/Grammatical-Error-Correction-A-Survey-of-the-State>
- <https://blogs.nvidia.com/blog/microsoft-triton-ai-grammar-word/>
- <https://paperswithcode.com/datasets?q=&v=lst&o=newest&task=grammatical-error-correction&mod=texts&page=1>
- <https://onlinelibrary.wiley.com/doi/10.1002/spe.2110>
- <https://www.youtube.com/watch?v=3rVn14m8zaM>
- <https://www.youtube.com/watch?v=pYV8OydsnQA>
- <https://www.youtube.com/watch?v=9ZkM-jn-EWA>
- <https://www.hindawi.com/journals/sp/2021/4213791/>
- <https://www.youtube.com/watch?v=3rVn14m8zaM>
- <https://sapling.ai/grammar-check>
- https://www.youtube.com/watch?v=2RU5egfeb_g
- <https://www.youtube.com/watch?v=0MiFUES-0F4>
- [Analyze Spelling & Grammar - Spark NLP Demos & Notebooks](#)

- <https://colab.research.google.com/github/piyush0511/SpellChecker-AutoCorrect/blob/main/SpellCheck%20-%20seq2seq.ipynb#scrollTo=eqcJXSpf0fPW>
- <https://colab.research.google.com/drive/1ysEKrw LE2jMndo1snrZUh5w87LQsCxk>



BITS Pilani
Pilani Campus

Natural Language Processing Applications

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



Session Content

- Question Answering – Business Value
- QA versus Conversational AI
- QA Systems Journey
- Current Approaches
 - Extractive QA
 - Generative QA
 - Retrieval Based QA
 - RAG
 - Hybrid QA
- Case study

What is Question Answering

A field of Natural Language Processing (NLP) focused on building systems that can automatically answer questions posed by humans in natural language.

The Goal: Move beyond simple keyword search to provide a **direct, accurate, and concise answer** to a user's query.

- **QA:** You ask "what temperature to bake bread," and it answers, "**350°F (175°C).**"

Why is QA So Important? (The Business Value)



Customer Support: Instantly answer 80% of repetitive customer questions 24/7.

Enterprise Knowledge: Allow employees to "ask questions" of your company's internal documents (HR policies, technical manuals, sales data).

Data Analysis: Enable executives to ask, "What was our top-selling product in Q3?" and get a direct answer, not a complex dashboard.

User Experience: Powers the voice assistants and search engines we use every day (Siri, Alexa, Google).

QA versus Conversational AI

Feature	Question Answering (QA)	Conversational AI
Core Goal	1. Answer a single question.	1. Hold a multi-turn dialogue.
Scope	2. Transactional (one-shot).	2. Relational and Goal-Oriented.
Context	3. Stateless (forgets immediately).	3. Stateful (must remember conversation history). You: "I need to book a flight." Bot: "Sure, where are you flying to?"
Example	You: "What is the capital of France?" Bot: "Paris."	You: "Paris." Bot: "Great. When do you want to leave for Paris?"

When should we use conversational AI for QA

Conversational AI can absolutely perform QA, but you should only use it when the QA task is complex enough to benefit from:

- reasoning
- multi-turn context
- unstructured text understanding
- flexibility

For simple, repetitive questions, **conversational AI is overkill**—a simple QA system is cheaper, faster, safer.

QA Real world applications

1. Questions are predictable or repetitive

Example: "What are your store hours?", "What is the refund policy?"

2. Answers must be fixed and controlled

Example: Banking compliance messages, Medical safety instructions

3. You need very fast responses with low cost

Example: An FAQ chatbot handling thousands of queries per minute

4. High accuracy is required and hallucinations are risky

Example: Legal policies, insurance rules, company procedures

5. Single-turn interactions (no context needed)

Example: "What is the capital of Japan?"

6. Information is pulled directly from a database or documents

Example: "Show me my order status.", "What is CPU usage right now?"

Conversational AI Real world applications



1. Multi-turn conversations

Example: User: “Book a flight.”

Bot: “Where to?”

User: “London.”

2. Questions require reasoning or explanation

Example: “Explain how transformers differ from SSMs.”, “Summarize this report.”

3. The question depends on previous context

Example: User: “Who is ?”, “How old is he?”

4. Handling unclear or ambiguous questions

Example: “Tell me about delivery charges.” (Bot: “Domestic or international?”)

5. Working with unstructured documents (PDFs, long texts)

Example: “What are the main points of this 30-page contract?”

6. Conversational tasks beyond QA

Example: recommending products, tutoring, guiding, troubleshooting.

Apple's Siri



 **WolframAlpha**^{computational... knowledge engine}

how many calories are in two slices of banana cream pie? ≡

[Examples](#) [Random](#)

Assuming any type of pie, banana cream | Use pie, banana cream, prepared from recipe or pie, banana cream, no-bake type, prepared from mix instead

Input interpretation:

pie	amount	2 slices	total calories
	type	banana cream	

Average result: **702 Cal** (dietary Calories) Show details

Types of Questions

- Factoid questions
 - *Who wrote “The Universal Declaration of Human Rights”?*
 - *How many calories are there in two slices of apple pie?*
 - *What is the average age of the onset of autism?*
 - *Where is Apple Computer based?*
- Complex (narrative) questions:
 - *In children with an acute febrile illness, what is the efficacy of acetaminophen in reducing fever?*
 - *What do scholars think about Jefferson’s position on dealing with pirates?*

Commercial systems: mainly factoid questions

Where is the Louvre Museum located?	In Paris, France
What's the abbreviation for limited partnership?	L.P.
What are the names of Odin's ravens?	Huginn and Muninn
What currency is used in China?	The yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	drums
What is the telephone number for Stanford University?	650-723-2300

Journey of QA Systems

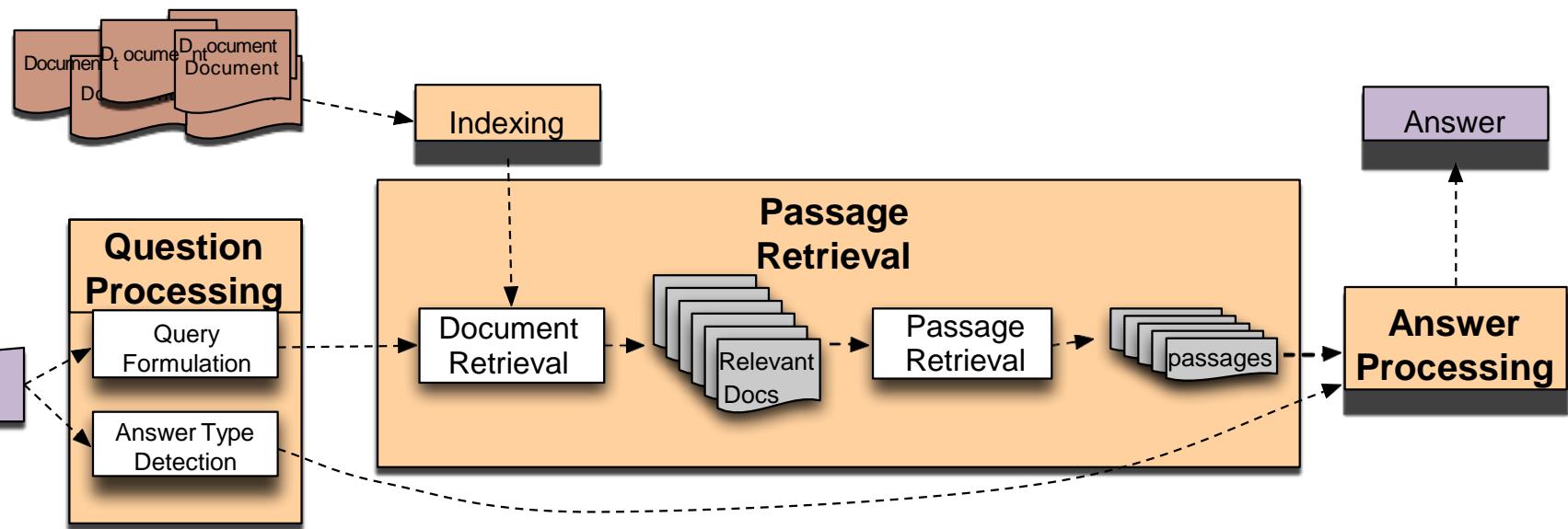
Evolution Path:

IR-based → Knowledge-based → Hybrid (IR + Knowledge) →
Deep Learning → Transformers → RAG

IR- based:

- **Approach:** Retrieve relevant documents/passages from large text collections.
- **Techniques:** Keyword matching, TF-IDF, BM25.
- **Examples:** Early QA in search engines (AskJeeves, early Google).
- **Limitations:** No true understanding; returns documents, not direct answers.

IR-based Factoid QA



IR-based Factoid QA

- **QUESTION PROCESSING**
 - Detect question type, answer type, focus, relations
 - Formulate queries to send to a search engine
- **PASSAGE RETRIEVAL**
 - Retrieve ranked documents
 - Break into suitable passages and rerank
- **ANSWER PROCESSING**
 - Extract candidate answers
 - Rank candidates
 - using evidence from the text and external sources

Knowledge--based approaches (Siri)



- Build a semantic representation of the query
 - Times, dates, locations, entities, numeric quantities
- Map from this semantics to query structured data or resources
 - Geospatial databases
 - Ontologies (Wikipedia infoboxes, dbpedia, WordNet, Yago)
 - Restaurant review sources and reservation services
 - Scientific databases

Hybrid approaches (IBM Watson)



- Build a shallow semantic representation of the query
- Generate answer candidates using IR methods
 - Augmented with ontologies and semi-structured data
- Score each candidate using richer knowledge sources
 - Geospatial databases
 - Temporal reasoning
 - Taxonomical classification

Question Answering: IBM's Watson



- Won Jeopardy on February 16, 2011!

WILLIAM WILKINSON'S
“AN ACCOUNT OF THE PRINCIPALITIES OF
WALLACHIA AND MOLDOVIA”
INSPIRED THIS AUTHOR’S
MOST FAMOUS NOVEL



Bram Stoker

IBM Watson QA

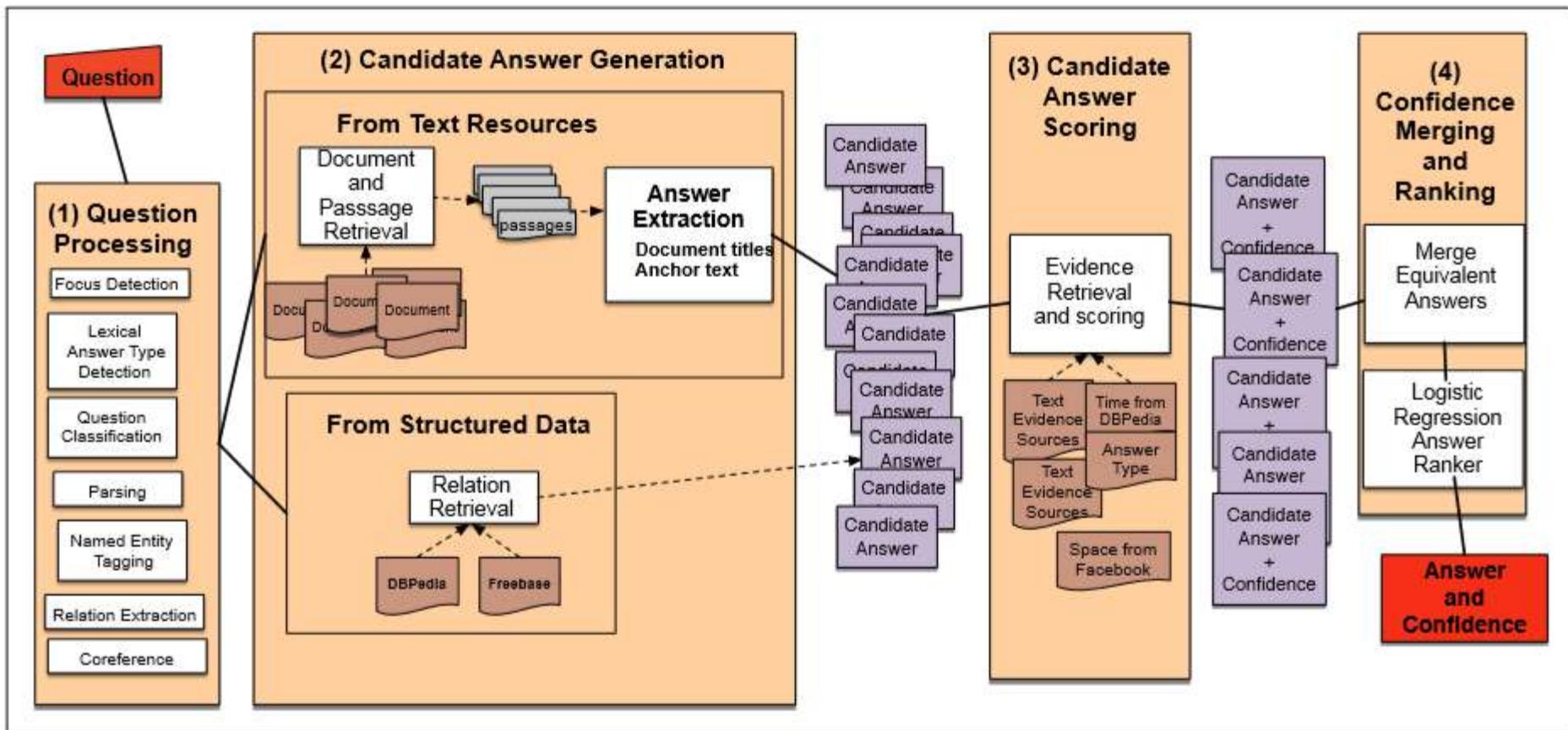


Figure 25.11 The 4 broad stages of Watson QA: (1) Question Processing, (2) Candidate Answer Generation, (3) Candidate Answer Scoring, and (4) Answer Merging and Confidence Scoring.

Deep Learning approach - QA

- **Neural QA:**

CNNs, RNNs, and attention mechanisms learn semantic matching.

- **Transformers (BERT, RoBERTa):**

Contextual embeddings enable **Extractive QA** — selecting answer spans from text.

- *Example:* SQuAD dataset benchmarks

Generative QA: Models like GPT, T5 generate free-form answers.

Extractive QA

How it Works: The model is given a **Question** and a **Context** (a piece of text). It then "highlights" the span of text that contains the answer.

Analogy: A digital highlighter.

Example:

Context: "The Eiffel Tower was completed in 1889 and is located in Paris, France."

Question: "Where is the Eiffel Tower located?"

Answer: "Paris, France"

Pros: High accuracy, factually grounded, low risk of "hallucination."

Cons: Only works if the answer is *explicitly* in the text. Can't synthesize info.

QA using Deep Learning

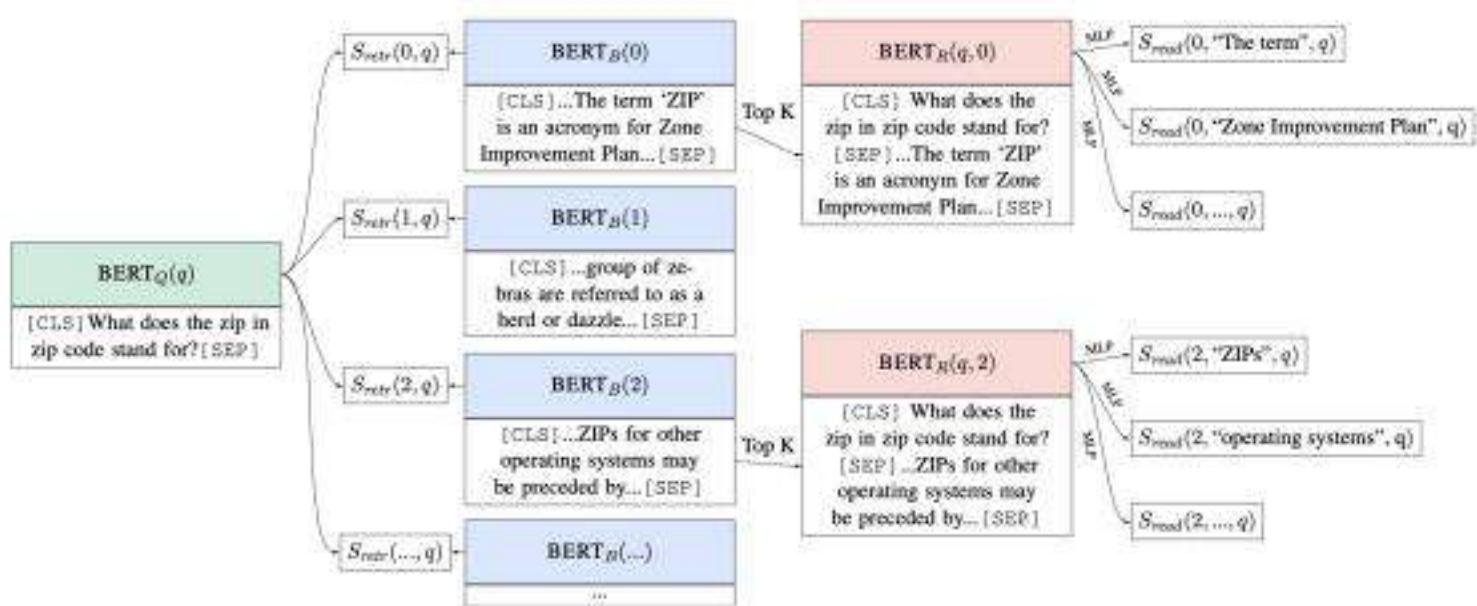


Image credit: (Lee et al., 2019)

Almost all the state-of-the-art question answering systems are built on top of end-to-end training and pre-trained language models (e.g., BERT)!

QA training datasets

Stanford question answering dataset (SQuAD)

- 100k annotated (passage, question, answer) triples

Large-scale supervised datasets are also a key ingredient for training effective neural models for reading comprehension!
- Passages are selected from English Wikipedia, usually 100~150 words.
- Questions are crowd-sourced.
- Each answer is a short segment of text (or span) in the passage.

This is a limitation— not all the questions can be answered in this way!
- SQuAD was for years the most popular reading comprehension dataset; it is “almost solved” today (though the underlying task is not,) and the state-of-the-art exceeds the estimated human performance.

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

QA training datasets

Other question answering datasets

- TriviaQA: Questions and answers by trivia enthusiasts. Independently collected web paragraphs that contain the answer and seem to discuss question, but no human verification that paragraph supports answer to question
- Natural Questions: Question drawn from frequently asked Google search questions. Answers from Wikipedia paragraphs. Answer can be substring, yes, no, or NOT_PRESENT. Verified by human annotation.
- HotpotQA. Constructed questions to be answered from the whole of Wikipedia which involve getting information from two pages to answer a multistep query:
Q: Which novel by the author of “Armada” will be adapted as a feature film by Steven Spielberg? A: *Ready Player One*

QA using Deep Learning

How can we build a model to solve SQuAD?

(We are going to use **passage**, **paragraph** and **context**, as well as **question** and **query** interchangeably)

- Problem formulation
 - Input: $C = (c_1, c_2, \dots, c_N), Q = (q_1, q_2, \dots, q_M), c_i, q_i \in V$ $N \sim 100, M \sim 15$
 - Output: $1 \leq \text{start} \leq \text{end} \leq N$ answer is a span in the passage
- A family of LSTM-based models with attention (2016–2018)

Attentive Reader (Hermann et al., 2015), Stanford Attentive Reader (Chen et al., 2016), Match-LSTM (Wang et al., 2017), BiDAF (Seo et al., 2017), Dynamic coattention network (Xiong et al., 2017), DrQA (Chen et al., 2017), R-Net (Wang et al., 2017), ReasoNet (Shen et al., 2017)..
- Fine-tuning BERT-like models for reading comprehension (2019+)

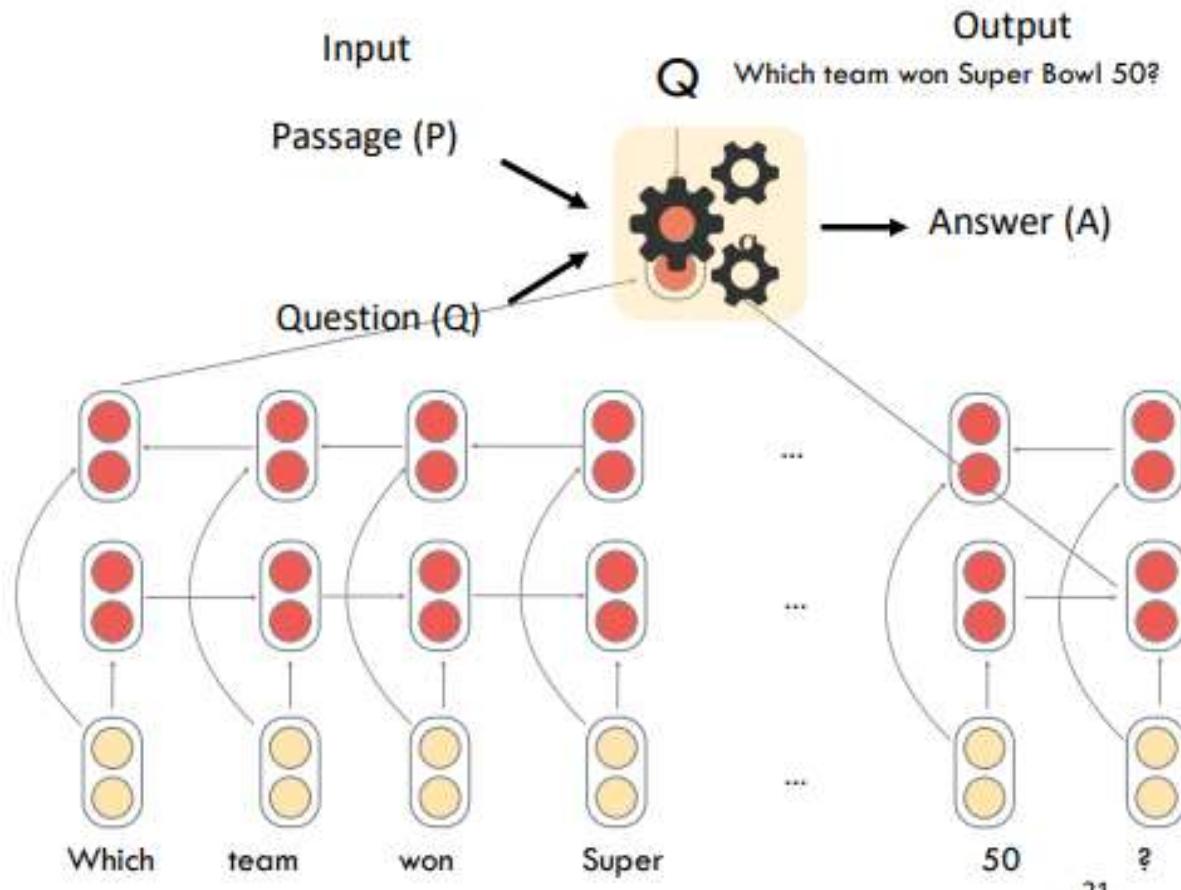
QA in Deep Learning

- Transfer learning is the application of knowledge learned while solve one problem on other similar problems.
- Latest deep learning based word embedding's such as Bidirectional Encoder Representation from Transformers (BERT) *enable pre-trained Question Answer models trained on corpus from one domain to easily answer questions from another domain.*
- This makes is easy to introduce support for Question Answering in newer applications using pre-trained models.

Word embedding of Question

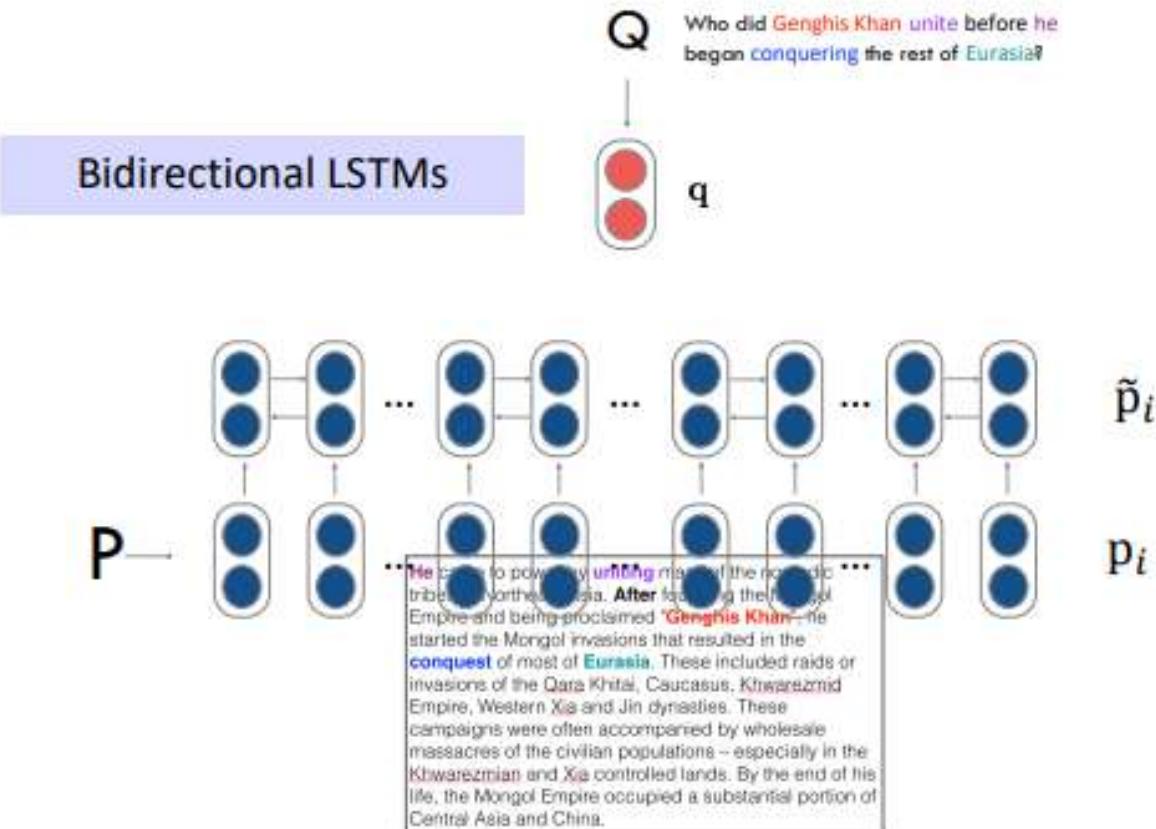
The Stanford Attentive Reader

(Use RNN in both directions - BiLSTM)



Word embedding of passage

Stanford Attentive Reader



Word embedding of passage

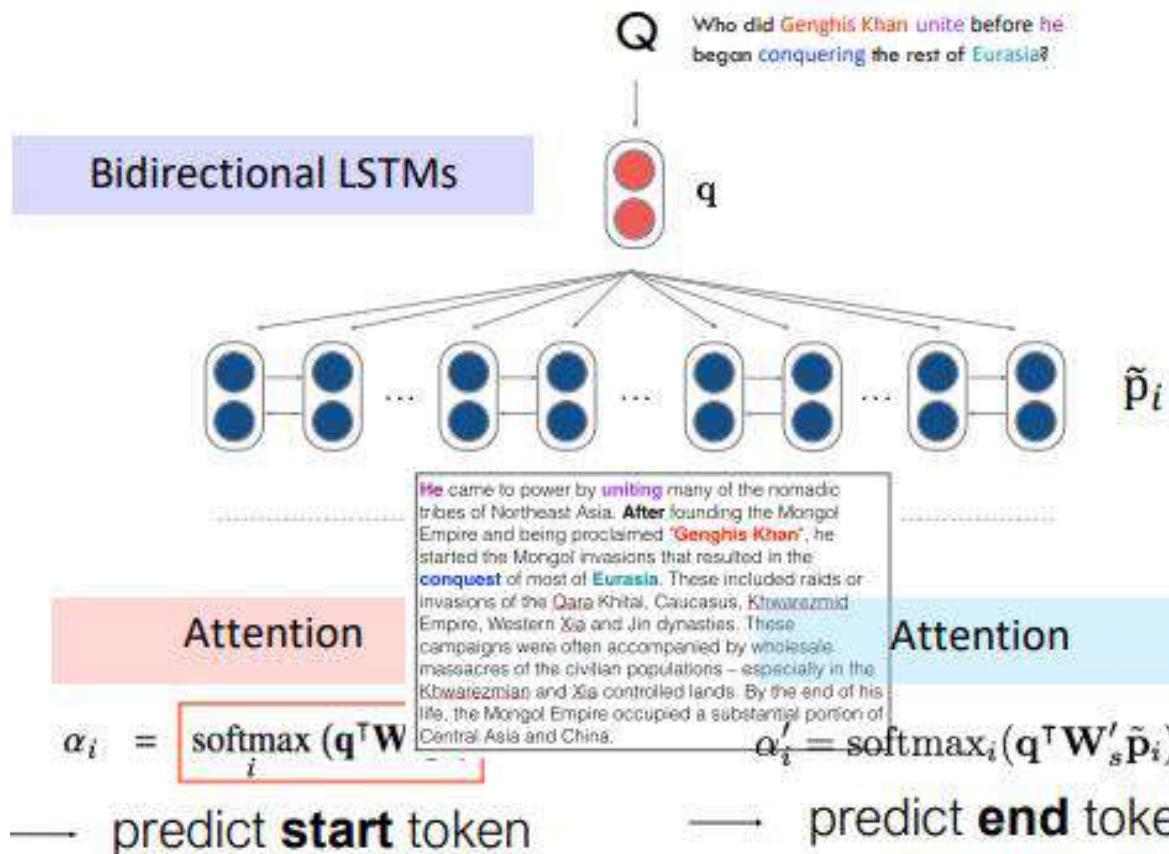
\mathbf{p}_i : Vector representation of each token in passage

Made from concatenation of

- Word embedding (GloVe 300d)
- Linguistic features: POS & NER tags, one-hot encoded
- Term frequency (unigram probability)
- Exact match: whether the word appears in the question
- 3 binary features: exact, uncased, lemma
- Aligned question embedding (“car” vs “vehicle”)

Attention Mechanism to predict start and end token of answer

Stanford Attentive Reader



Deep Learning architecture for QA

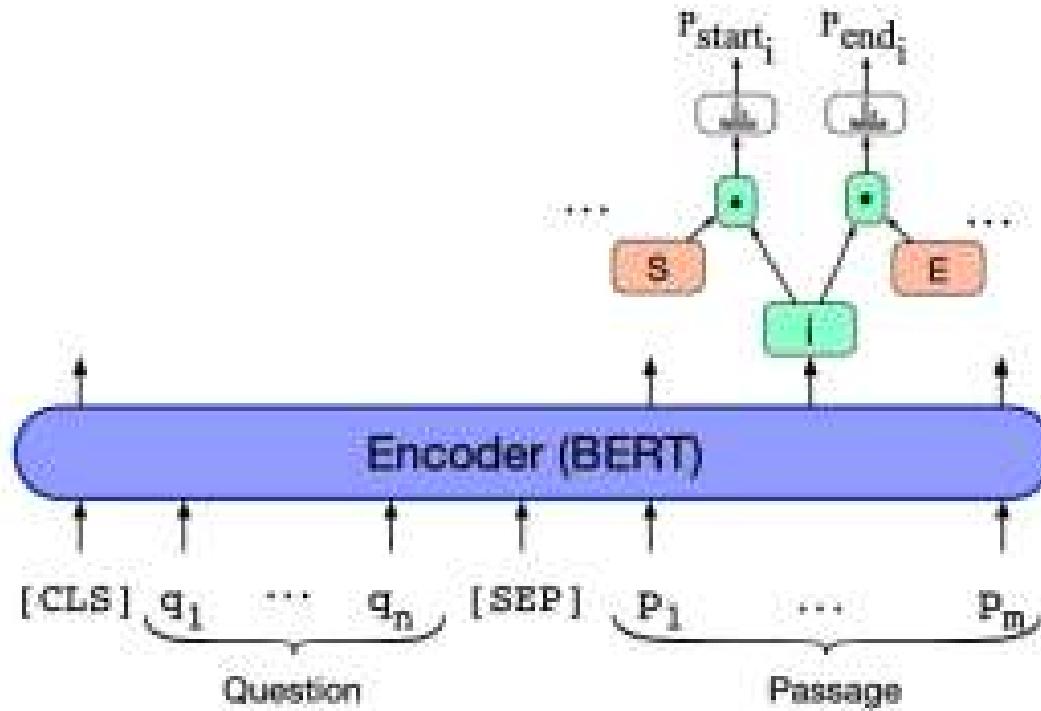
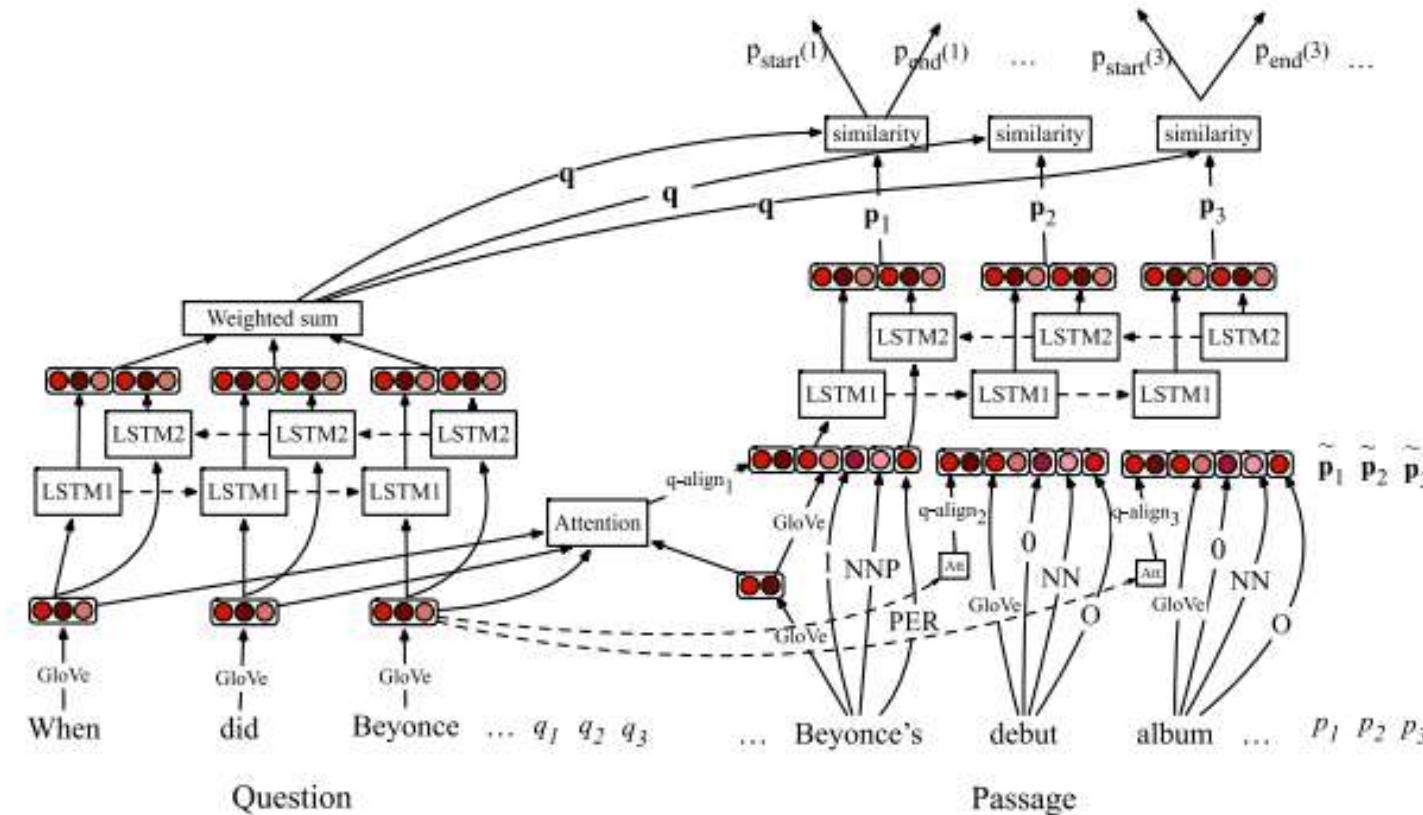


Image credit: J & M, edition 3

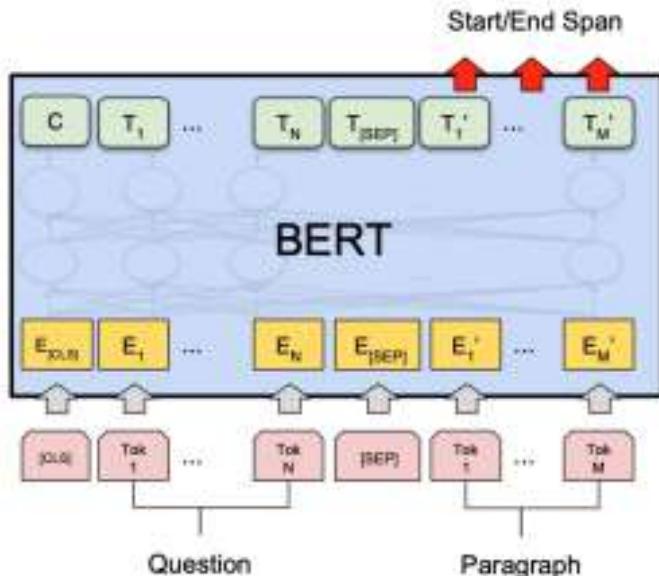
Deep Learning architecture for QA



Training objective: $\mathcal{L} = - \sum \log P^{(\text{start})}(a_{\text{start}}) - \sum \log P^{(\text{end})}(a_{\text{end}})$

Deep Learning architecture for QA

BERT for reading comprehension



$$\mathcal{L} = -\log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*)$$

$$p_{\text{start}}(i) = \text{softmax}_i(\mathbf{w}_{\text{start}}^\top \mathbf{h}_i)$$

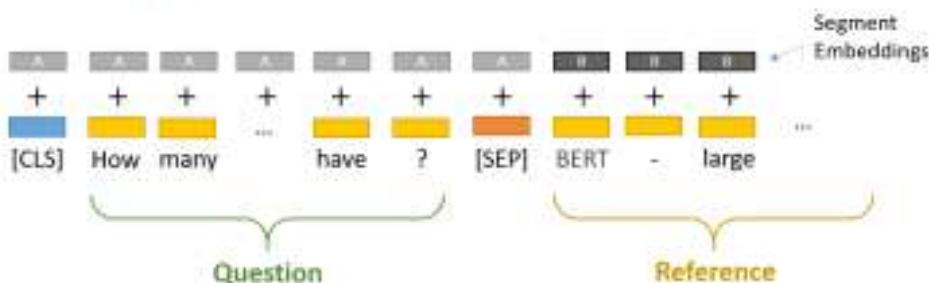
$$p_{\text{end}}(i) = \text{softmax}_i(\mathbf{w}_{\text{end}}^T \mathbf{h}_i)$$

where \mathbf{h}_l is the hidden vector of C_l , returned by BERT

Question = Segment A

Passage = Segment B

Answer = predicting two endpoints in segment B



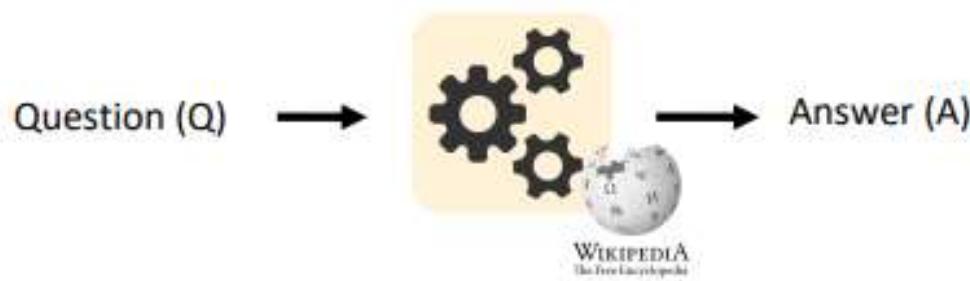
Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

Image credit: <https://mccormickml.com/>

Open Domain QA

Open-domain question answering

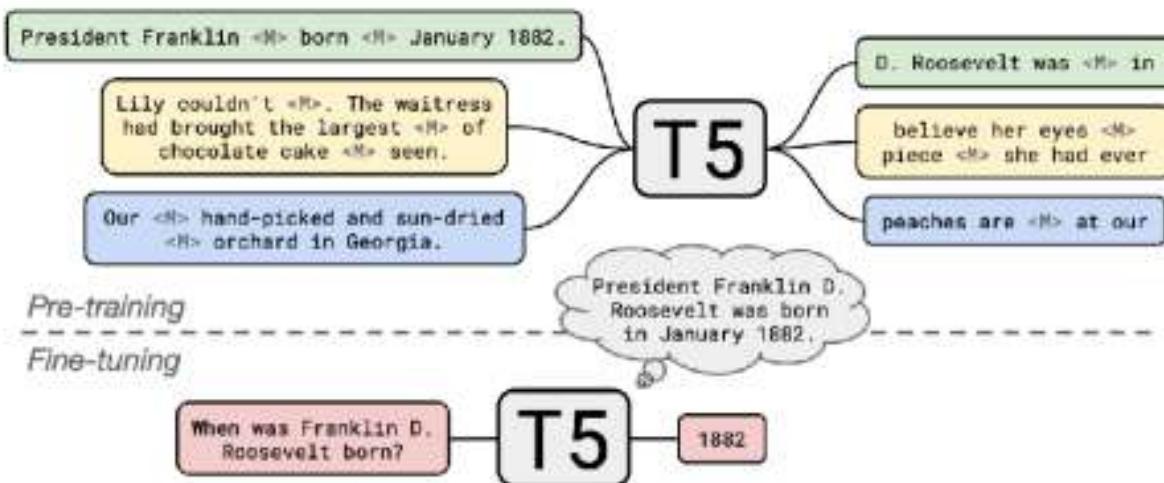


- Different from reading comprehension, we don't assume a given passage.
- Instead, we only have access to a large collection of documents (e.g., Wikipedia). We don't know where the answer is located, and the goal is to return the answer for any open-domain questions.
- Much more challenging and a more practical problem!

T5 LLM for QA

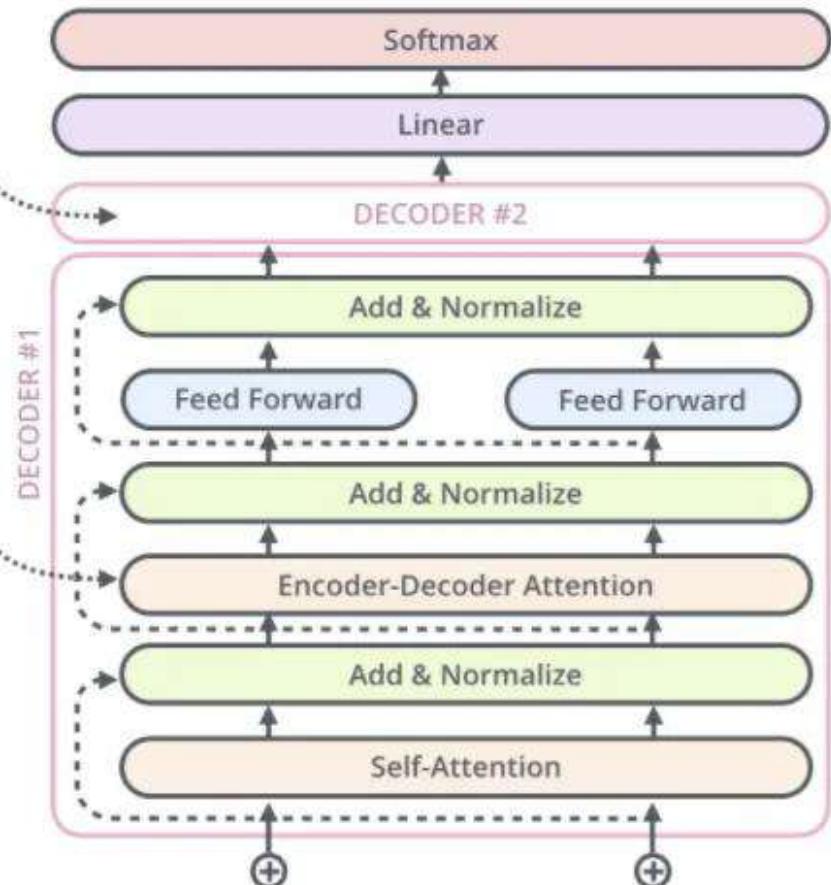
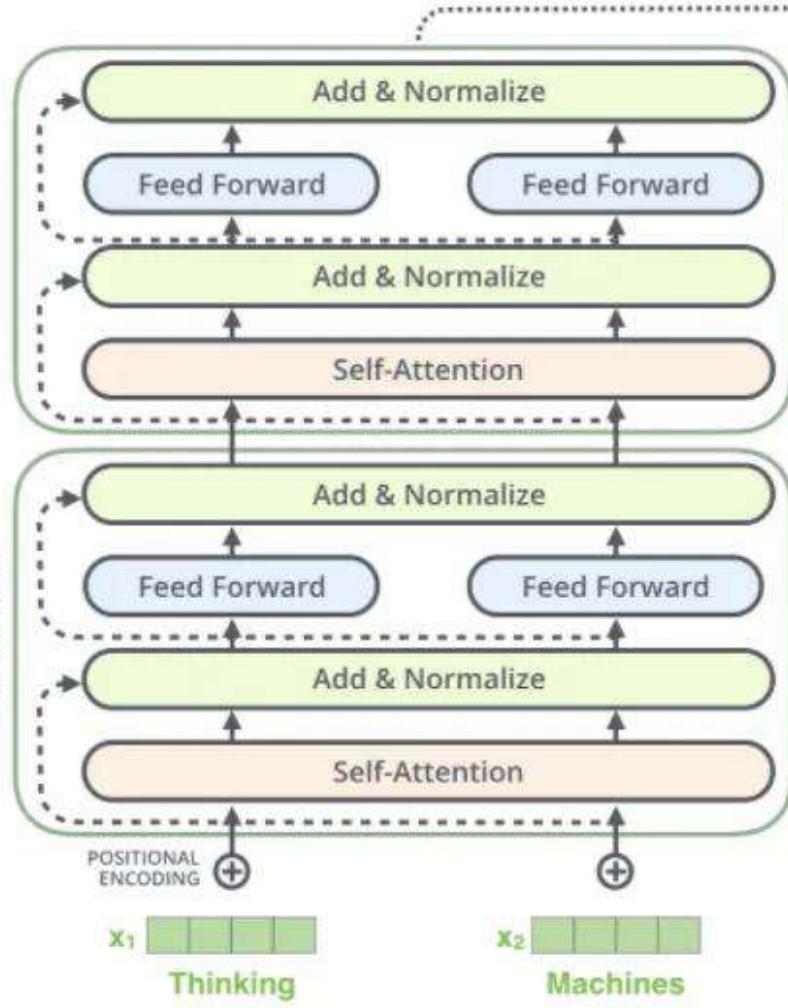
Large language models can do open-domain QA well

- ... without an explicit retriever stage



T5 model Architecture

ENCODER #2



Text-to-Text Transfer Transformer (T5)

- The architecture of T5 model is almost the same as the BERT.
- Both the encoder and decoder consist of 12 blocks.
- This model has 220 million parameters.
- Only a few changes have been made to the architecture like they have removed the Layer norm bias and placed the layer normalization outside the residual path.
- There is a different position embedding scheme used in T5.

Challenges of Deep learning approaches



Every manager is an employee.
Rose is a manager.
Rose is an employee?

Reasoning

Google was founded by Larry Page. Sergey Brin was a co-founder.
Who founded Google?

Answer spanning multiple parts of the document

Who are the founders of Google and Facebook?

Multiple Questions

Challenges of Information Retrieval Based Systems Developed Using Deep Learning Based Question Answer models

Challenges of Deep learning approaches



- Incapable of answering questions that require reasoning.
- Deep learning based models for Question Answering take the input passage and questions and output the start and end position in the passage that contains the answer.
- Consequently they can not answer questions whose answer is spread across the document.
- Can not answer questions that have multiple sub questions whose answers are spread across the document.

QA Modern Approaches

- **Generative QA:** Generates a new answer from scratch.
 - **Retrieval-Augmented Generation (RAG):** The modern hybrid that gets the best of both worlds.
-

Generative QA

How it Works: The model (a Large Language Model like GPT-4) uses its vast, pre-trained knowledge to generate a new, human-like answer.

Analogy: A subject-matter expert answering from memory.

Example:

- **Question:** "Why is the Eiffel Tower significant?"
- **Answer:** "The Eiffel Tower is significant because it was an architectural marvel for its time and has become an iconic symbol of Paris and French culture."

Pros: Very flexible, can answer "why" and "how" questions, highly conversational.

Cons: "**Hallucinations**" (making up facts) and **stale data** (its knowledge is frozen in time).

The Big Problem: Hallucinations & Stale Data

You can't trust a standard Generative LLM with your private company data.

- **It's a "Black Box":** You don't know *why* it gave an answer (no citations).
- **It's Stale:** It has no idea your new product (Project X) launched last week.
- **It's Insecure:** It wasn't trained on your *private* documents.

This leads to the most important approach used in business today...

Retrieval Based Methods

- Modern large language models (LLMs) — like ChatGPT, Gemini, or Claude — use **retrieval-based methods** behind the scenes.
- Instead of generating answers purely from memory, they:
- **Retrieve** relevant text chunks from a large document database (IR stage).
- **Generate** the final answer using a language model conditioned on those documents.
- This is called **Retrieval-Augmented Generation (RAG)** or **retrieval-augmented QA**.

Retrieval Based Methods

- Example:
- User: “Who discovered penicillin?”
- System: retrieves passages mentioning “Alexander Fleming” → summarizes →
Answer: “Penicillin was discovered by Alexander Fleming in 1928.”
- So — modern IR-based QA = **search + LLM reasoning.**

Retrieval-Augmented Generation

The "Best of Both Worlds" Solution.

- RAG combines a **Retriever** (like a search engine) with a **Generator** (an LLM).
- **Analogy:** An "open-book" exam. The LLM doesn't answer from memory. It is *given* the relevant documents (the "book") and told, "Use *only* these documents to answer the question."
- This solves the hallucination and stale data problem.

RAG Pipeline

- **Query:** User asks, "What is our company's WFH policy?"
- **Retrieve:** The system *first* searches a private database (a **Vector Database**) of all company HR documents. It finds the 3 most relevant docs.
- **Augment:** The system builds a new prompt:
 - "User Question: What is our WFH policy?"
 - "Context: [Here is the text of the 'Work-From-Home Policy' doc...]"
- **Generate:** This prompt is fed to the LLM. The LLM then generates an answer *based only on the provided context*.
 - **Answer:** "Our WFH policy allows employees to work from home 2 days per week..."



Knowledge-based QA (Knowledge Graphs / Semantic QA)

- Google Knowledge Graph answers like:
- “Who is the president of France?”
→ directly shows *Emmanuel Macron* from its structured knowledge base.
- Enterprise QA systems use **ontologies or databases** to give **precise factual answers**.
- Knowledge-based QA is especially useful when:
- Accuracy and interpretability matter (e.g., finance, medicine).
- The data is structured and up to date.
- LLMs can even **query** these knowledge sources dynamically (using tools or APIs).

Hybrid QA

Component	Role
IR-based retriever	Finds relevant unstructured text.
Knowledge-based backend	Provides verified facts from structured sources.
LLM / neural model	Reads retrieved info, reasons, and generates a fluent answer.

RAG Tech stack

The Brains (Models): LLMs

The Memory (Data): Vector Databases

The Glue (Orchestration): Frameworks

The Interface (App): Frontend/Backend

RAG Tech stack

Large Language Models (LLMs): These generate the final answer.

- **Closed Source:** OpenAI (GPT-4, GPT-4o), Anthropic (Claude 3), Google (Gemini).
- **Open Source:** Meta (Llama 3), Mistral (Mixtral).

Embedding Models: These are crucial for the "Retrieval" step. They turn text documents into numbers (vectors) so we can find *semantically similar* documents.

Examples: all-MiniLM-L6-v2, text-embedding-3-small

Vector Datastore

You can't find "semantically similar" documents in a traditional SQL database.

Vector Datastore are purpose-built to store and query these "embedding" vectors at massive scale.

How it works: It finds documents that are *conceptually related* to the query, not just ones that share keywords.

Query: "rules for leave"

Finds: "policy for taking time off"

Examples: Pinecone, ChromaDB, FAISS, Weaviate, Milvus

RAG Orchestration

These frameworks manage the entire RAG pipeline (Query -> Retrieve -> Augment -> Generate). They are the "plumbing" that connects everything.

LangChain: The most popular and comprehensive library. A "Swiss Army Knife" for building complex LLM applications (agents, chains).

LlamaIndex: A simpler, data-focused framework that excels at *one thing*: building powerful RAG pipelines. It's built for "ingesting, indexing, and querying" your data.

Comparison of QA approaches

Feature	Extractive QA	Generative QA (LLM)	RAG
Answer Source	Exact text span	Model's internal memory	External, current documents
Hallucination Risk	Very Low	High	Low (mitigated)
Data Freshness	Depends on context	Stale (frozen in time)	Real-time (just update docs)
Citations?	Yes (the source doc)	No	Yes (can cite sources)
Best For	Fact-checking	Creative chat	Enterprise QA, Chatbots

Case Study: "AskHR"

- An Internal Enterprise Chatbot

- Company: A 10,000-person global tech firm.
- **The Problem:** The HR department is overwhelmed with thousands of repetitive employee questions every month:
 - "How much PTO do I have?"
 - "What's the policy for travel expenses?"
 - "Where do I find my tax forms?"
- These answers exist but are buried in a 500-page PDF on an internal site
- **Goal:** Build a 24/7 chatbot that can answer 90% of HR questions instantly and accurately, citing its sources.
- **The Choice: RAG**
- **Why not Generative?** A base LLM (like ChatGPT) knows *nothing* about the company's specific policies and would make up answers (hallucinate).
- **Why not Extractive?** The policies are complex and require synthesis, not just "highlighting" a single sentence

Techstack for AskHR

Data Ingestion (Offline):

- All 200+ HR documents (PDFs, Word docs, site pages) are chunked.
- An **Embedding Model** converts these chunks into vectors.
- All vectors are loaded into a **Vector Database (Pinecone)**.

QA Application (Real-time):

- An employee asks, "Can I use my corporate card for conference tickets?"
- **LangChain** (the "glue") takes the query.
- It queries **Pinecone** to find the top 3 relevant doc chunks (e.g., the "Travel Policy" and "Corporate Card" docs).
- It passes the query + chunks to the **LLM (Claude 3)**.

The LLM generates the answer: "Yes, you can use your corporate card for conference tickets, but you must get manager approval first (as per the Travel Policy, pg. 4)."

Common Evaluation Metrics

1. *Accuracy* (does answer match gold-labeled answer?)

2. *Mean Reciprocal Rank*

- For each query return a ranked list of M candidate answers.
- Query score is 1/Rank of the first correct answer
 - *If first answer is correct: 1*
 - *else if second answer is correct: ½*
 - *else if third answer is correct: ⅓, etc.*
 - *Score is 0 if none of the M answers are correct*
- Take the mean over all N queries

$$MRR = \frac{\sum_{i=1}^N \frac{1}{rank_i}}{N}$$

Common Evaluation Metrics

QA Type	Key Metrics
Extractive	EM, F1, Precision, Recall
Generative/Abstractive	ROUGE, BERTScore, BLEURT, Human Eval
Multiple Choice	Accuracy, Macro/Micro Accuracy
KG-based QA	Hits@k, MRR, Precision@k, Recall@k
Modern/LLM QA	Faithfulness, Explainability, Robustness

Summary

- NLP QA has moved from simple "extraction" to sophisticated "generation."
- **Retrieval-Augmented Generation (RAG)** is the dominant, business-ready approach for safe, accurate, and current-data-driven QA.
- The modern tech stack (**LLMs + Vector DBs + LangChain**) has made it possible for any company to build its own powerful, private QA system.

Research Papers

KERAG: Knowledge-Enhanced RAG	Integrates a broad subgraph from a knowledge graph + LLM chain-of-thought to improve answer coverage and reduce hallucination.	ACL Anthology
BYOKG-RAG: Multi-Strategy Graph Retrieval for KGQA	Combines LLM-generated artifacts with graph tools to iteratively retrieve from custom KGs, improving robustness across varied graphs.	ACL Anthology

Research Papers

D-RAG: Differentiable RAG for KGQA	Jointly optimizes retriever and generator via differentiable subgraph sampling to boost performance on KGQA.	ACL Antholog Y
Enhancing Complex Reasoning in KGQA (Aqua-QA)	Approximates query graphs from natural language to handle incomplete KGs and logical reasoning.	ACL Antholog Y
RGR-KBQA: Generating Logical Forms with KG-Enhanced LLMs	Uses a retrieve-generate-retrieve pipeline to improve logical form generation and reduce hallucination in KBQA.	ACL Antholog Y

Research Papers

Knowledge Graph-Extended RAG (KG-RAG)	Integrates LLMs with KGs using in-context learning + chain-of-thought for explainable multi-hop QA.	EmergentMind
LLMs + Knowledge Graphs for QA: Survey	Categorizes and analyzes methods of combining LLMs + KGs for QA, highlighting future directions.	ACL Anthology
MDKAG: Multimodal KG for Educational QA	Builds a QA system using a multimodal discipline-specific knowledge graph for educational questions.	MDPI
RAGulating Compliance: Multi-Agent KG QA	Uses a multi-agent system and a dynamically maintained KG to answer regulatory compliance questions with traceability.	arXiv

Research Papers

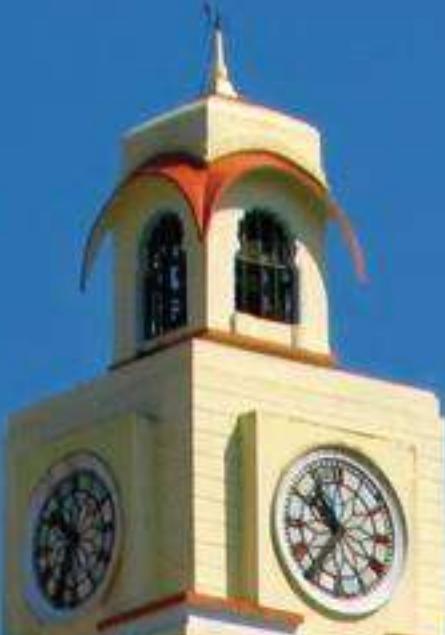
StepChain GraphRAG: Multi-Hop QA	Decomposes questions and builds evidence chains via BFS traversal on dynamically constructed subgraphs for interpretable multi-hop QA.	arXiv
Interpretable QA with Knowledge Graphs	Answers via KG retrieval without LLMs, using paraphrasing of KG edge relations for human-readable output.	arXiv
AutoGraph-R1: RL for KG Construction for QA	Uses reinforcement learning to build KGs optimized for downstream QA performance.	arXiv

QA approaches and applications

QA Type	Source of Answer	Approach	Example Application
Extractive	Given passage	Span selection	Reading comprehension, FAQs
Generative	Given + multiple passages	Text generation	Chatbots, long-form answers
Open-Domain	Large corpus / web	Retrieval + generation	Search engines, real-time info QA
Knowledge-Base	Structured KB	Query / reasoning	Enterprise KBQA, Wikidata QA

References

- Speech and Language processing: An introduction to Natural Language Processing, Computational Linguistics and speech Recognition by Daniel Jurafsky and James H. Martin[3rd edition] - Chapter 25
- <https://medium.com/@akashgoel12/question-answering-system-on-corona-approach-01-6ef9799695cb>
- <https://www.machinelearningplus.com/nlp/chatbot-with-rasa-and-spacy/>
- <https://analyticsindiamag.com/10-question-answering-datasets-to-build-robust-chatbot-systems/>
- <https://github.com/ElizaLo/Question-Answering-based-on-SQuAD>
- <https://intersog.com/blog/the-basics-of-qa-systems-from-a-single-function-to-a-pre-trained-nlp-model-using-python/>
- <https://paperswithcode.com/task/question-answering/latest>
- https://www.youtube.com/watch?v=NcqfHa0_YmU
- https://www.youtube.com/watch?v=3XiJrn_8F9Q



Natural Language Processing Applications



BITS Pilani
Pilani Campus

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



Session 5: Dialogue Systems and Chatbots

These slides are prepared by the instructor, with grateful acknowledgement of Prof. Dan Jurafsky and many others who made their course materials freely available online.

Session Content

- Properties of Human Conversation
 - Chabot's
 - GUS: Simple Frame-based Dialogue Systems
 - The Dialogue-State Architecture
 - Evaluating Dialogue Systems
 - Dialogue System Design
-

AI Assistants

Notification Assistant



Hi there - just a friendly reminder that your insurance policy expires in a month. Make sure to renew it in our member portal.

FAQ Assistant



I need to renew my renters insurance. How much will it be?



You can calculate your renewal price on our website here:
xyz.com/renew

Contextual Assistant



I need to renew my renters insurance. How much will it be?



I'd be happy to check for you. Firstly, are you still living in the same apartment?

Yes



Great - so just confirming it's 980 sq ft?

Yes



Thanks! Your new rate from September 1st onwards would be \$10 / month.



Would you like me to renew your policy for you right now?

Sure



Great. I've sent you a confirmation to your email.

AI Assistants

Personalized Assistant

- Assistant knows you much more in detail
- Quickly checks a few final things before giving you a quote tailored to your actual situation.



I can see your details are almost the same, except now you might want coverage for your new laptop. Additional coverage is only \$4 a month more for full coverage.
Sound ok?

Sounds good!



Autonomous Organization of Assistants

- Group of AI assistants that know every customer personally
- Eventually run large parts of company operations—from lead generation over marketing, sales, HR, or finance



Conversational AI Approaches and real world applications



Approach	Short Summary	Real-World Applications
Rule-Based	Fixed rules, decision trees, predictable responses	<ul style="list-style-type: none">IVR phone menusBanking/telecom FAQ botsHR onboarding flowsSimple customer support chat
IR-Based	Retrieves best match from a knowledge base using keywords or vector search	<ul style="list-style-type: none">Enterprise FAQ searchE-commerce supportAirline & telecom chatbotsGovernment info portals
Deep Learning (Pre-LLM)	Seq2Seq / early Transformer models that generate basic responses	<ul style="list-style-type: none">Smart reply suggestionsDomain-specific assistantsSimple virtual assistants
LLM-Based	Large pretrained models with reasoning, context understanding, and generative ability	<ul style="list-style-type: none">Advanced chatbotsCoding assistantsLegal/medical Q&A (supervised)Enterprise RAG systems
Agentic AI	LLM + tools + memory → autonomous task execution and multi-step planning	<ul style="list-style-type: none">Travel booking agentsFinancial analysis botsCustomer service copilotsDevOps/coding agentsWorkflow automation

QA versus conversational AI

Aspect

Question Answering (QA)

Conversational AI

Definition

A system that answers specific user questions.

A system that engages in a multi-turn dialogue with users.

Scope

Narrow: Answering questions (often factual).

Broad: Dialogue, intent detection, emotion, context, etc.

Example Task

“What is the capital of France?” → “Paris”

“I’m planning a trip.” →
“Great! Where are you going?”



Conversational Agents

AKA Dialog Agents

- Phone-based Personal Assistants
 - SIRI (Apple), Alexa (Amazon),
 - Cortana (Microsoft), Google Assistant
- Talking to your car, Pay bills...
- Sales, Marketing, Insurance....
- Clinical uses for mental health, Nurses, Dr. Bot
- Lawyer bots
- Chatting for fun, Mr. FriendBot

Context and Conversation in Virtual Assistants like Siri

- Coreference helps resolve ambiguities
 - U: “Book a table at Il Fornaio at 7:00 with **my mom**”
 - U: “Also send **her** an email reminder”
- Clarification questions:
 - U: “Chicago pizza”
 - S: “Did you mean pizza restaurants in Chicago or Chicago--style pizza?”

Conversation Characteristics

Turns

- dialogue is a sequence of turns
- C1: I need to travel in May.
A1: And, what day in May did you want to travel?
C2: OK uh I need to be there for a meeting that's from the 12th to the 15th.
A2: And you're flying into what city? C3: Seattle.

Speech Acts of Dialog Acts:

- Constitutives: answering, claiming, confirming, denying, disagreeing, stating)
- Directives: to do something (advising, asking, forbidding, inviting, ordering, requesting)
- Commissives: committing the speaker to some future course of action (promising, planning, vowing, betting, opposing)
- Acknowledgments: apologizing, greeting, thanking, accepting an acknowledgment

Challenges of Dialog Systems

Grounding:

- acknowledging that the hearer has understood the speaker

Dialog Structure:

- QUESTIONS set up an expectation for an ANSWER. PROPOSALS are followed by ACCEPTANCE (or REJECTION). COMPLIMENTS (“Nice jacket!”) often give rise to adjacency pair DOWNPLAYERS (“Oh, this old thing?”). These pairs, called adjacency pairs are composed of a first pair part and a second pair part

Clarification/correction question

- Which can form a sub dialogue between a REQUEST and a RESPONSE
- Ex: User: What do you have going to UNKNOWN WORD on the 5th?
System: Let's see, going where on the 5th?

Inference:

- A1: And, what day in May did you want to travel? C2: OK uh I need to be there for a meeting that's from the 12th to the 15th. Notice that the client does not in fact answer the agent's question. The client merely mentions a meeting at a certain time

Conversational AI journey

ELIZA (1960s): pattern matching.

1990s–2000s: statistical dialogue models.

2010s: deep learning & seq2seq.

2020+: transformers & LLMs.

Two classes of systems

1. Chatbots
2. (Goal-based) Task oriented
Dialog agents
 - *SIRI, interfaces to cars, robots,*
 - *booking flights or restaurants*

The word "chatbots" is sometimes used in the popular press for both. We'll use it only for #1.

Chatbot Architectures

Rule-based

1. Pattern-action rules (Eliza)
+ a mental model (Parry)

Corpus-based (from large chat corpus)

2. Information Retrieval
3. Neural network encoder-decoder

Chatbots

- Sometimes for fun
- But originally designed as a testbed for clinical therapy
(And still used that way!)

Eliza: Weizenbaum (1966)

Dialog in Green are by Eliza system.

Men are all alike.

IN WHAT WAY

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.

I AM SORRY TO HEAR YOU ARE DEPRESSED

...

WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?

My father

YOUR FATHER

You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK I AM NOT AGGRESSIVE

You don't argue with me.

WHY DO YOU THINK I DON'T ARGUE WITH YOU

You are afraid of me.

DOES IT PLEASE YOU TO BELIEVE I'M AFRAID OF YOU

My father is afraid of everybody.

WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER

Bullies.

DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE

Each sentence is mapped into a conceptualization

- A predication on a conceptual object
- A predication on a relation between two objects
- A predication on an attribute:

What is your work?

What sort of work do you do?

Where do you work?

→ (your work?)

What do you do for a living?

What is your job?

Do you have a job?

What is your occupation

- Complex Pattern/transform rules

- Different predicates (fear, afraid of)
- Ordering (You are afraid of me = I frighten you)

Woebot

- Chatbot for delivering Cognitive Behavior Therapy via brief daily conversations
 - Little decision tree of language and menu responses

"What's going on in your world right now?"

"How are you feeling?"

In response to endorsed loneliness:

"I'm so sorry you're feeling lonely. I guess we all feel a little lonely sometimes"

- In a 2-week experiment seemed to reduce depression, but not anxiety

Fitzpatrick, Darcy, Vierhile, 2017. Delivering Cognitive Behavior Therapy to Young Adults with Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial. JMIR Ment Health 4:2.

Simple Chatbot – Rule based Demo

- NLTK has a module, nltk.chat, which simplifies building Chatbots
- **Chat**: This is a **class** that has all the logic that is used by the chatbot.
- **Reflections**: This is a **dictionary** that contains a set of **input values and its corresponding output values**. It is an optional dictionary that you can use. You can also create your own dictionary in the same format as below and use it in your code.

IR-based chatbots

- Idea: Mine conversations of human chats or human-machine chats
 - Microblogs: Twitter or Weibo
 - Movie dialogs
- Cleverbot (Carpenter 2017
<http://www.cleverbot.com>)
- Microsoft Tay

A Cleverbot conversation

User:	Do you like Doctor Who?
Cleverbot:	Yes, so funny
User:	What's your favorite show?
Cleverbot:	I don't have one
User:	What's your name?
Cleverbot:	Jason I already told you.
User:	No your real name.
Cleverbot:	Stephanie IS my real name.
User:	I thought it was Jason?
Cleverbot:	How did you come to that conclusion.
User:	Because you said so.
Cleverbot:	I have to do my homework.

Two IR-based chatbot architectures

1. Return the response to the most similar turn

- Take user's turn (q) and find a (tf-idf) similar turn t in the corpus C
 $q = \text{"do you like Doctor Who"}$

$$t' = \text{"do you like Doctor Strangelove"}$$

- Grab whatever the response was to t .

$$r = \text{response} \left(\operatorname{argmax}_{t \in C} \frac{q^T t}{\|q\| \|t\|} \right)$$

Yes, so funny

2. Return the most similar turn

$$r = \operatorname{argmax}_{t \in C} \frac{q^T t}{\|q\| \|t\|}$$

Do you like Doctor Strangelove

IR-based models of chatbots

- Also fine to use other features like user features, or prior turns
- Or non-dialogue text
 - COBOT chatbot (Isbell et al., 2000)
 - sentences from the Unabomber Manifesto by Theodore Kaczynski, articles on alien abduction, the scripts of “The Big Lebowski” and “Planet of the Apes”.
 - Wikipedia text

IR based Chatbot Demo

- Web scraping using Python
- Retrieve Web page on Chatbots
- Bot answers questions related to the Web page content.

Neural chatbots

- Think of response generation as a task of *transducing* from the user's prior turn to the system's turn.
- Train on:
 - movie dialogue databases
 - Twitter conversations
- Train a deep neural network
 - map from user1 turn to user2 response

Sample neural chatbot output

Input	tell me ... how long have you had this falling sickness ?
Vanilla-SEQ2SEQ	i'm not a doctor.
Adversarial	a few months, i guess .
Input	so i had the doctors test sammy 's response to conditioning .
Vanilla-SEQ2SEQ	sammy wrote the test sammy wrote the test .
Adversarial	so he took the pills .
Input	they didn 't have much success with this problem commander .
Vanilla-SEQ2SEQ	they 're not the only ones who have been in the system .
Adversarial	can we find someone else ?

Adversarial neural network use generative models

Generate new data with the same statistics as the training set. For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic

Oh I've never seen that! How long does it take you
? guys to learn the drill?
Like 2 weeks ago!!

Chatbots: pro and con

Pro:

- Fun
- Applications to counseling
- Good for narrow, scriptable applications

Cons:

- They don't really understand
- Rule-based chatbots are precise but expensive
- IR-based or neural networks chatbots can only mirror training data
 - (Garbage-in, Garbage-out)

The future: combining chatbots with frame-based agents

Frame-based dialog agents

- Sometimes called "task-based dialog agents"
- Based on a "domain ontology"
 - A knowledge structure representing user intentions
- One or more **frames**
 - Each a collection of **slots**
 - Each slot having a **value**

The Frame

- A set of **slots**, to be filled with information of a given **type**
- Each associated with a **question** to the user

Slot	Type	Question
ORIGIN	city	What city are you leaving from?
DEST	city	Where are you going?
DEP DATE	date	What day would you like to leave?
DEP TIME	time	What time would you like to leave?
AIRLINE	line	What is your preferred airline?

Slot types can be complex

The type *DATE*

DATE

MONTH NAME

DAY (BOUNDED-INTEGER 1 31)

YEAR INTEGER

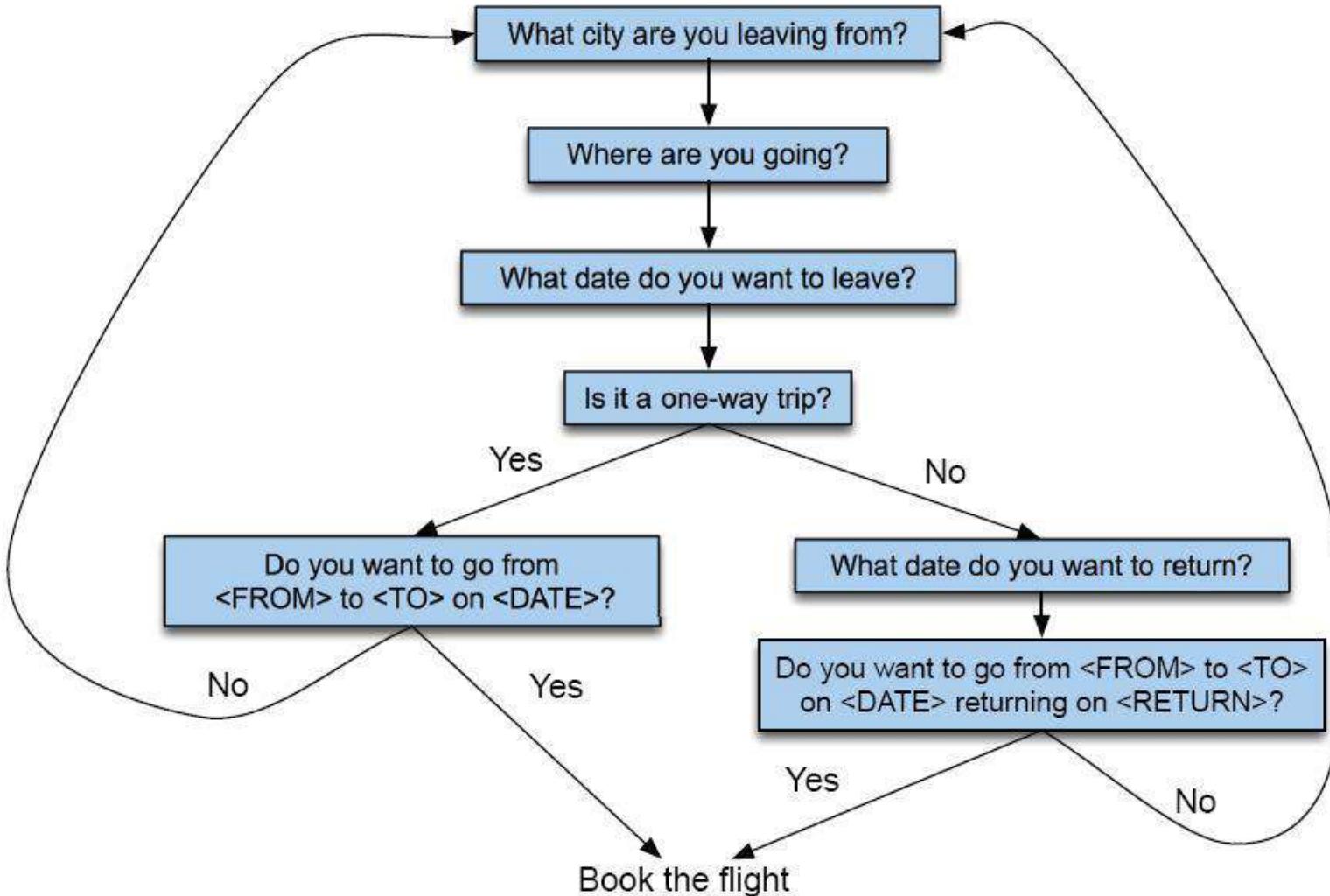
WEEKDAY (MEMBER (SUNDAY MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY)]

Control structure for frame-based dialog

Consider a trivial airline travel system:

- Ask the user for a departure city
- Ask for a destination city
- Ask for a time
- Ask whether the trip is round-trip or not

Finite State Dialog Manager



Finite-state dialog managers

- System completely controls the conversation with the user.
- It asks the user a series of questions
- Ignoring (or misinterpreting) anything the user says that is not a direct answer to the system's questions

Dialogue Initiative

- Systems that control conversation like this are called **single initiative**.
- **Initiative**: who has control of conversation
- In normal human-human dialogue, initiative shifts back and forth between participants.

System Initiative

System completely controls the conversation

- Simple to build
 - User always knows what they can say next
 - System always knows what user can say next
 - Known words: Better performance from ASR
 - Known topic: Better performance from NLU
 - OK for VERY simple tasks (entering a credit card, or login name and password)
-
- - Too limited



Problems with System Initiative

- Real dialogue involves give and take!
- In travel planning, users might want to say something that is not the direct answer to the question.
- For example answering more than one question in a sentence:

Hi, I'd like to fly from Seattle Tuesday morning

I want a flight from Milwaukee to Orlando one way leaving after 5 p.m. on Wednesday.

Single initiative + universals

- We can give users a little more flexibility by adding **universals**: commands you can say anywhere
 - As if we augmented every state of FSA with these
 - **Help**
 - **Start over**
 - **Correct**
 - This describes many implemented systems
 - But still doesn't allow user much flexibility
-

Instead, the GUS architecture

- GUS
 - Genial(friendly) Understannder System
 - A kind of ***mixed initiative***
 - The conversational initiative shifts between system and user
 - The structure of the **frame** guides dialogue

Frames are mixed-initiative

- System asks questions of user, filling any slots that user specifies
 - When frame is filled, do database query
- If user answers 3 questions at once, system can fill 3 slots and not ask these questions again!

Frame-based dialogue agents

1977:

versation with a client who wants to make a simple return trip to a single city in California.

There is good reason for restricting the domain of discourse for a computer system which is to engage in an English dialog. Specializing the subject matter that the system can talk about permits it to achieve some measure of realism without encompassing all the possibilities of human knowledge or of the English language. It also provides the user with specific motivation for participating in the conversation, thus narrowing the range of expectations that GUS must have about the user's purposes. A system restricted in this way will be more able to guide the conversation within the boundaries of its competence.

<https://nlp.stanford.edu/acvogel/gus.pdf>

Artificial Intelligence Journal, 1977

Still the industrial state of the art

- Many task-based bots use GUS architecture

The state of the art in 1977 !!!!

GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you want to go ?

Client: I want to go to San Diego on May 28

GUS: What time do you want to leave?

Client: I must be in San Diego before 10 am

GUS: Would you like P.S.A. flight 102 that arrives at 9.15 am ?

Client: What is the next flight ?

GUS: Air California flight 310 that leaves at 8.30am and arrives at 10.00am

Client: I'll take the first one

GUS: In what name should I make a reservation, please?

Client: Dan Bobrow

GUS: I have confirmed the following flight: P.S.A. flight 102 on Wednesday May 28 from San Jose to San Diego that leaves at 7.30 am and arrives at 9.15 am.

GUS: What date do you want to return on ?

Client: On Friday in the evening.

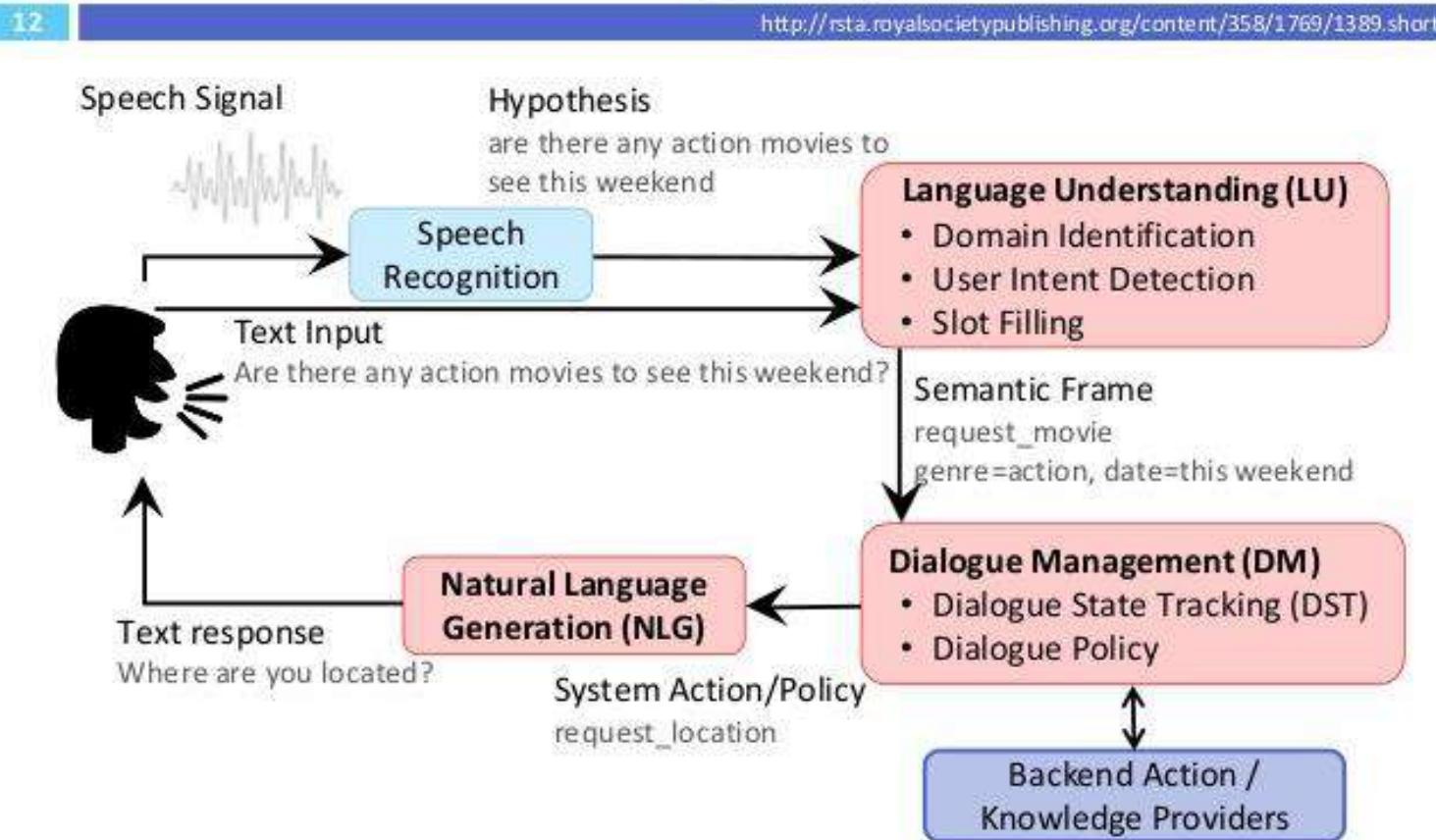
GUS: Would you like the flight that leaves at 7.45 pm ?

Client: That's fine.

GUS: I have confirmed the following flight: P.S.A. flight 307 on Friday May 30 from San Diego to San Jose that leaves at 7.45 pm and arrives at 9.30 pm Thank you for calling. Goodbye

Components of Conversational AI systems

Task-Oriented Dialogue System (Young, 2000)

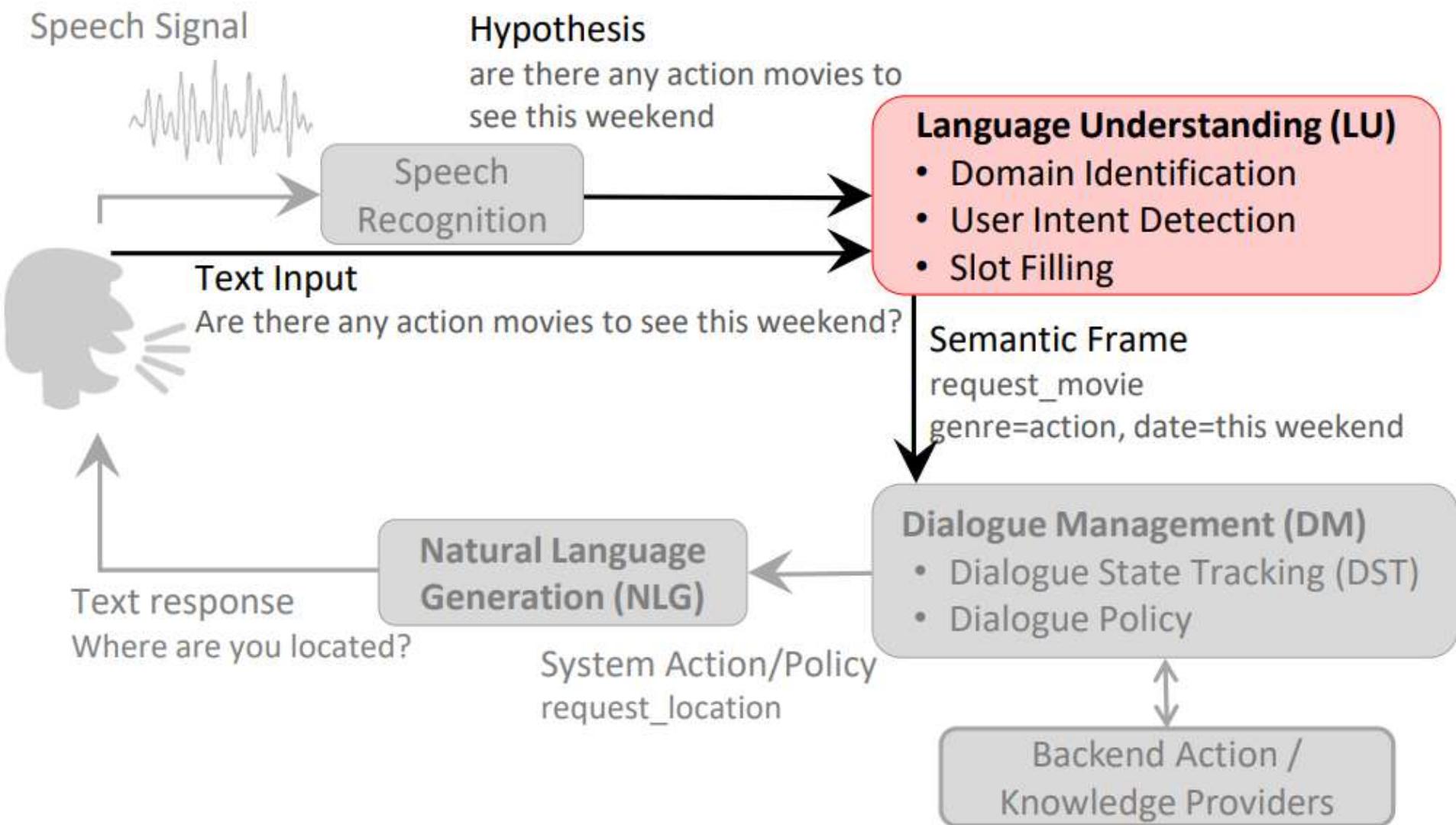


Approaches for different components of conversational AI system

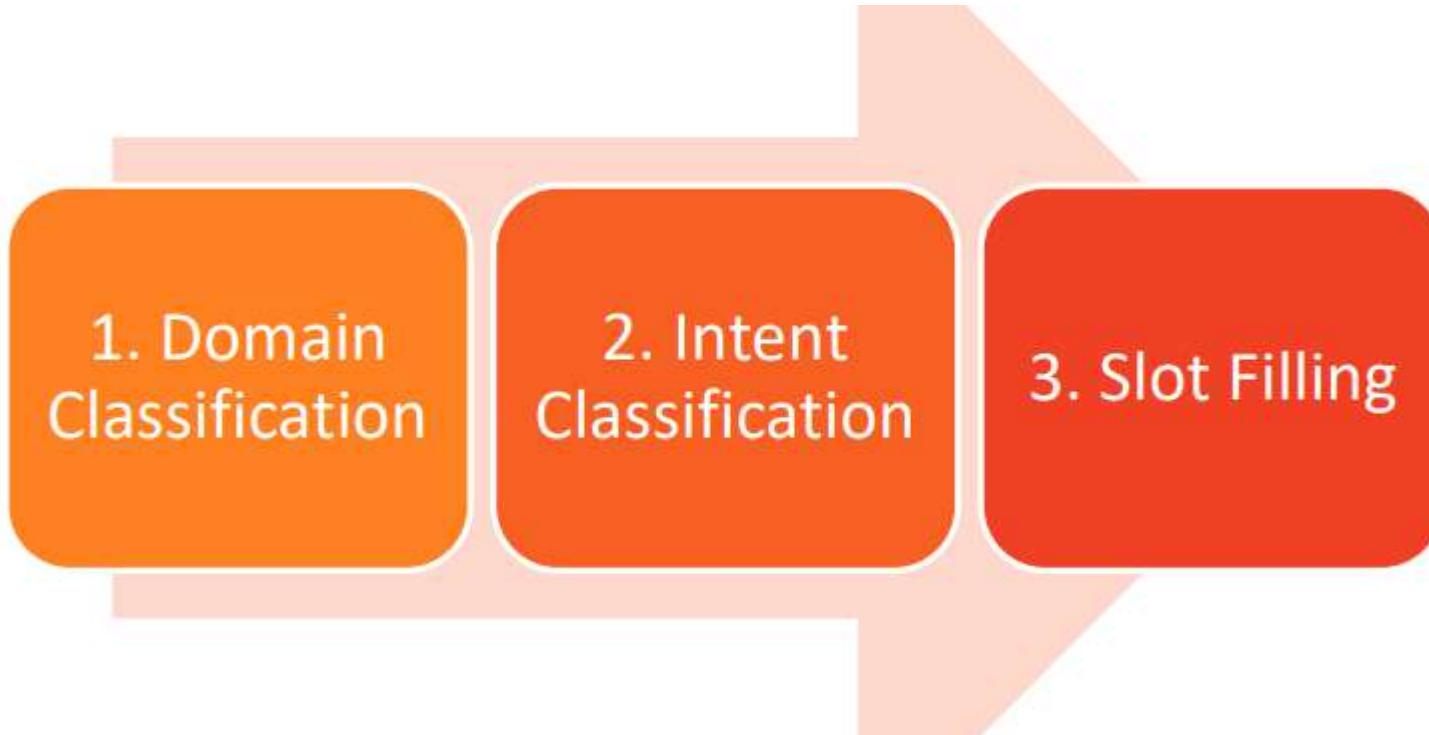
Approach	NLU (Understand Intent + Extract Slots)	NLG (Generate Responses)	Dialog Management (Planning Flow)	Example Applications
Rule-Based (Finite State / Pattern Matching)	Manual rules, pattern matching (if–else, regex). No learning.	Template-based responses; no variation; deterministic.	Hard-coded flowcharts; state machines; rigid paths.	IVR systems, early chatbots (ELIZA), FAQ bots, customer service scripts.
Statistical NLP (Pre-DL, ML Models)	Machine learning classifiers (SVM, CRF, HMM) for intent & slot tagging; requires labelled data.	Template-based with statistical ranking. Limited generation.	Probabilistic dialog policies (POMDPs).	Early Dialogflow, IBM Watson, telecom/banking bot NLU engines.
Deep Learning (RNNs, LSTMs, Seq2Seq)	Neural intent classifiers; BiLSTM-CRF slot tagging; moderate generalization; still domain-specific.	Seq2Seq or encoder–decoder models; limited creativity; domain-tuned.	Policy networks, trained with supervised or reinforcement learning; still domain-specific.	Alexa Skills (early versions), Rasa DL, Siri NLU pipelines.

Approaches for different components of conversational AI system

Approach	NLU (Understand Intent + Extract Slots)	NLG (Generate Responses)	Dialog Management (Planning Flow)	Example Applications
LLM-Based Conversational AI (GPT, Claude, Gemini)	Intent + slots extracted implicitly using prompts/in-context learning; no task-specific training needed; works zero-shot.	Free-form, high-quality generation; style control; multi-modal; creative.	LLM predicts next system actions; can manage context dynamically; no fixed flow.	ChatGPT-style assistants, enterprise Q&A, customer support bots.
Agentic AI (ReAct, Tool Use, Function Calling, Frameworks)	LLM extracts intents/slots in structured format (JSON, schema) for tools; works zero/low-shot; highly robust.	Multi-step reasoning, tool-calling, API execution, planning. NLG adapts after actions.	Autonomous dialog planning using reasoning loops (Thought → Action → Observation → Response); dynamic workflows; memory.	AI agents, workflow automation bots, enterprise assistants, travel/finance/HR multi-step task assistants.



Language Understanding



1. Domain
Classification

2. Intent
Classification

3. Slot Filling

Natural Language Understanding for filling dialog slots



1. Domain classification

Asking weather? Booking a flight?
Programming alarm clock?

2. Intent Determination

Find a Movie, Show Flight, Remove
Calendar Appt

3. Slot Filling

Extract the actual slots and fillers

Natural Language Understanding for filling slots

Show me morning flights from Boston to SF on Tuesday.

DOMAIN:	AIR-TRAVEL
INTENT:	SHOW-FLIGHTS
ORIGIN-CITY:	Boston
ORIGIN-DATE:	Tuesday
ORIGIN-TIME:	morning
DEST-CITY:	San Francisco

Natural Language Understanding for filling slots

Wake me tomorrow at six.

DOMAIN: ALARM-CLOCK

INTENT: SET-ALARM

TIME: 2017-07-01 0600-0800

Rule-based Slot-filling

Write regular expressions or grammar rules

Wake me (up) | set (the|an) alarm | get me up

Do text normalization

Siri uses GUS architecture: Condition-Action Rules

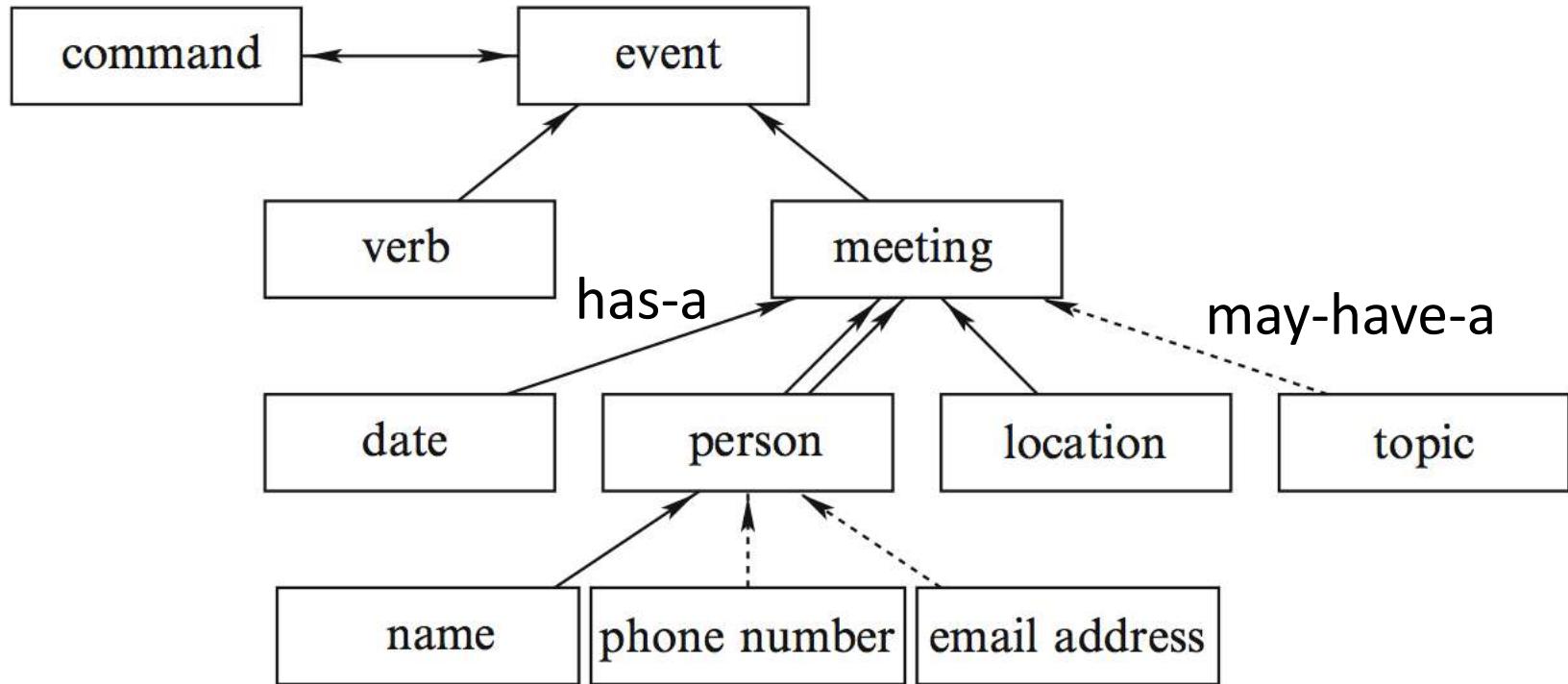
Active Ontology: relational network of concepts

- **data structures**: a **meeting** has
 - a date and time,
 - a location,
 - a topic
 - a list of attendees
- **rule sets** that perform actions for concepts
 - the **date** concept turns string
 - *Monday at 2pm* into
 - date object

Rule sets

- Collections of **rules** consisting of:
 - condition
 - action
- When user input is processed, facts added to store and
 - rule conditions are evaluated
 - relevant actions executed

Part of ontology for meeting task



meeting concept: if you don't yet have a location, ask for a location

Machine learning for slot-filling:

- Machine learning classifiers to map words to semantic frame-fillers
- Given a set of labeled sentences
 - “I want to fly to San Francisco on Tuesday”
 - Destination: SF
 - Depart-date: Tuesday
 - Build a classifier to map words to the slots
- Requirements: Lots of labeled data



Machine learning for slot-filling: Domain and Intent

I want to fly to San Francisco on Monday afternoon please

Use 1-of-N classifier (naive bayes, logistic regression, neural network, etc.)

- Input:
features like word N-grams

- Output:
Domain: AIRLINE
Intent: SHOWFLIGHT

Machine learning for slot-filling: Slot presence



I want to fly to San Francisco on Monday afternoon please

Use 1-of-N classifier (naive bayes, logistic regression, neural network, etc.)

- Input:
features like word N-grams, gazetteers (lists of cities)
- Output:
Destination-City

Machine learning for slot-filling: Slot filler



I want to fly to San Francisco on Monday afternoon please

Use 1-of-N classifier (naive bayes, logistic regression, neural network, etc.) for Destination City

- Input:
 - features like word N-grams, gazetteers (lists of cities)
- Output:
 - San Francisco

More sophisticated algorithm for slot filling: IOB/ BIO Tagging

- IOB or BIO Tagging
 - tag for the beginning (B) and inside (I) of each slot label,
 - plus one for tokens outside (O) any slot label.
 - $2n + 1$ tags, where n is the number of slots.

B-DESTINASTION

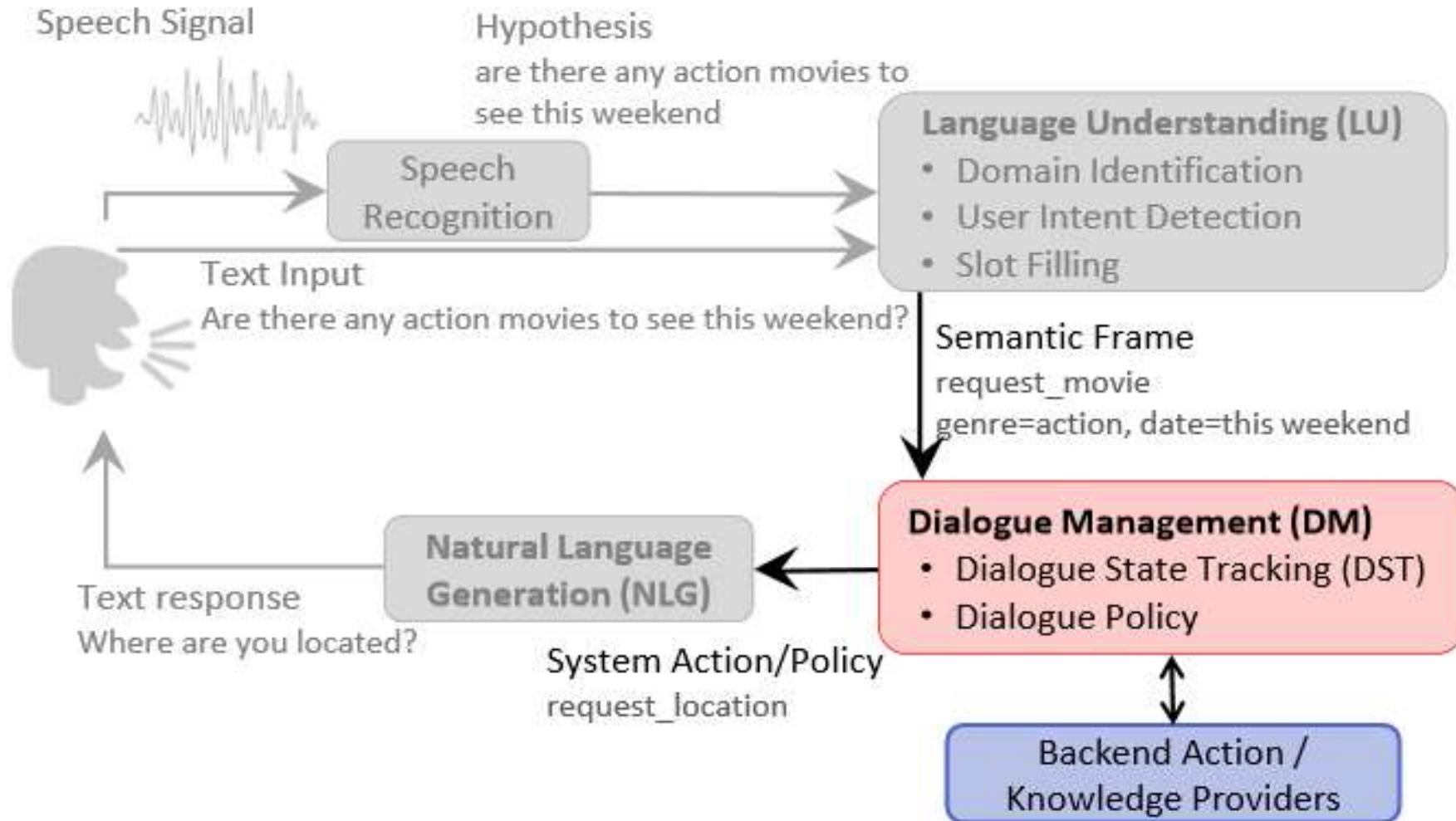
I-DESTINATION

B-DEPART_TIME

I-DEPART_TIME

0 0 0 0 B-DES I-DES 0 B-DEPTIME I-DEPTIME 0
I want to fly to San Francisco on Monday afternoon please

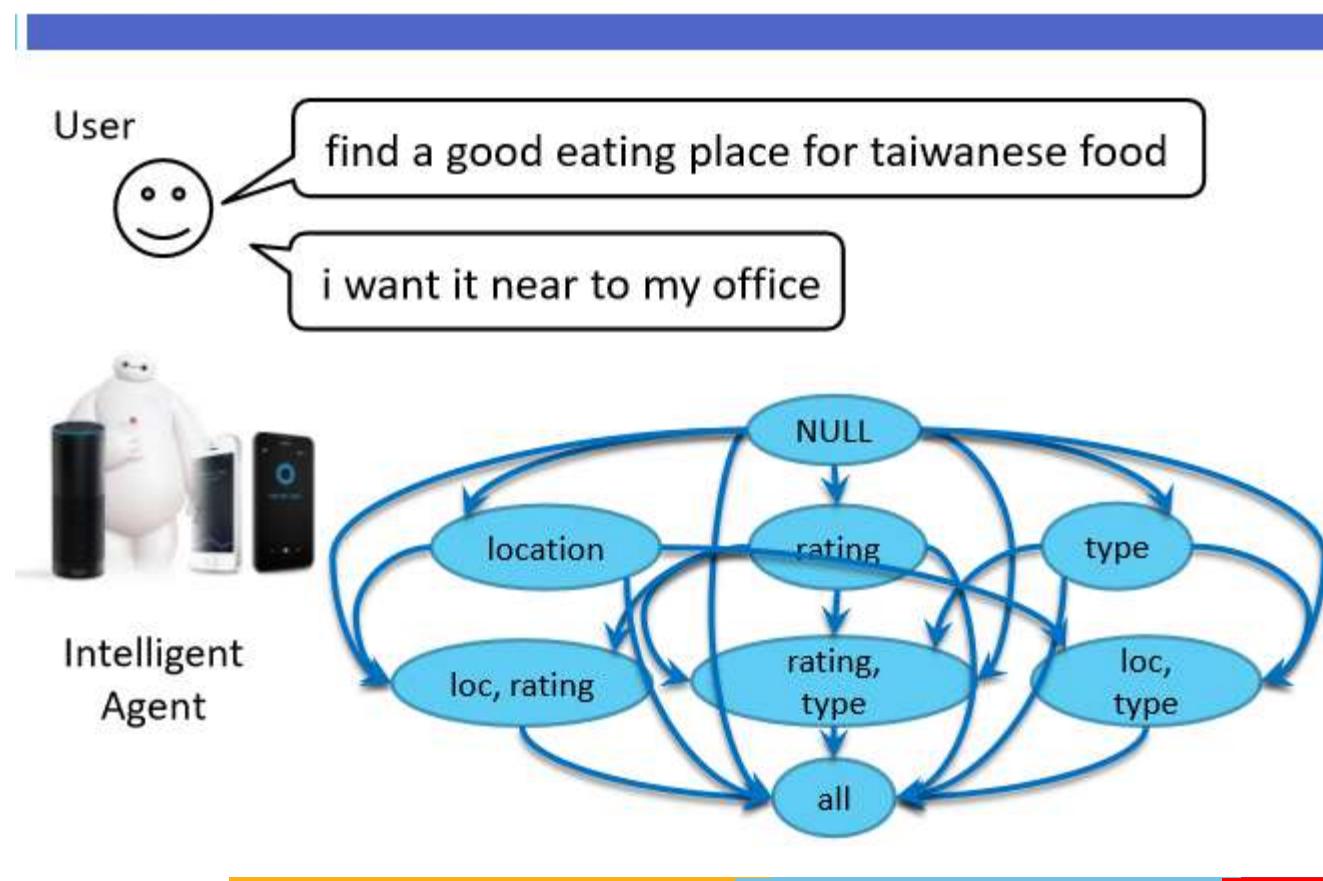
Dialog Management



Dialog Manager : Dialog State tracking

State Tracking

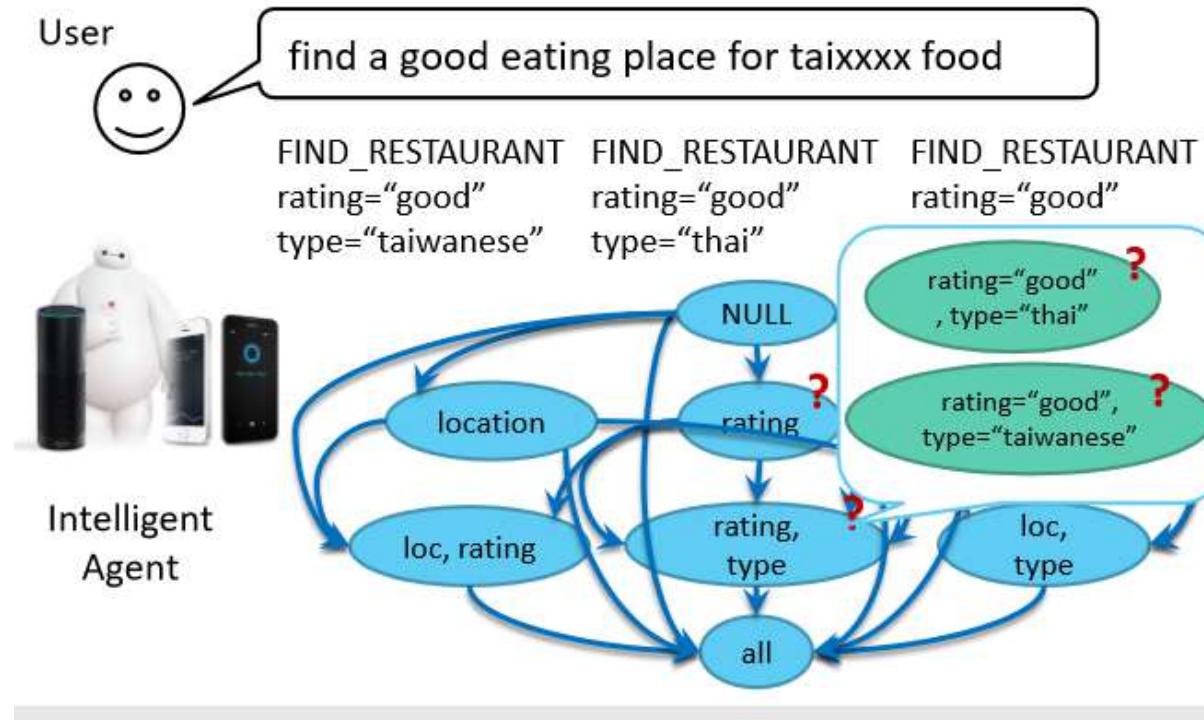
Requires Hand-Crafted States



Dialog Manager : Dialog State tracking

State Tracking

Handling Errors and Confidence



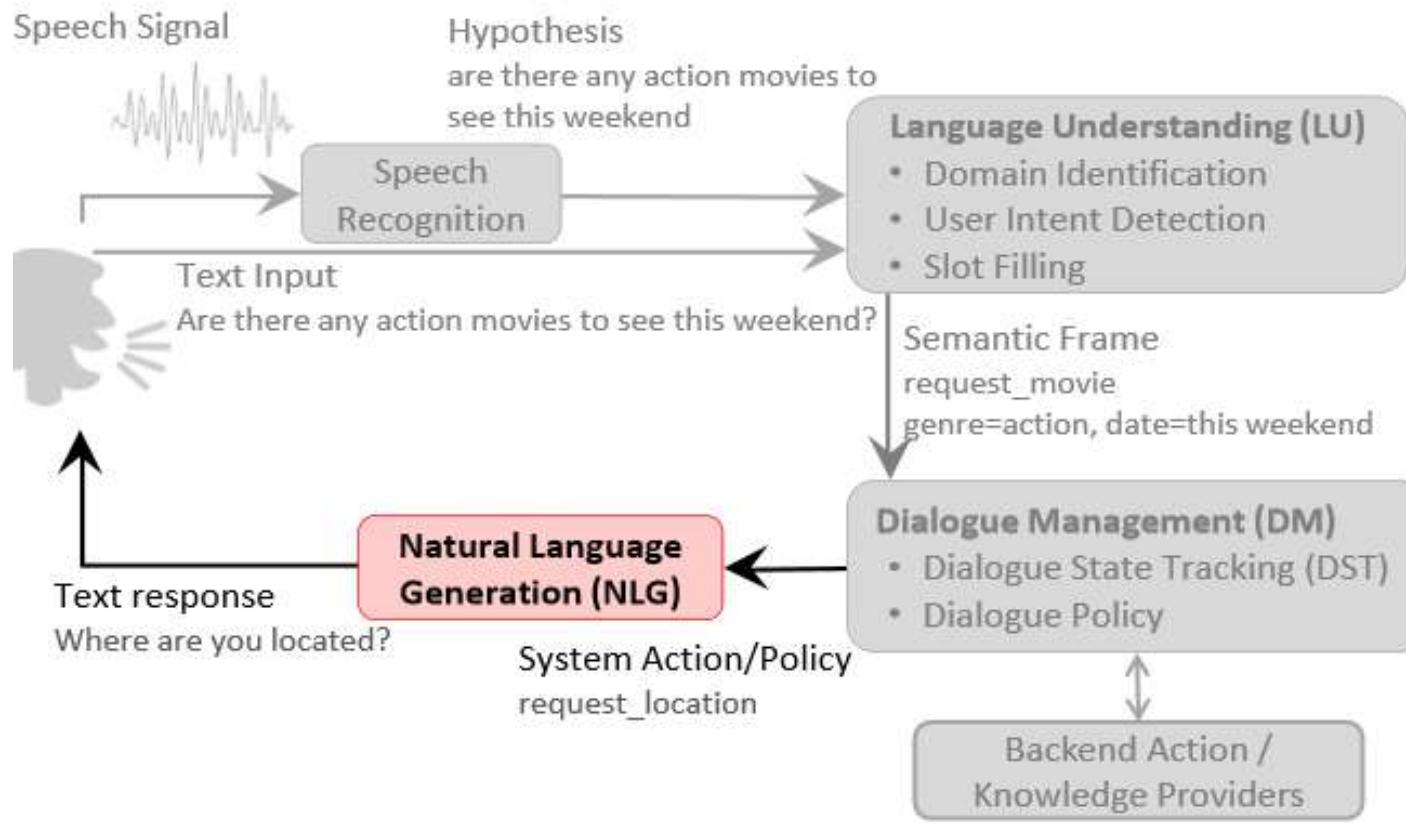
Dialog Manager : Dialog Policy

Dialogue Policy for Agent Action

- Inform(location="Taipei 101")
 - ▣ "The nearest one is at Taipei 101"
- Request(location)
 - ▣ "Where is your home?"
- Confirm(type="taiwanese")
 - ▣ "Did you want Taiwanese food?"

Natural Language Generation

Task-Oriented Dialogue System (Young, 2000)



Natural Language Generation

Output / Natural Language Generation

- Goal: generate natural language or GUI given the selected dialogue action for interactions



- Inform(location="Taipei 101")
 - "The nearest one is at Taipei 101" v.s.
- Request(location)
 - "Where is your home?" v.s.
- Confirm(type="taiwanese")
 - "Did you want Taiwanese food?" v.s.



Comparison of approaches

Approach	Architecture	Pros	Cons	Best Used For / Applications
1. Rule-Based	Decision trees, flowcharts	- Easy to implement- Predictable behavior	- Hard-coded- Not scalable- Poor with ambiguity	- Simple FAQs- IVR systems- Basic customer service
2. Slot-Filling (Modular)	NLU → DM → NLG	- Structured- Good for well-defined tasks	- Struggles with context- Needs labeled data	- Food ordering bots- Travel booking assistants
3. End-to-End Neural	Seq2Seq / Transformer	- Learns dialogue patterns- More fluid conversations	- Requires lots of data- Poor control/explainability	- Open-domain chat- Virtual companions- Dynamic support
4. Retrieval-Based	Query + Similarity Match → Predefined Reply	- Fast- Accurate with proper data- Grounded	- Limited to existing replies- Low flexibility	- Customer support- Helpdesk bots- FAQ assistants
5. LLM-Based	Prompt → Foundation Model (e.g., GPT)	- Flexible- Handles multiple tasks- Few-shot capable	- Costly- May hallucinate- Lacks structure	- Virtual agents- General assistant bots- Support chatbots
6. Hybrid (LLM + Rules/APIs)	LLM + Dialogue Manager + Tools/APIs	- Best of both worlds- Robust and natural	- Complex to design- Needs integration effort	- Enterprise assistants- Multistep workflows- Copilots

Evaluation

- Natural Language Understanding component Evaluation:
 1. Slot Error Rate for a Sentence =
$$\frac{\text{\# of inserted/deleted/substituted slots}}{\text{\# of total reference slots for sentence}}$$
Slot tagging is most important since it contributes maximum to quality of Dialog System
 2. Intent Accuracy: Correct intent identified
- Dialog Manager Component Evaluation
 1. State tracking Accuracy
 2. End-to-end evaluation (Task Success)

Evaluation Metrics

“Make an appointment with Chris at 10:30 in Gates 104”

Slot	Filler
PERSON	Chris
TIME	11:30 a.m.
ROOM	Gates 104

Slot error rate: 1/3

Task success: At end, was the correct meeting added to the calendar?



Dialog System Design: User-centered Design

1. Study the user and task
2. Iteratively test the design on users

Ethical Issues in Dialog System Design

- Machine learning systems replicate biases that occurred in the training data.
- Microsoft's Tay chatbot
 - Went live on Twitter in 2016
 - Taken offline 16 hours later
- In that time it had started posting racial slurs, conspiracy theories, and personal attacks
 - Learned from user interactions (Neff and Nagy 2016)

Ethical Issues in Dialog System Design

- Machine learning systems replicate biases that occurred in the training data.
- Dialog datasets
 - Henderson et al. (2017) examined standard datasets (Twitter, Reddit, movie dialogs)
 - Found examples of hate speech, offensive language, and bias
 - Both in the original training data, and in the output of chatbots trained on the data.
- Fairness in Machine Learning
- Dialog agents overwhelmingly given female names, perpetuating female servant stereotype(Paolino, 2017).

Ethical Issues in Dialog System Design: Privacy



- Remember this was noticed in the days of Weizenbaum
- Agents may record sensitive data
 - (e.g. “Computer, turn on the lights [answers the phone –Hi, yes, my password is...”],
- Which may then be used to train a seq2seq conversational model.

References

<https://www.topbots.com/building-conversational-ai/>

https://www.mindmeld.com/docs/intro/approaches_for_building_conversational_applications.html

<https://cibusgreyling.medium.com/large-language-models-langs-will-not-replace-traditional-chatbot-nlu-for-now-8a0c80623b8e>

<https://typeset.io/questions/how-are-langs-used-for-task-oriented-dialog-11zynlb2ys>

<https://zapier.com/blog/best-langs/>

References

- Speech and Language processing: An introduction to Natural Language Processing, Computational Linguistics and speech Recognition by Daniel Jurafsky and James H. Martin[3rd edition] - Chapter 26 Page 495
- <https://www.sciencedirect.com/science/article/abs/pii/0004370277900182>
- <https://blog.kore.ai/conversational-ai-top-20-trends-for-2020>
- <https://chatbotsmagazine.com/tutorials/home>
- <https://rasa.com/docs/rasa/user-guide/building-assistants/>

Training Data

- <https://rasa.com/docs/rasa/nlu/training-data-format/>
- <https://rasa.com/docs/rasa/user-guide/command-line-interface/#train-a-model>

NLU Components: <https://rasa.com/docs/rasa/core/slots/>

- <https://rasa.com/docs/rasa/nlu/components/>

Supporting Multiple languages: <https://rasa.com/docs/rasa/nlu/language-support/>

Voice Bots : <https://rasa.com/docs/rasa/user-guide/messaging-and-voice-channels/>

- <https://blog.rasa.com/connect-your-rasa-ai-assistant-to-amazon-alexa/>
- <https://blog.rasa.com/integrating-rasa-with-knowledge-bases/>

Transfer learning versus Fine-Tuning

Feature	Transfer Learning	Fine-Tuning
Weight Update	Primarily trains new layers or last few layers	Updates weights of pre-trained layers
Computational Cost	Lower	Higher
Training Time	Shorter	Longer
Data Requirements	Can work with less data	Benefits from larger, labeled datasets
Application	Related tasks, limited data	Closely related tasks, more data

Component	Traditional Stack	LLM Stack	RAG Stack	Agent Stack
Intent/Slot NLU	spaCy, sklearn, CRF	GPT-4, Claude (prompt-based)	GPT + retrieval	LLM (with memory)
DST	Rasa tracker, rules, neural DST	GPT prompt or function schema	Persisted in vector DB or JSON state	Planning memory, scratchpad
Policy/Planner	Rule-based, ML (e.g., TED)	Prompt-controlled LLM	Retrieval+prompt routing	Agent planner (LangGraph, ReAct)
NLG	Templates or seq2seq	GPT/Claude/Bard	Context-aware GPT response	Tool-aware LLM generation
Backend APIs	REST APIs	LLM function-calling	Tool calls + knowledge base	Chained tool/API calls
Memory	Tracker store (SQLite, Redis)	Prompt state or LangChain memory	Vector store (FAISS, Pinecone)	Stateful planning memory
Deployment	Docker, Kubernetes, Azure Bot Services	OpenAI/AWS Lambda + API Gateway	Streamlit + vector DB	Agent server or serverless functions



Natural Language Processing Applications



BITS Pilani
Pilani Campus

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



Session 6 : Knowledge Graph Real world Applications

These slides are prepared by the instructor, with grateful acknowledgement of Prof. Philip Cimiano , Johanna Völker , Paul Buitelaar and many others who made their course materials freely available online.

Session Content

- How to Use Knowledge Graphs
 - Why we need Knowledge Graphs
 - Real world Case studies
 - Semantic Search
 - Recommendation Systems
 - KG in conversational AI
 - Tourist Domain Use case
 - Energy Domain Use case
 - Knowledge Graph and LLM
-

Knowledge Graph

- **ontology + data = knowledge graph**
- Information extraction techniques for performing [named entity recognition](#). Another subfield which has gained much interest from the community is [keywords extraction](#).
- A **knowledge graph** is a way of storing data that resulted from an information extraction task. Many basic implementations of knowledge graphs make use of a concept we call **triple**, that is a set of three items(a subject, a predicate and an object)

Knowledge is the key

- **Semantic descriptions** of entities and their **relationships**
 - Uses a **knowledge representation formalism**
(Focus here: RDF, RDF-Schema, OWL)
 - **Entities**: real world objects (things, places, people) and abstract concepts (genres, religions, professions)
 - **Relationships**: graph-based data model where relationships are first-class
 - **Semantic descriptions**: types and properties with a well-defined meaning (e.g. through an ontology)
 - Possibly axiomatic knowledge (e.g. rules) to support automated reasoning
-

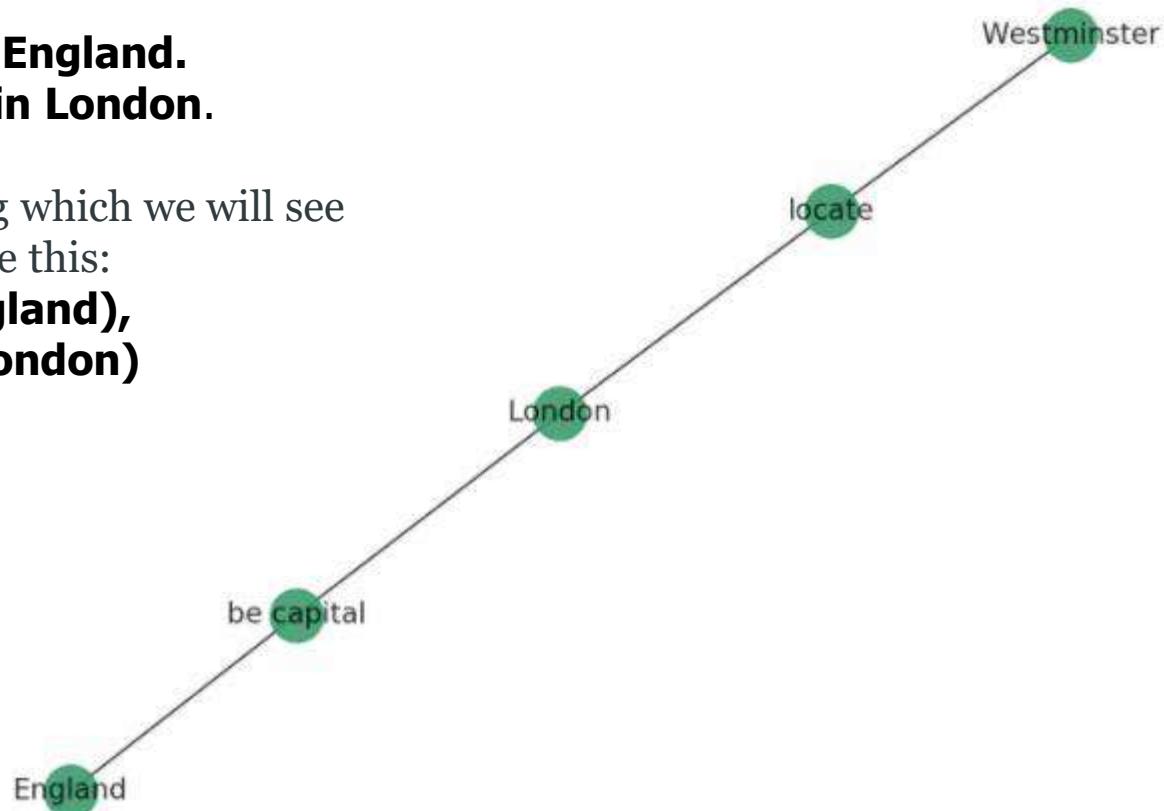
Example

London is the capital of England.

Westminster is located in London.

After some basic processing which we will see later, we would 2 triples like this:

**(London, be capital, England),
(Westminster, locate, London)**



Knowledge graph representation

Knowledge Graph technologies

Knowledge Graph technologies consists of:

- knowledge representation and reasoning (languages, schema and standard vocabularies),
- knowledge storage (graph databases and repositories),
- knowledge engineering (methodologies, editors and design patterns), and
- knowledge learning, including schema learning and population

Building Knowledge Graph

- Identifying the entities and the relation between them is not a difficult task for us. However, **manually building a knowledge graph is not scalable.**
- Nobody is going to go through thousands of documents and extract all the entities and the relations between them!
- **To build a knowledge graph from the text, it is important to make our machine understand natural language.**
- This can be done by using NLP techniques such as sentence segmentation, dependency parsing, parts of speech tagging, and entity recognition

Extracting structured data from Wikipedia

```
@prefix dbpedia <http://dbpedia.org/resource/>.  
@prefix dbterm <http://dbpedia.org/property/>.
```

dbpedia:Amsterdam

```
dbterm:officialName "Amsterdam" ;  
dbterm:longd "4" ;  
dbterm:longm "53" ;  
dbterm:longs "32" ;  
dbterm:leaderName dbpedia:Lodewijk_Asscher ;  
...  
dbterm:areaTotalKm "219" ;  
...
```

dbpedia:ABN_AMRO

```
dbterm:location dbpedia:Amsterdam ;  
...
```

Amsterdam	
	
The Keizersgracht at dusk	
Location of Amsterdam	
Coordinates:	52°22'23"N 4°53'32"E
Country	Netherlands
Province	North Holland
Government	
- Type	Municipality
- Mayor	Job Cohen ^[1] (PvdA)
- Aldermen	Lodewijk Asscher Carolinus Gehrels Tjeerd Herrema Maarten van Poelgeest Marijke Vos
- Secretary	Erik Gerritsen
Area <small>[2][3]</small>	
- City	219 km ² (84.6 sq mi)
- Land	166 km ² (64.1 sq mi)
- Water	53 km ² (20.5 sq mi)
- Urban	1,003 km ² (387.3 sq mi)
- Metro	1,815 km ² (700.8 sq mi)
Elevation <small>[4]</small>	2 m (7 ft)
Population <small>(1 October 2008)[5][6]</small>	
- City	756,289
- Density	4,459/km ² (11,548.8/sq mi)
- Urban	1,364,422
- Metro	2,158,372
- Demonym	Amsterdamer
Time zone	CET (UTC+1)
- Summer (DST)	CEST (UTC+2)
Postcodes	1011 – 1109
Area code(s)	020
Website: www.amsterdam.nl	

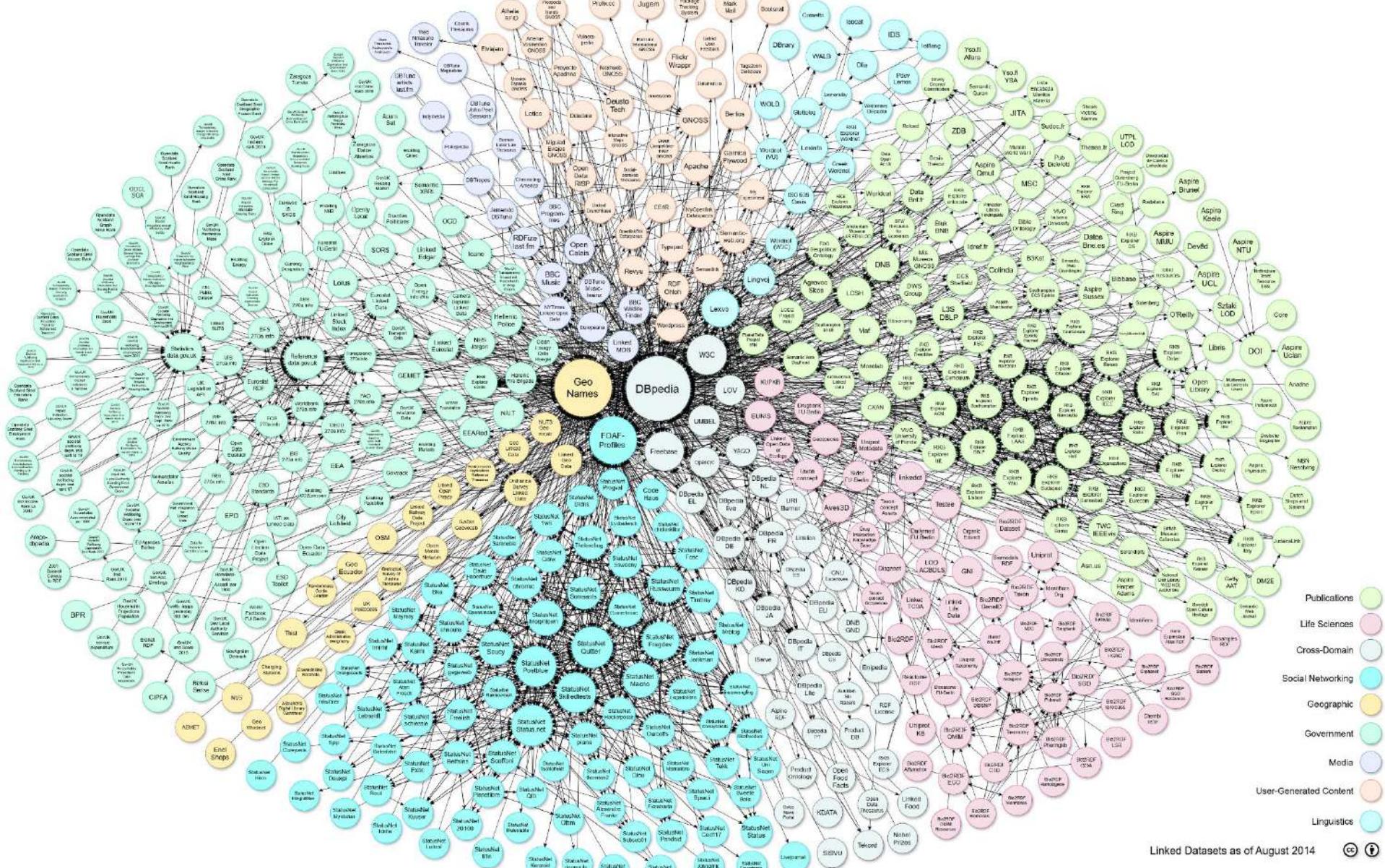
Automatic links among open datasets

```
<http://dbpedia.org/resource/Amsterdam>
owl:sameAs <http://rdf.freebase.com/ns/...> ;
owl:sameAs <http://sws.geonames.org/2759793> ;
...
```

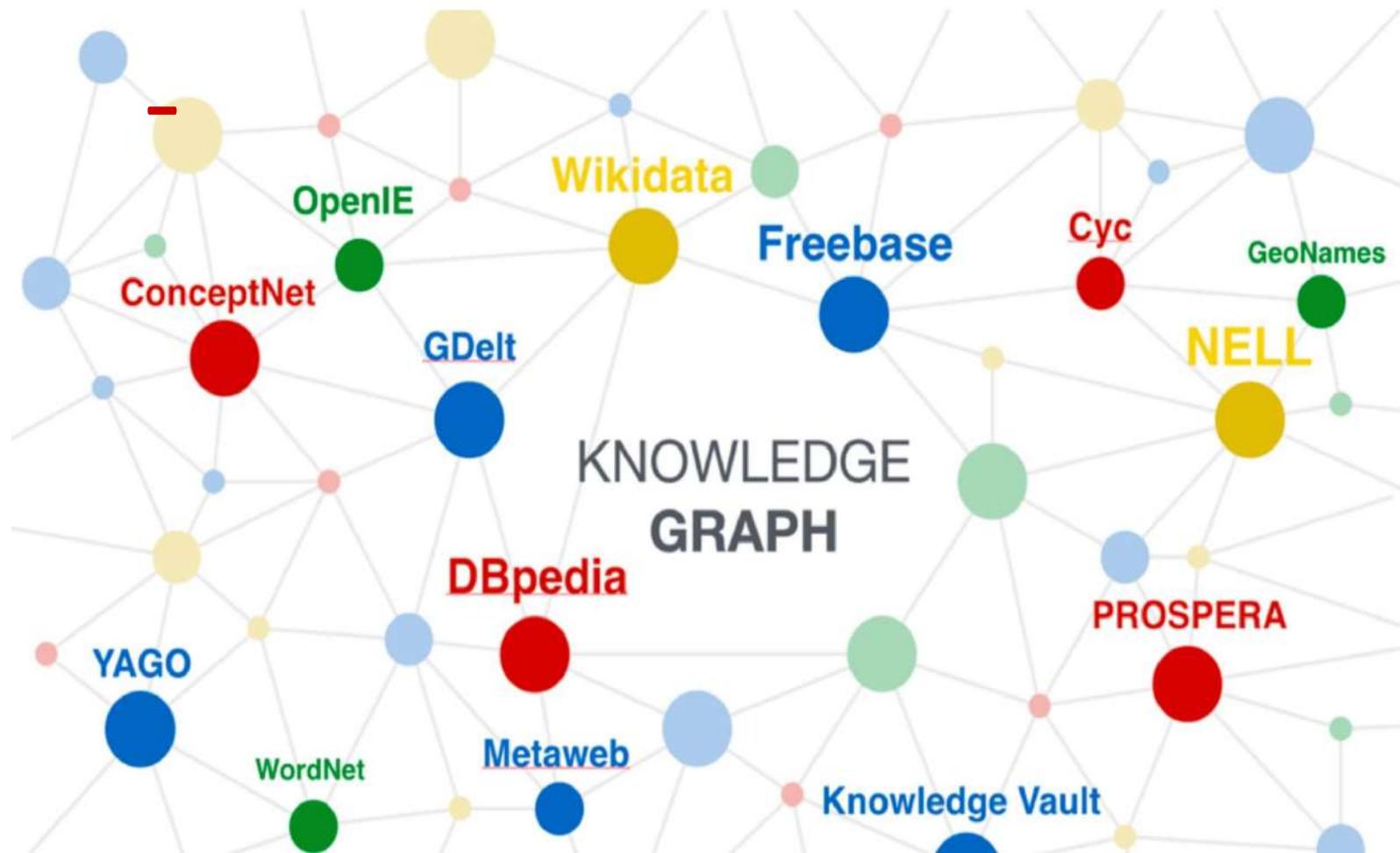
```
<http://sws.geonames.org/2759793>
owl:sameAs <http://dbpedia.org/resource/Amsterdam>
wgs84_pos:lat "52.3666667" ;
wgs84_pos:long "4.8833333";
geo:inCountry <http://www.geonames.org/countries/#NL> ;
...
```

Processors can switch automatically from one to the other...

The LOD “cloud”



Open source KGs



Knowledge Graph Use

Knowledge Graphs:

Connect siloed data. **Simplify** complicated data.



Simplifies access to complex data to address unanticipated questions



Quickly profiles, connects and harmonizes data from multiple sources, including unstructured



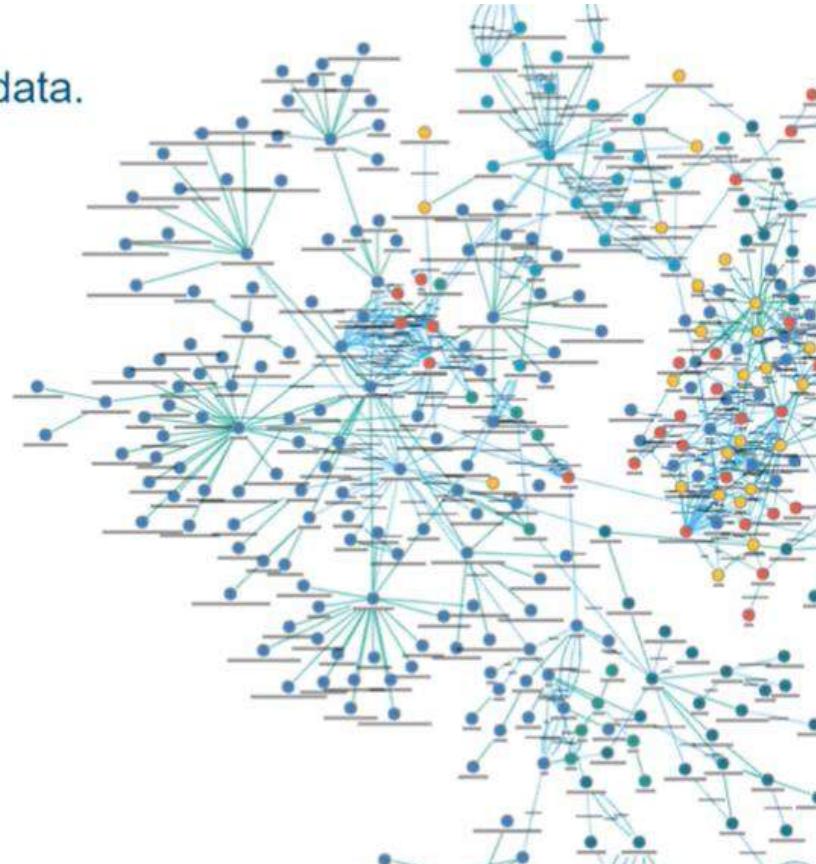
Presents tailored views, services and experiences to different personas with conceptual models



Flexibly accommodates new data sources and use cases on the fly, with minimal impact



Enables scalable knowledge graphs capable of accomodating enterprise data sources and use cases



ODSC Webinar | Unlocking the Power of Knowledge Graphs for Generative AI in Enterprise Environments

Few domains KG is used

Bioportal – medical
Government
Cyber security
Smart City – Urban Data Analytics
Navigation
Amazon-Usage, movies, series, trends etc
Weather
Financial Services
Manufacturing- supply chain

KG Motivation

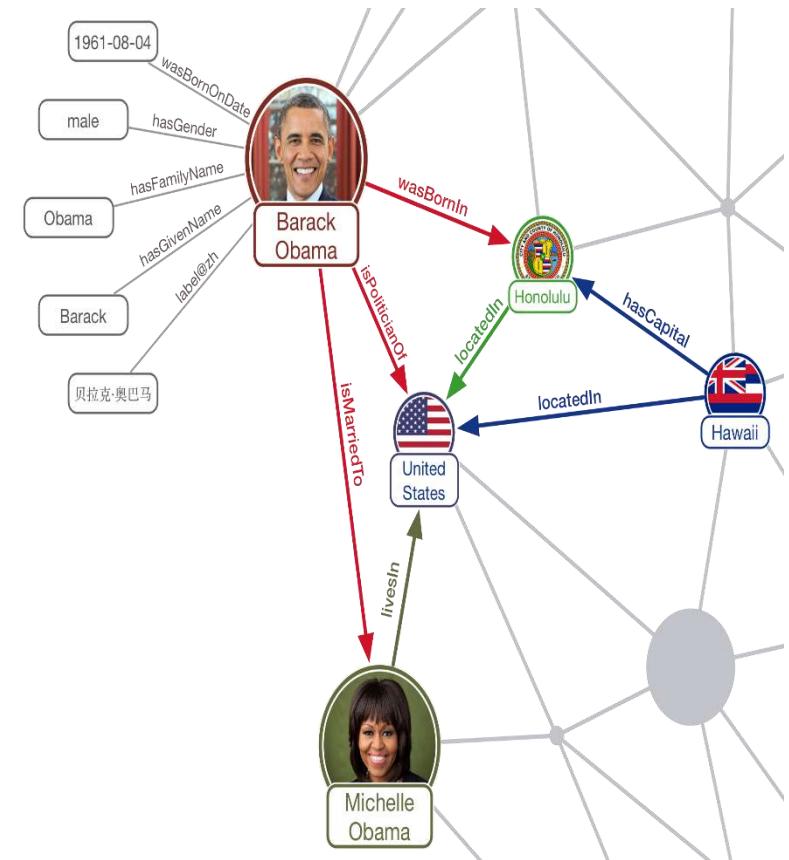
- Netflix recommends movies of your choice?
- How do airports streamline their flights?
- How cell phone towers are connected?
- Or any other application that deals with understanding the human context of data connections.
- Technology behind all these applications is a
Knowledge Graph

[Gartner](#) predicts that knowledge graphs will be a part of 80% of data and analytics innovations by **2025**.

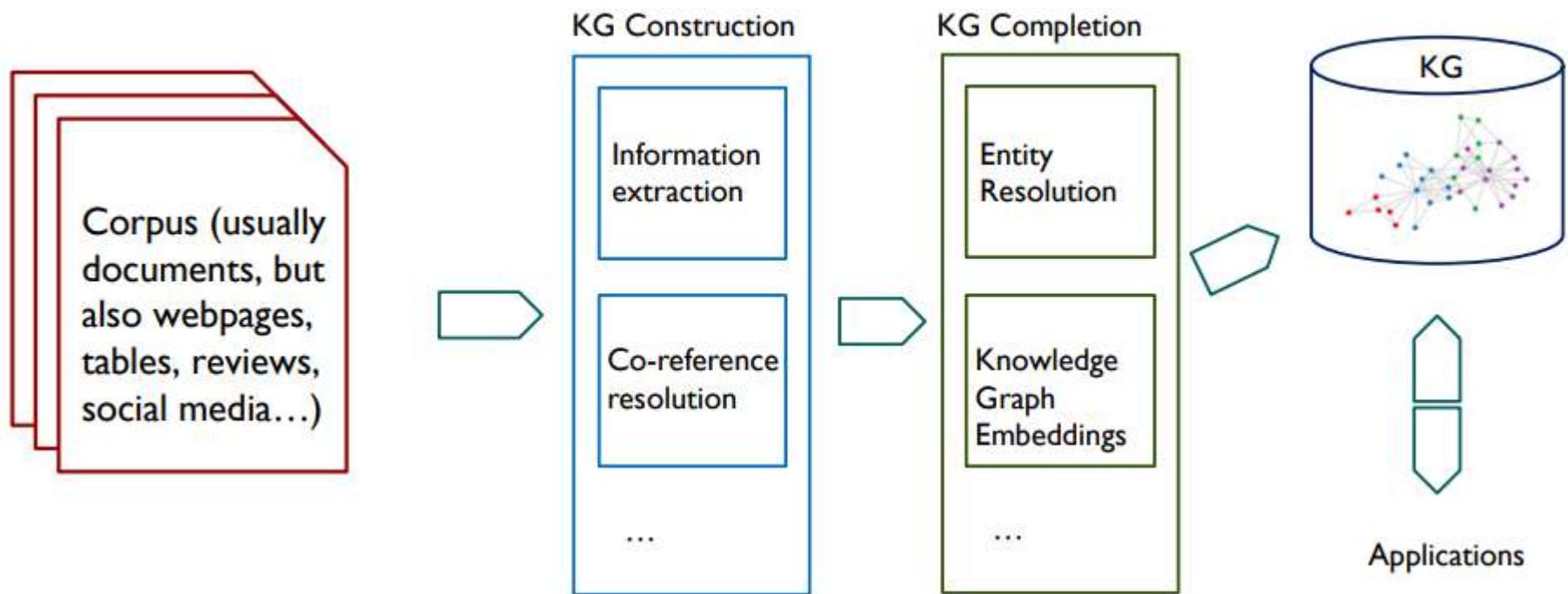
KG Example

Set of triples, where each triple (h, r, t) represents a **relationship r** between **head entity h** and **tail entity t**

(Barack Obama, wasBornOnDate, 1961-08-04),
 (Barack Obama, hasGender, male),
 ...
 (Hawaii, hasCapital, Honolulu),
 ...
 (Michelle Obama, livesIn, United States)



KG Workflow



Information Extraction

Isaac Newton

From Wikipedia, the free encyclopedia.

<Person>

Sir Isaac Newton (25 December 1642 – 20

March 1727 by the Julian calendar in use in

England at the time; or 4 January 1643 – 31

March 1727 by the Gregorian calendar) was an

English physicist, mathematician, astronomer,
philosopher, and alchemist; who wrote the

Philosophiae Naturalis Principia Mathematica

(published 5 July 1687)¹, where he described

universal gravitation and, via his laws of motion,

laid the groundwork for classical mechanics

Newton also shares credit with Gottfried Wilhelm

Leibniz for the development of differential calculus.

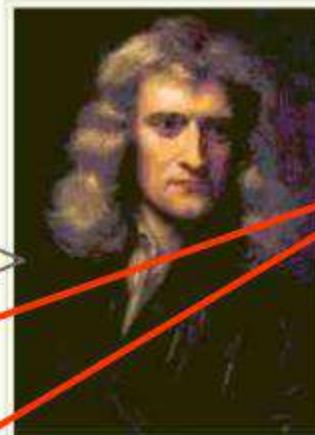
However, their work was not a collaboration; they both discovered

calculus separately but nearly contemporaneously.

<TimePeriod>

<Scientist>

<Scientist>



Sir Isaac Newton
in Kneller's <Painter>
portrait of 1689.

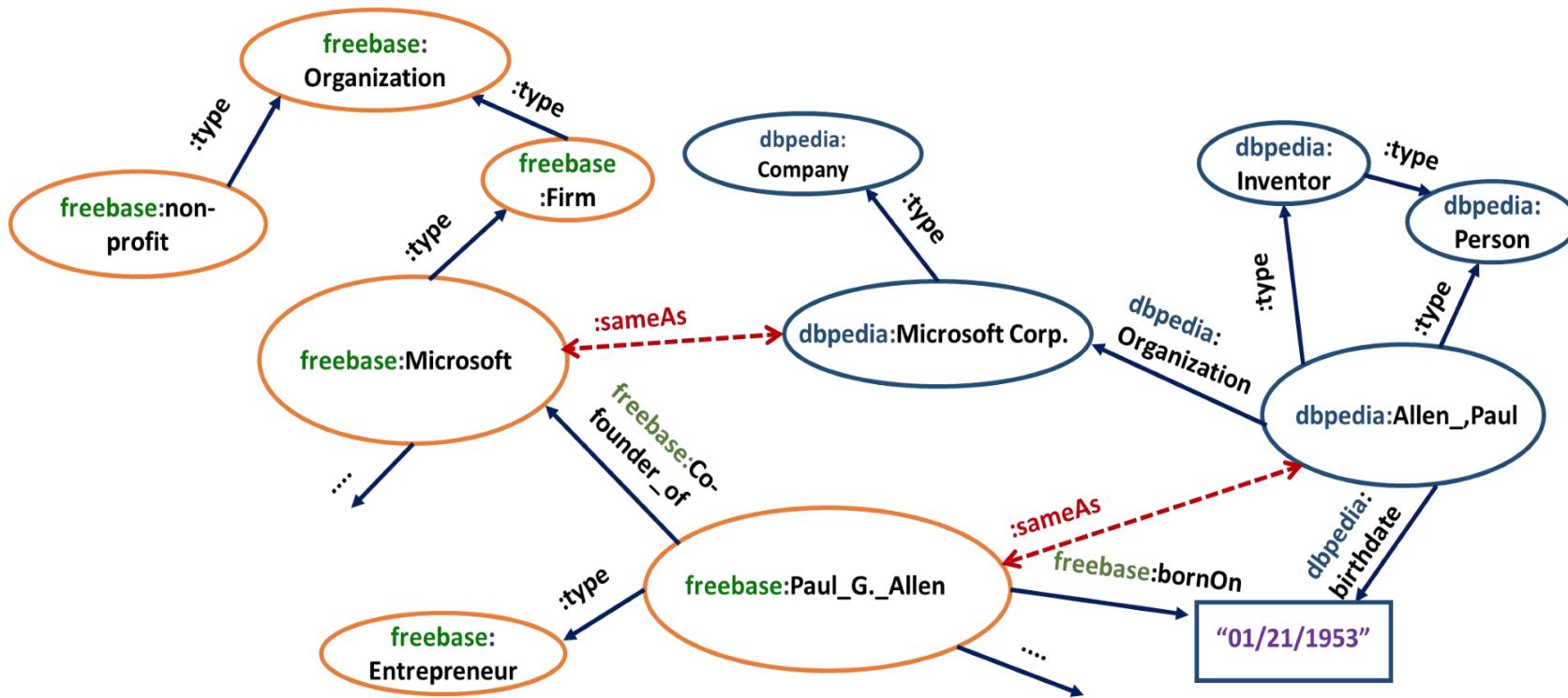
<Publication>

IRDM WS 2005 <Person>

Coreference Resolution



KG Example



Types of Data

- Knowledge Graphs and applications working with them deal with three different types of data:
 - **Static data** are data that do not likely change frequently. For instance, the address of a hotel or the birthdate of a person do not change very often. This low velocity allows to directly store these data in a Knowledge Graph repository.
 - **Dynamic data** change with a high frequency, for instance, age, weather forecast, traffic information or share prices. Here the actual data must be accessed periodically or on demand. It is possible to store these data for future data mining, but the current data must be accessed dynamically.
 - **Active data**, which changing the state of an external resource. For instance, a booking action changes the state of an external resource (e.g. creation of new room reservations at a hotel)

Semantic Web Services

- For dynamic data like temperature at a given location and time, a request (e.g. HTTP GET) to the weather web service must be made to retrieve the necessary information at the query-time.
- For active data, a request (e.g. HTTP PUT) to a hotel's booking web service must be sent to make a room reservation.
- Intelligent applications like conversational interfaces then access the web services on the fly

Knowledge Graphs and Dialog systems



- Two main directions in which one can use a Knowledge Graphs for this purpose:
 - to power the language understanding part of the dialogue system and
 - to react on the conversations and provide additional interactions, information, and recommendations to the user engaged in conversations with the dialog system

Language Understanding

- Use the Knowledge Graphs to provide training data for the Natural Language Understanding service (e.g., DialogFlow).
- Ingest from the Knowledge Graph as training data for the entity recognition task (e.g., Vienna is a City) and provide (semi-) automatically generate intents and example questions.
- Based on the Knowledge Graph structure
 - we can generate on the one hand entities and synonyms
 - intents needed in the Natural Language Understanding service based on the entities,
 - the relations between these entities in the Knowledge Graph
- ontology-to-text approaches to generate example questions that can be used to train the Natural Language Understanding service

React on the conversations

- Can elaborate on the topic of discussion and provide additional interesting facts.
 - Let us consider a dialogue system providing information about events.
 - Based on the knowledge from a touristic Knowledge Graph, system can provide the user with detailed information about an event, but also additional information about the connected entities, e.g., artist, location, and means of transportation related to that event
- Improve the handling of the conversation context
 - Using a template or rule-based approach one can drive interaction with the user to provide context-dependent answers (e.g., prices based on the role of the user), the invocation of actions (booking of a hotel room with follow up questions), or a combination of both (e.g. which outdoor events happen on the weekend if the weather permits).
- Refine the search for products or services in a dialogue system
 - actions and APIs that are modelled as part of the Knowledge Graph might require input data of a specific type, and can be used to steer the dialogue

Semantic Annotation and Dialog Systems

- Dialogue system can generate a backend driven intent and training sentences semi-automatically.
- It can benefit from the hypermedia links of a Web API to extract the invocation flow that guides the dialogue

User: *I want a hotel room in Seefeld from 12.09.2019 to 14.09.2019*

Bot: *How many guests?*

User: 2

Bot: *Ok I have found following hotel rooms: 1. Double room, Shower or Bathtub, WC offered by Alpenpark Resort for 200 EUR per night. 2. Double room, shower or bath, toilet offered by Central Seefeld for 119 EUR per night. You can purchase an offer.*

User: *Ok I want number 2. . . .*

.....

Bot: *Done. Here is the reservation:*

Semantic Annotation and Dialog Systems

- First the action is processed and a search intent for lodging reservation including its required parameters is created.
- Then user's request that already contains some of the required information for booking a room is matched with this intent.
- After the missing input is elicited (i.e. occupancy), the search is done via the endpoint defined in the action.
- The action returns hotel rooms with offers and potential buy actions attached to them.
- After user selects an offer, the buy action on that offer is completed and reservation is shown to the user.

Semantic Search and KG

- Google Knowledge Graph, a knowledge base containing already around 100 billion facts about more than 1 billion entities
- Present information from external sources in a structured way become possible
- Only around 25% of the result shows classical web resources
- Google Assistant on mobile devices traditional web results are no longer shown but only content selected and aggregated by Google.
- Alexa, Bixby, Cortana, Facebook messenger, Google Assistant, Siri, and others provide personalized and (spoken) message-based access to information.

Google Knowledge Graph

About 38,700,000 results (0.67 seconds)

Wonder Woman (2017) - IMDb
<https://www.imdb.com/title/tt6042790/>
 ★★★★☆ (Rating: 7.6/10 - 360,588 votes)
 When a pilot crashes and finds herself in the outside world, Diana, an Amazonian warrior in training, arrives on Earth to discover her full powers and thus destiny.
 Full Cast & Crew · Chris Pine · Gal Gadot · Parris Goebel

Wonder Woman (2017 film) - Wikipedia
[https://en.wikipedia.org/wiki/Wonder_Woman_\(2017_film\)](https://en.wikipedia.org/wiki/Wonder_Woman_(2017_film))
 Wonder Woman is a 2017 American superhero film based on the DC Comics character of the same name. It was directed by Patty Jenkins, with a screenplay by Allan Heinberg, from a story by Heinberg, Zack Snyder, Gal Gadot, Patty Jenkins, Blanca Guerra, and Doctor Pesci.

Top stories

- Oscars voting ends: today. Will Wonder Woman finally break the anti-superhero streak? [Washington Post](#)
- Fashion War: Wonder Woman Gal Gadot Inflates Lebanese with Dress Design [Bleitman](#)
- Gal Gadot Diet and Fitness Routine | POPSUGAR Fitness Australia [POPSUGAR Australia](#)

Google Larry Page

Web Images Maps Shopping News More

Larry Page - Wikipedia, the free encyclopedia
en.wikipedia.org/w/index.php?title=Larry_Page&oldid=800000301
 Lawrence "Larry" Page (born March 29, 1973) is an American computer scientist and Internet entrepreneur who is the co-founder of Google, alongside Sergey Brin. He was born in Humble, Texas, and grew up in Austin, Texas. He studied computer science at Stanford University, where he met Sergey Brin. They founded Google in 1995 while still students.

Larry Page - Forbes
www.forbes.com/people/larry-page/
 Larry Page is 1 of Forbes' #20 Billionaires, #12 Power People, #13 Forbes 400.

Larry Page - Google
<https://relax.google.com/r/LarryPage>

Larry Page
 Larry Page (born March 29, 1973) is an American computer scientist and Internet entrepreneur who is the co-founder of Google, alongside Sergey Brin. On April 4, 2011, Page succeeded Eric Schmidt as the chief executive officer of Google. Wikipedia

Born March 29, 1973 (age 46), East Lansing, MI
 Height: 5' 11" (1.80 m)
 Spouse: Lucinda Southworth (m. 2007)
 Siblings: Carl Victor Page, Jr.
 Education: East Lansing High School (1991–1997); More
 Awards: Macmillan Prize, TR100

Recent posts

Just learned the new phone release! [HOBUM](#) (Aug 2, 2013)

People also search for

- Sergey Brin
- Eric Schmidt
- Larry Ellison
- Mariah Carey
- Bill Gates

Search Results (entity specific)

chetana gavankar



 LinkedIn · Chetana Gavankar
1.8K+ followers

Chetana Gavankar - Pune, Maharashtra, India

Chetana Gavankar. Pune, Maharashtra, India. 2K followers 500+ connections. See your mutual connections. View mutual connections with Chetana.

 BITS Pilani
<https://www.bits-pilani.ac.in> › pilani › chetana-gavankar

Dr. Chetana Gavankar

Dr. Chetana is a Professor in the Work Integrated Learning Programmes (WILP) and serves as the Program Lead for the CSIS department at BITS Pilani.

 Google Scholar
<https://scholar.google.com> › citations

Chetana Anoop Gavankar

Chetana Anoop Gavankar. Professor. Verified email at pilani.bits-pilani.ac.in - Homepage · Natural Language ProcessingMachine learning ...

 Department of Computer Science and Engineering. IIT Bombay
<https://www.cse.iitb.ac.in> › ~chetana

chetana-gavankar - CSE, IIT Bombay

I am a PhD student at IITB-Monash Research Academy , the joint PhD program between IIT Bombay and Monash University, Australia. My PhD supervisors at IIT ...

 ResearchGate
<https://www.researchgate.net> › profile › Chetana-Gavan...

About

Research interests: Natural Language Processing, Machine learning, Deep Learning, Semantic Web

Profiles



LinkedIn



Google Scholar

Query specific UI

Google search results for "good hotels in kerala". The search bar shows the query. Below it are filters for travel type, dates (Sat 21 Jun to Sat 28 Jun), and guest count (2). The results section shows two hotel listings: "Sterling Guruvayur" and "Grand harmony beach resort", each with a deal price and a map view.

Sterling Guruvayur

- 4-star hotel
- Free parking
- Room service
- Breakfast (\$)
- Air conditioning
- Full-service laundry
- Free Wi-Fi
- Restaurant
- Accessible

4.2 ★ (1.9K) "Good service, calm and quiet place."

Grand harmony beach resort

- 4.7 ★ (60)

DEAL ₹2,688
23% less than usual

DEAL ₹1,399
24% less than usual

Map View: A map of Kerala showing various locations with price tags for different hotels. Labels include Guruvayur, Thrissur, Palani, Kodaikanal, Periyar National Park, Thenmala, Thiruvananthapuram, Tirunelveli, and Tiruchirappalli.

Semantic Search

places to visit san jose

All Maps Images News Shopping More Settings Tools

About 77,800,000 results (1.02 seconds)

Top things to do in San Jose

- Winchester Mystery House
- The Tech Interactive
- Mission Peak
- Happy Hollow Park & Zoo

Quirky mansion with odd design details
Interactive displays & an IMAX theater
Mountain with an iconic summit pole
Animals, activities & conservation focus

[More things to do](#)

Things To Do & Places To See in San Jose | Visit San Jose

<https://www.sanjose.org/things-to-do>

Vibrant, diverse and accessible, San Jose shines with 300+ days of sunshine that assures indoor and outdoor activities for every inclination. Whether it's nightlife ...
Attractions and Amusement - Kids and Family · San Jose Wineries · Nightlife

Complete List of San Jose Attractions | Visit San Jose

<https://www.sanjose.org/things-to-do/attractions-and-amusement>

Find a complete list of San Jose attractions and things to do - perfect for tourists and ... Browse the best places to see in San Jose and the surrounding Bay Area.

25 Best Things to Do in San Jose (CA) - The Crazy Tourist

[https://www.thecrazytourist.com/.../United States/California \(CA\)](https://www.thecrazytourist.com/.../United%20States%20/California%20(CA)%20)

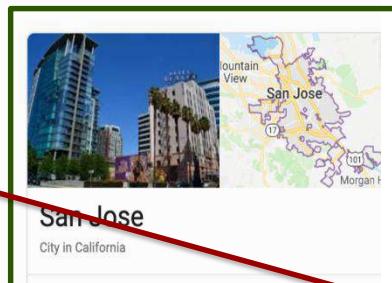
25 Best Things to Do in San Jose (CA): Winchester Mystery House: Flickr. Tech Museum Of Innovation: Flickr. Children's Discovery Museum: Flickr. Rosicrucian Egyptian Museum: Flickr. San Jose Heritage Rose Garden: wikimedia. Basilica Of St. Joseph: Flickr. Alum Rock Park: Flickr. Happy Hollow Zoo Entrance: Flickr.

THE 15 BEST Things to Do in San Jose - 2019 (with Photos ...

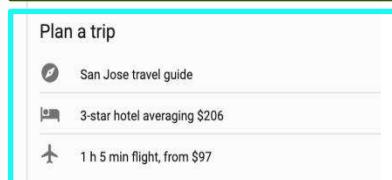
https://www.tripadvisor.com/Attractions-g33020-Activities-San_Jose_Calif...

... attractions. Find what to do today, this weekend, or in August. We have reviews of the best places to see in San Jose. Visit top-rated & must-see attractions.

Free Entry (36) · Things to Do in San Jose · Museums in San Jose · Santana Row



Knowledge panel



Recognition of user intent

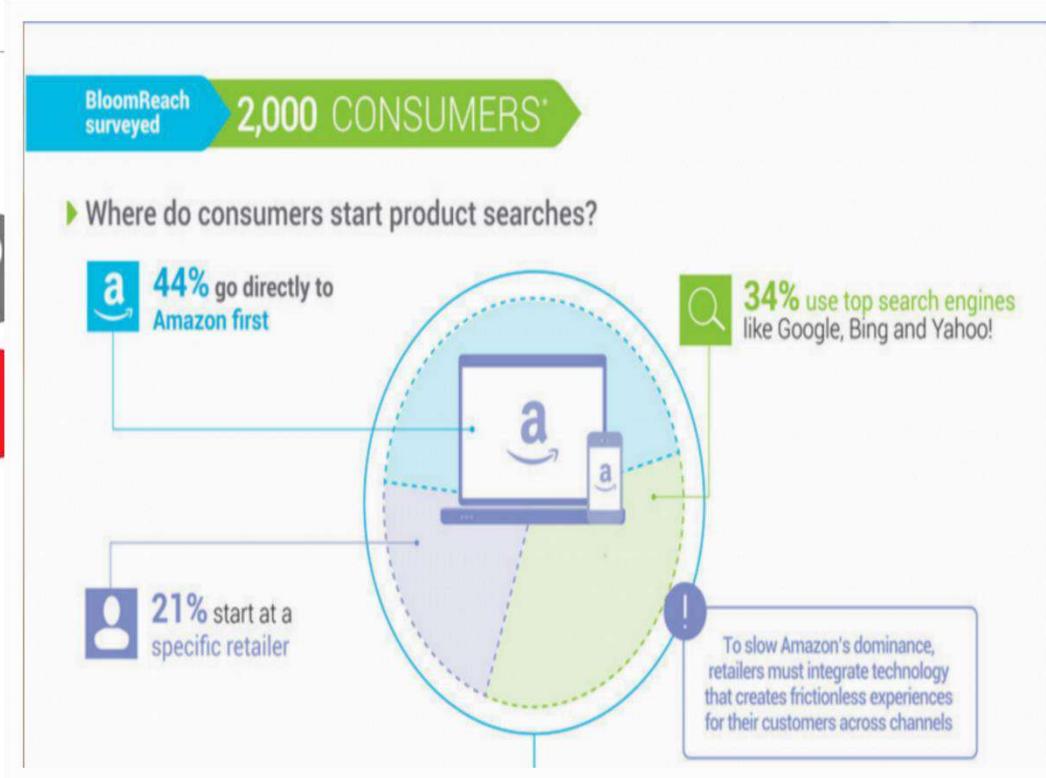


Recommendations

Exploration suggestions

Domain Specific Search

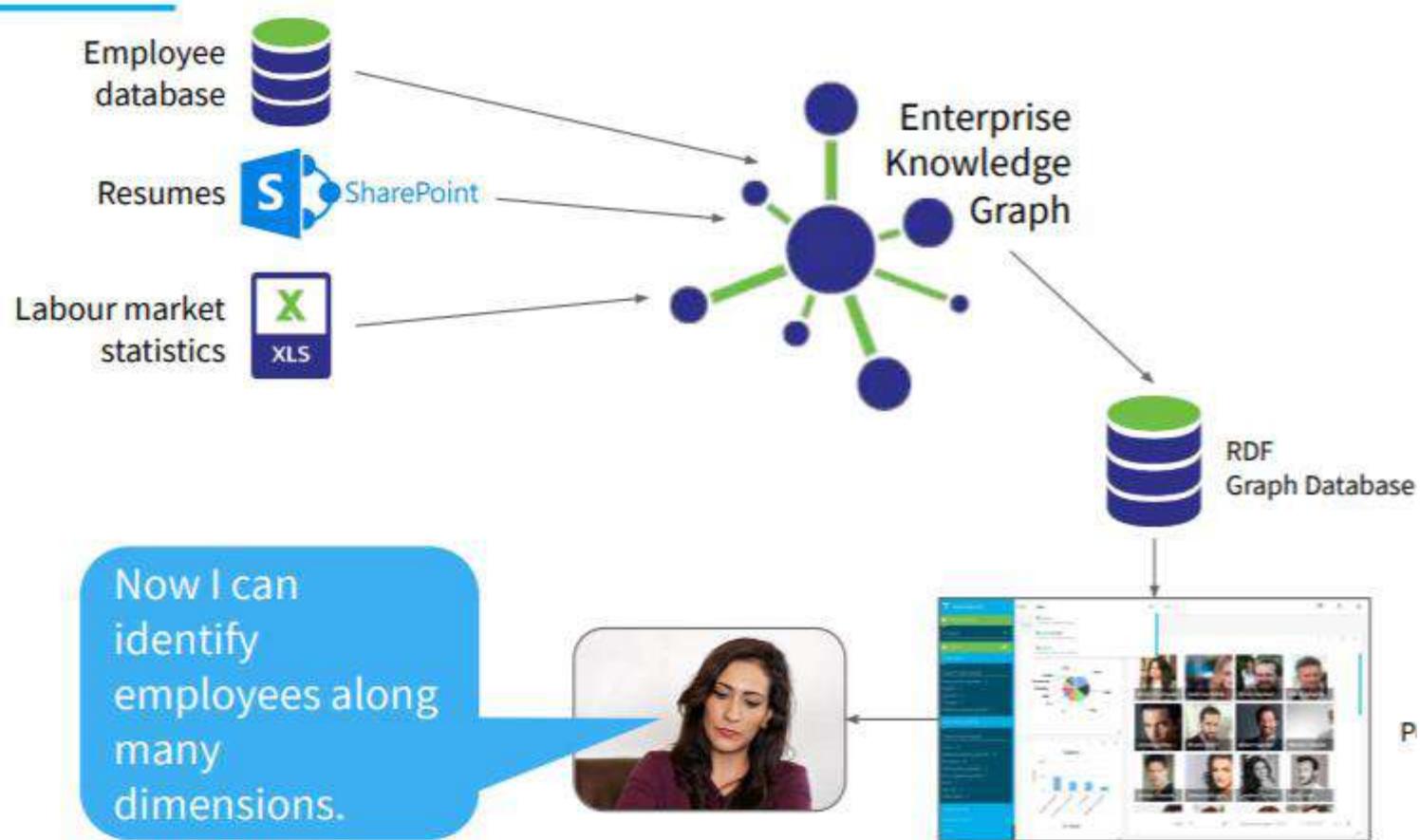
The Massive YouTube Ecosystem



source: <https://photos.prnewswire.com/prnfull/20151006/274273-INFO>

Knowledge Graphs for **Data Integration & Analytics**

Metadata enrichment, linked data, text mining, entity-centric search, agile reporting



Research in Life Sciences

As a researcher in pharmaceutical industry, I want to plan new experiments more efficiently.
I want to know what's already available.
I'm interested in former experiments where

- certain genes were tested
- under specific treatment conditions
- in a target therapeutic area
- with help from categorisation systems like 'disease hierarchies'

→ Linking Structured to Unstructured Data and to Industry Knowledge Graphs

UniProt, ChEMBL



Experiments Documentation



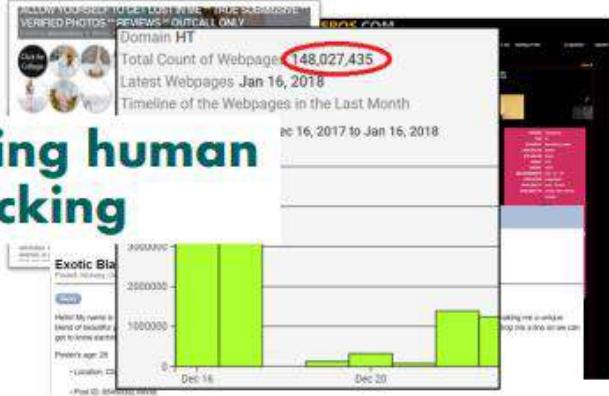
MeSH



DrugBank



Emerging applications of KG



Fighting human trafficking

Penny Stock Fraud Nets Millions
Scheme Mastermind Among Those Sentenced to Prison

Internet opens new avenue for penny stock fraud

NEW YORK — Most investors take e-mails advertising a 300 percent return on penny stocks with a grain of salt. But those Internet promotions are still irresistible for some, especially those looking to make a killing.

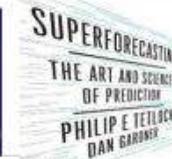
The Securities and Exchange Commission is increasingly taking legal action against individuals and companies that abuse the Internet to promote penny stocks. In one of its recent cases, involving iShares Health Co., the SEC reported a final judgment against M. Martin Lewis, for disseminating misleading information on the Internet.

State investors a total of \$1.25 million for, among other things, falsely claiming the company developed

Stopping Penny Stock Fraud

Defined by the SEC as stocks that sell below \$5 a share, penny stocks have always been considered speculative and easily manipulated. But stock market experts, seeing an increase in penny stock promotion online, say investors should be wary of

Predicting cyberattacks



Accurate geopolitical forecasting



- 1. Forecasters begin by gathering as much information as possible.
- 2. Forecasters make and develop the habit of thinking in terms of probabilities when exploring the likelihood of specific events.
- 3. Forecasting improves when individuals work in teams.
- 4. Forecasters ensure that they are regularly keeping score on their projections.
- 5. The most successful forecasters are willing to admit error and quickly change course on their projections.

Amazon Product Graph

❑ Mission: To answer any question about products and related knowledge in the world



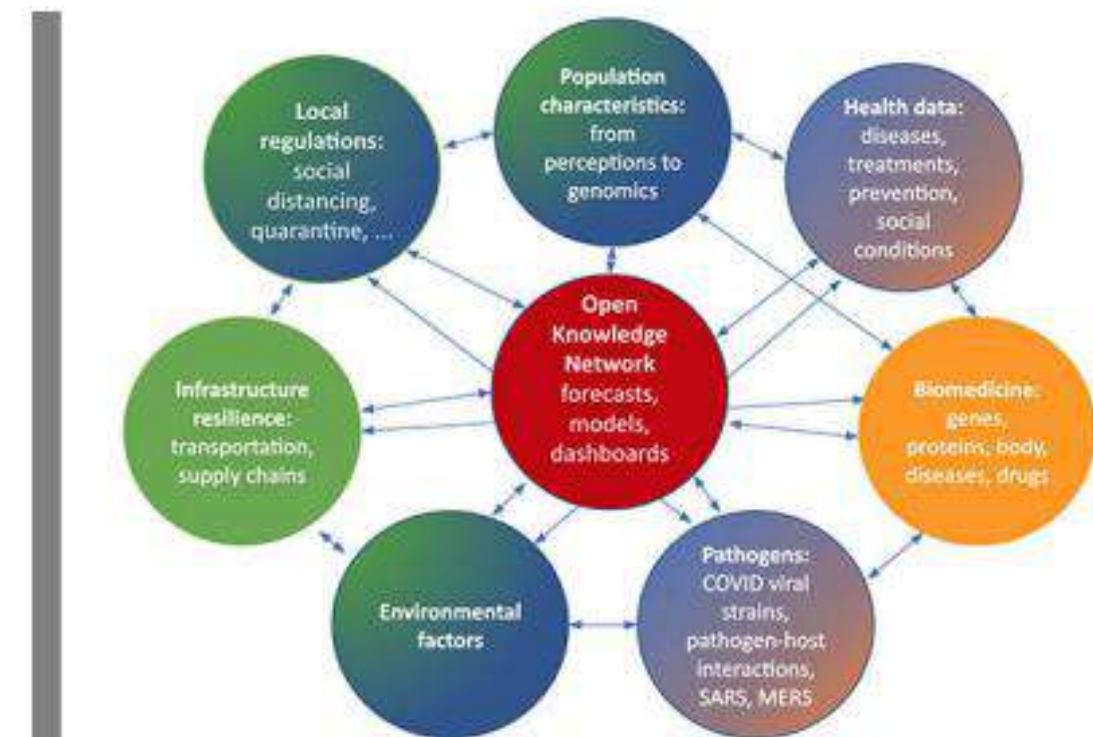
Source: Dong, Luna. Building a Broad Knowledge Graph for Products. Keynote at ICDE. 2019

June 01, 2020 | By Jan Zverina

Fighting COVID-19 with Knowledge Graphs

National Science Foundation awards funding for a semantic integration platform

"The project will be based on our knowledge graph prototype linking information about pathogens, health data, and environmental indicators and enabling cross-domain inferencing," said Peter Rose, director of SDSC's Structural Bioinformatics Laboratory and principal investigator (PI) for the project, called 'COVID-19-Net: Integrating Health, Pathogen and Environmental Data into a Knowledge Graph for Case Tracking, Analysis, and Forecasting.' "Such a graph lets researchers trace the spread of the coronavirus in different geographic conditions, focusing on specific virus strains and transmissions."



Knowledge Graph in Personal Assistant

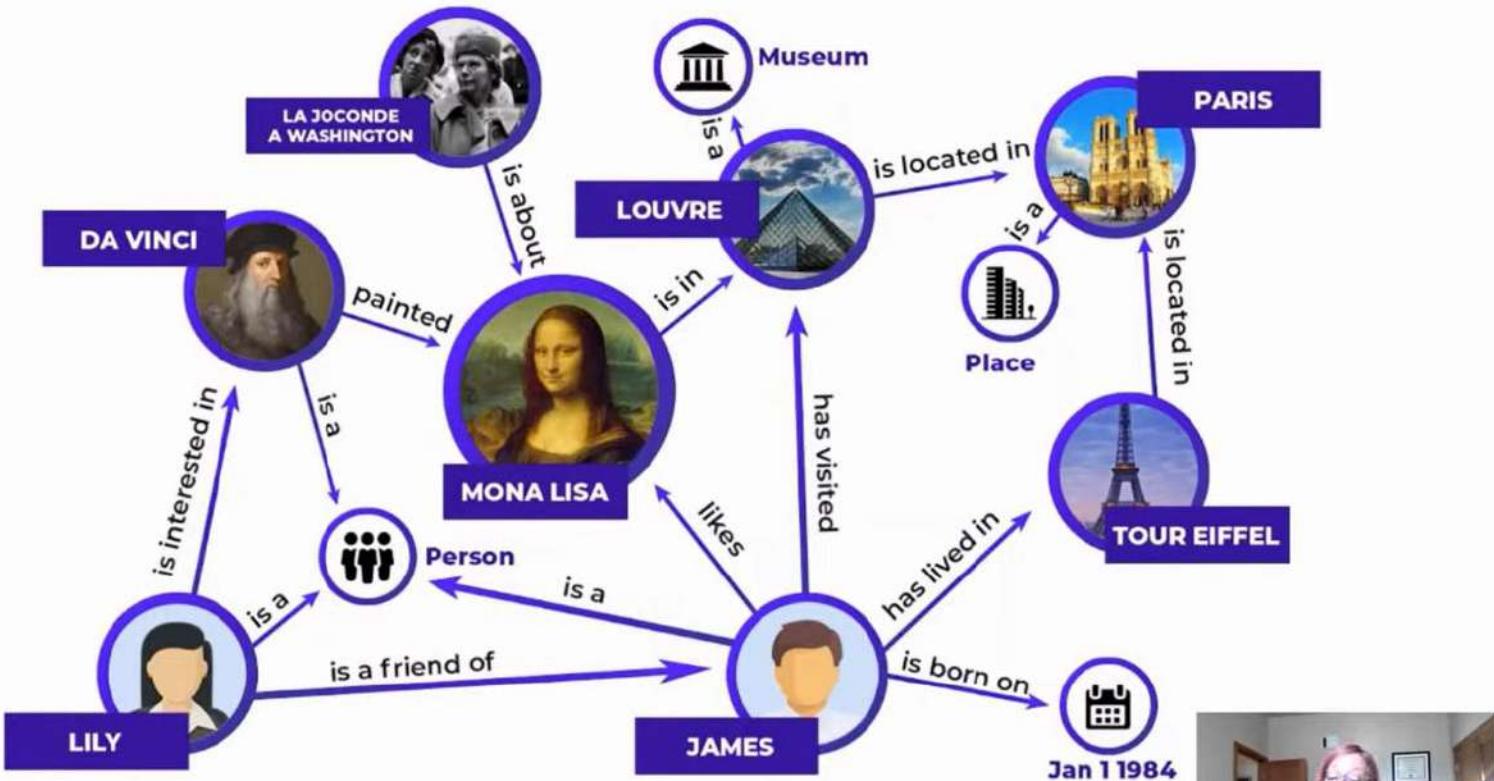


Taylor Swift > Songs

	Love Story Fearless · 2008
	Look What You Made Me... Reputation · 2017
	Shake It Off 1989 · 2014
	Delicate Reputation · 2017

Source: Dong, Luna. Building a Broad Knowledge Graph for Products. Keynote at ICDE. 2019

Recommendation system

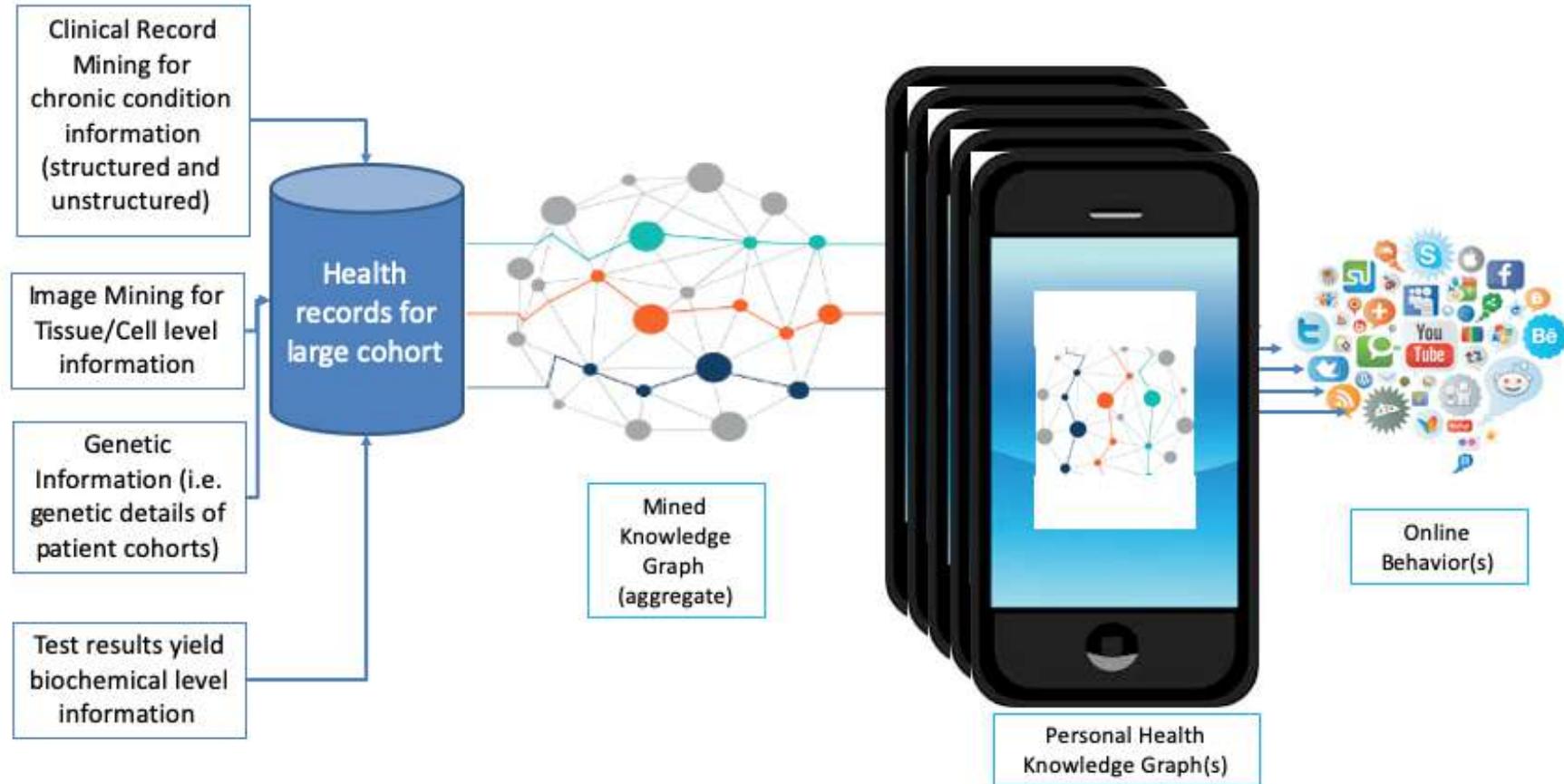


<https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/>

Neo4j, Inc. All rights reserved 2021



Health Knowledge Graph



KG in explainable AI

Explainable AI

On The Role of Knowledge Graphs in Explainable AI

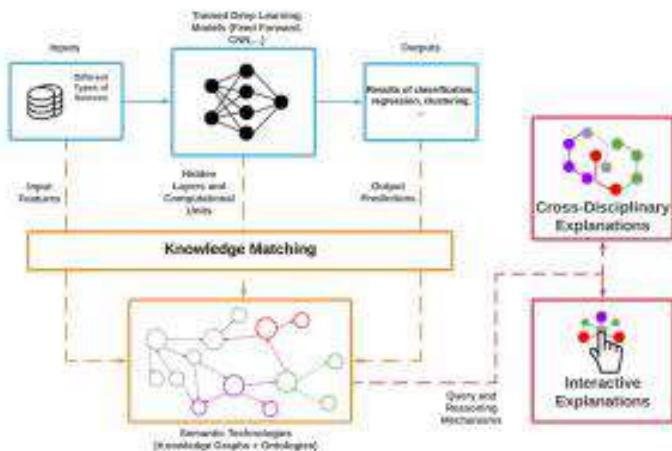
Freddy Lecue^{a,b}

^a CortAix, Thales, Montreal, Canada

E-mail: freddy.lecure@inria.fr

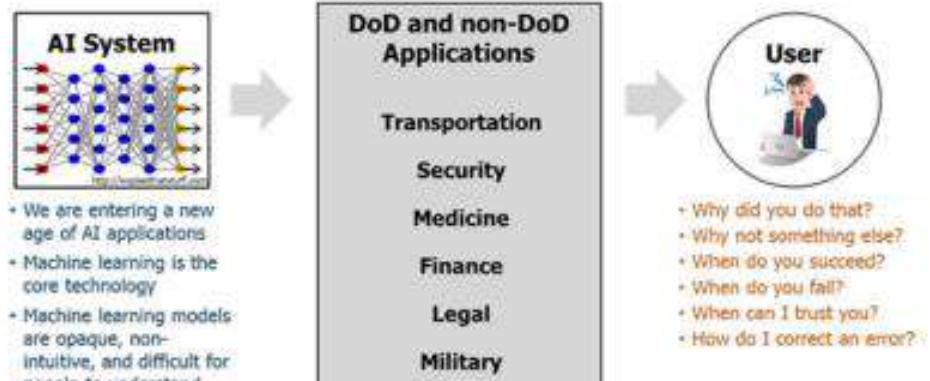
^b WIMMICS, Inria, Sophia Antipolis, France

E-mail: freddy.lecure@thalesgroup.fr



Explainable Artificial Intelligence (XAI)

Dr. Matt Turek



Source: Knowledge Graphs For eXplainable AI. On the Integration of Semantic Technologies and Symbolic Systems into Deep Learning Models for a More Comprehensible Artificial Intelligence.

<https://towardsdatascience.com/knowledge-graphs-for-explainable-ai-dcd73c5c016>

QA Results using Knowledge Graph

Google search results for "CO₂ of search versus llm models".

Large language models (LLMs) used for generative AI in search typically produce significantly more CO₂ emissions per query than a traditional, non-AI search engine query. The difference can be substantial, with estimates suggesting AI queries use anywhere from 10 to over 300 times more energy.

Carbon Footprint Comparison

- Traditional Search:** A standard search engine query produces about **0.2 grams of CO₂ equivalent (gCO₂e)**. This process is efficient, mainly involving retrieving and ranking existing web pages.
- LLM (Generative AI) Search:** The carbon footprint of an LLM query varies. It depends on the model's size, efficiency, query complexity, and the electricity grid's carbon intensity where the data centers are located.
 - Estimates for models like ChatGPT indicated a much higher footprint, with one study showing it produces roughly 340 times more CO₂ emissions than a standard Google search per query.

Certain AI prompts generate 50x more CO₂ than others

18 Jun 2025 — On average, reasoning models generated 543.5 tokens per question, compar...

Does ChatGPT use 10x more energy than a standard Google search?

It uses 10x more energy than a Google search!

Measuring the environmental impact of AI inference - Google Cloud

20 Aug 2025 — As more users use AI systems, the importance of inference efficiency rises....

Challenges in QA

Multilingualism

- refers to the notion of being able to handle questions and answers in multiple languages.
- Open Knowledge Graphs such as DBpedia and Wikidata have mappings of the data in different languages.
- Such mappings make a precious resource for developing multilingual QAS.
- use multilingual lexicon Ontologies for answering multilingual questions

Complex Questions and Queries:

- Can benefit from pre-built linguistic representations of concepts and relations in Ontologies, determining the core elements and building queries around it, even from multiple sentences
- handle such questions by inferring such temporal and spatial relationships based on the existing knowledge

Bots and Personal Assistants

- Bots and Intelligent Personal Assistants access and aggregate information on behalf of human users.
- First, this requires that these bots understand human language, written or spoken.
- Second, the descriptions of available resources no longer need to be provided to human users directly but must be enriched by semantic annotations to provide machine processability and understandability by bots.
- Both require results from Artificial Intelligence such as Natural Language Processing and Semantic Web Technology facilitating Knowledge Graphs.

Real world use of KG



KG in Conversational AI – Restaurant Booking



KG Solution

- A dialogue with an Amazon Echo or Google Home quite often ends in “Sorry, I do not know.” due to the lack of domain-specific knowledge these devices have
- To support the chatbot and voice assistant, need to design, implement, and deploy a knowledge-centered solution

KG Solution

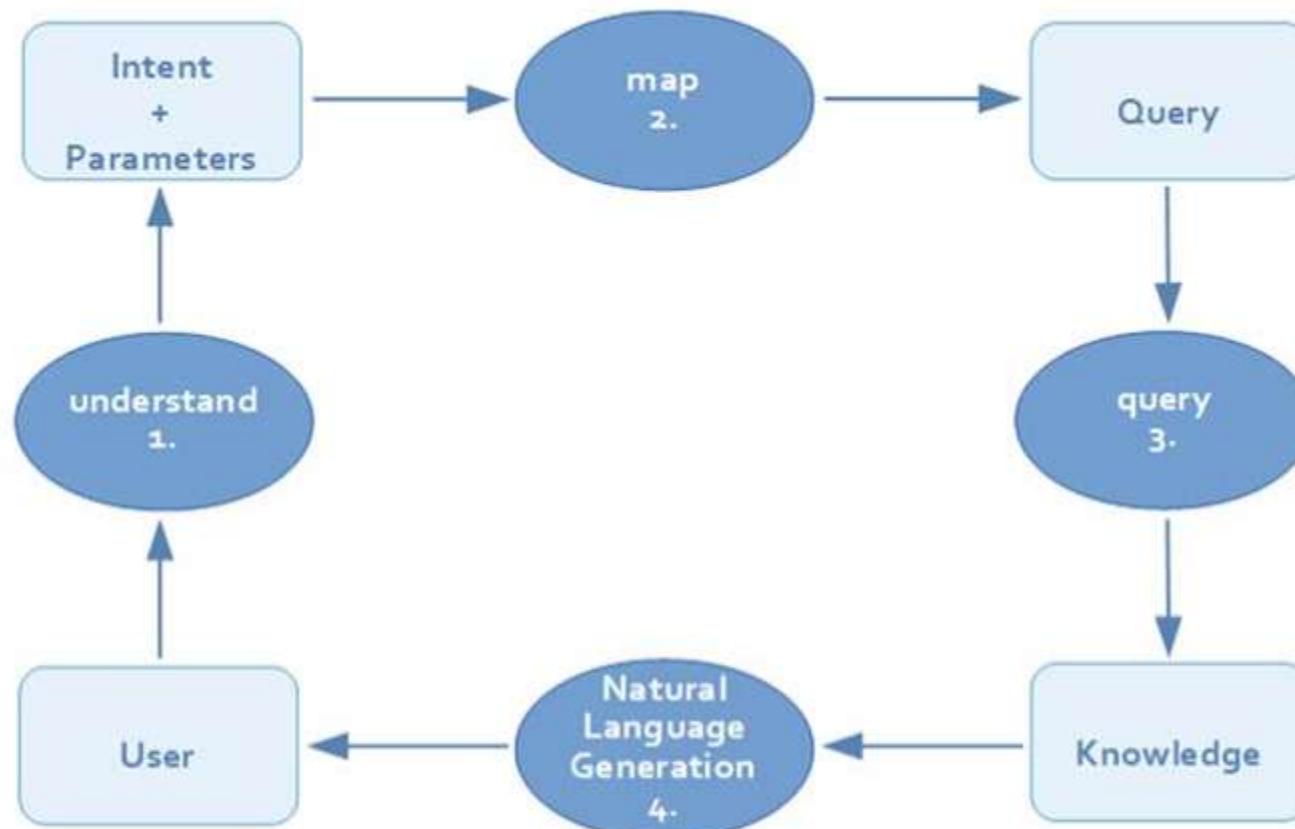
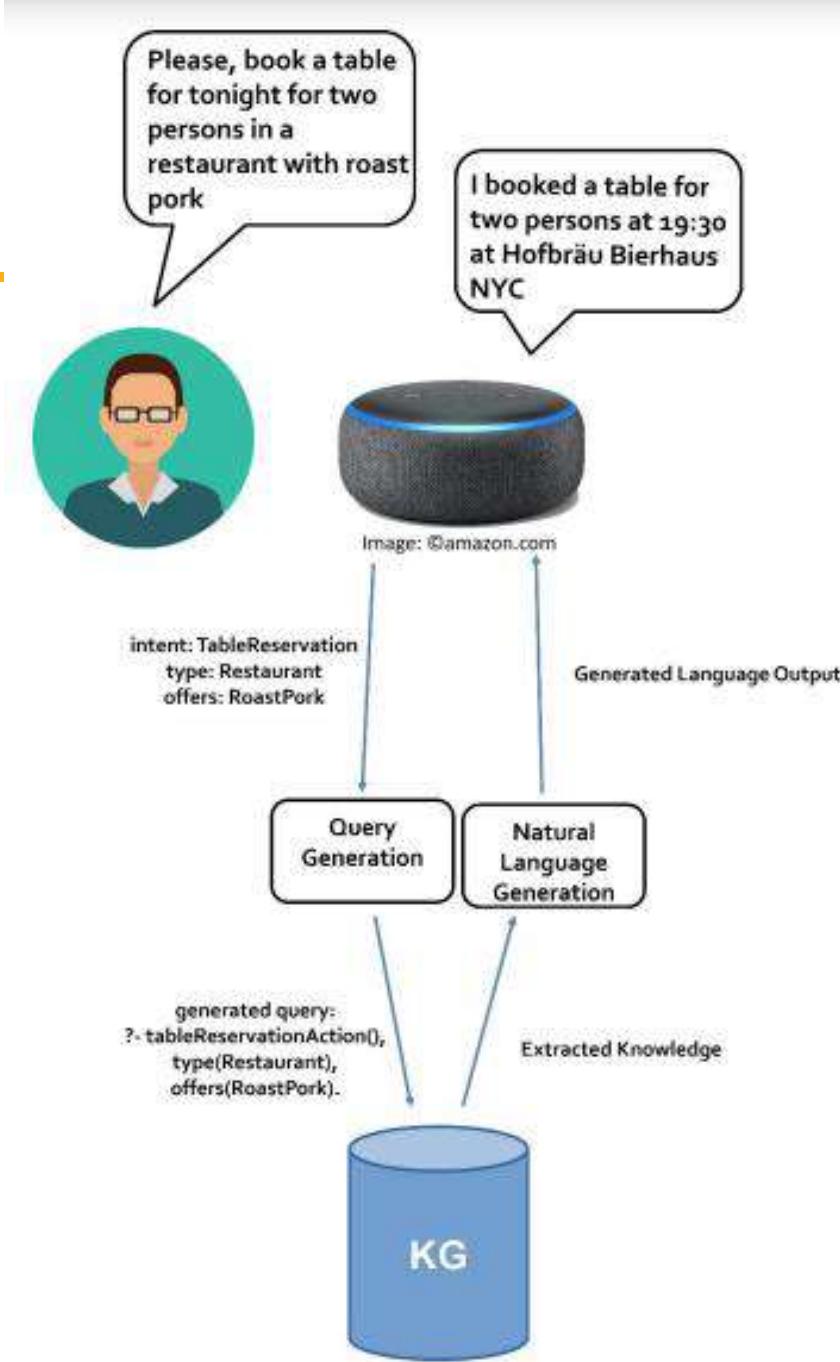


Fig. 4.2. The inner process of a Knowledge-centered Chatbots and Voice Assistants²²⁰

KG Solution

- Natural language input of a user, in written or spoken form, undergoes a natural language understanding step (understand 1.), in which the user intent, together with parameters, are identified.
- The intent needs then to be resolved to an action that typically translates in a set of queries (map 2.)
- This is then be executed (query 3.) against large volumes of heterogeneous, distributed, dynamic, and potentially (i.e., nearly for sure) inconsistent statements in order to identify the relevant knowledge parts
- Generate the user answer in natural language
- Chatbots and voice assistants are fed with knowledge from the Knowledge Graph in order to better answer human users' natural language queries and engage in meaningful dialogues with them



- Intent Table Reservation, for entities of type Restaurant can be generated.
- Restaurants, and in general, organizations can be connected in the Knowledge Graphs to other entities of type Offer. (e.g., Roast Pork offers)
- Generate the rules that restrict the view/access to the Knowledge Graph depending on the use cases.
- Such rules, together with the intent and parameters extracted by the NLU, are used to generate the queries to be executed against the Knowledge Graph
- Knowledge Graph can be used to generate templates for the answers, the textual answers or follow up questions to run the dialogues.

Case Study: Tirol Knowledge Graph Tourist Domain



Entity	Count
schema: Thing	453,841,147
schema: CreativeWork	175,787,490
schema: MediaObject	175,746,110
http://purl.org/dc/dcmitype/Image	175,735,868
schema: ImageObject	175,735,868
schema: Intangible	172,124,244
schema: StructuredValue	155,482,666
schema: Place	60,996,190
schema: ContactPoint	53,155,166
schema: PostalAddress	51,706,023

Table 2.5. Top 10 entities used in the TKG.

Use of TCG Knowledge Graph

Dialogue-driven room booking:

- Amongst the crawled websites, many are customers of the internet booking engine provider Easybooking
- Developed an Alexa skill that enables voice-driven booking of Easybooking hotels through the TKG.
- If the user asks for a certain hotel, it sends a request to a webhook.
- The result, a list of available hotel offers, is sent back and read to the user.
- The list also contains annotated API descriptions for the booking API.
- If the user decides on an offer, a booking can be executed through a voice command

Use of TCG Knowledge Graph

Showcase dialog system:

- built two dialog systems that fetch their data from the graph.
- One answers generically to touristic topics like hiking or opening hours.
- Other one goes one step further and conducts generic dialogs solely based on data taken from the Knowledge Graph.

Time series analysis of prices in touristic regions:

- prices of offers, if available, are stored permanently, a time series analysis can be conducted.
- price development of two touristic regions can be compared

Time series analysis work perfect with Knowledge Graphs and is a promising application of them in tourism.

Tourist Chatbot Use Case

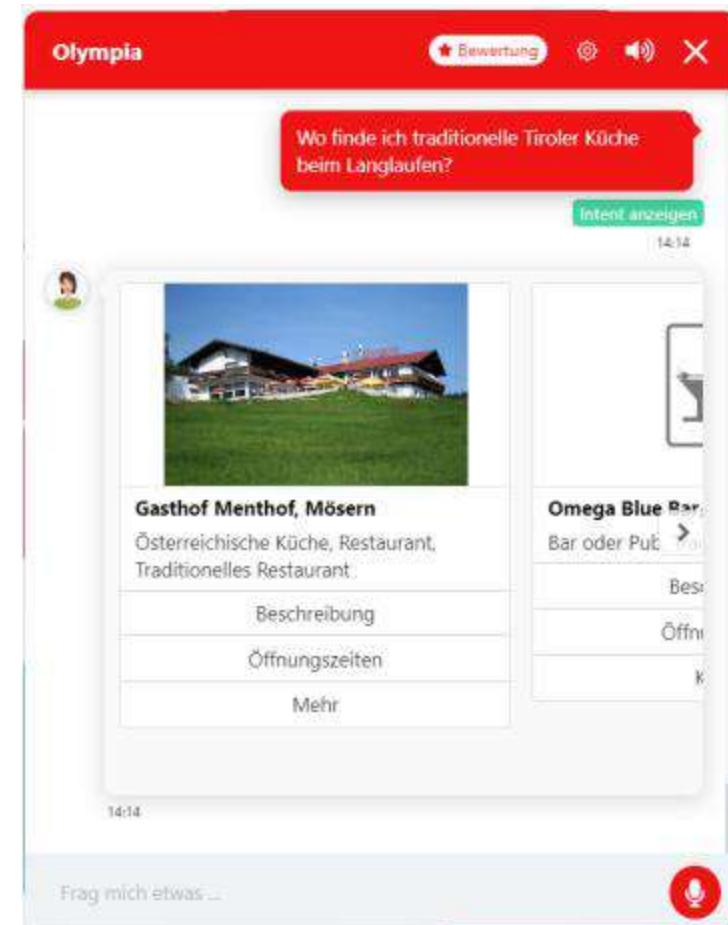
- Planning phase hotels and Destination Management Organizations (DMOs) can provide information about the hotel and the region, the surroundings, and weather condition
 - Booking phase, from booking the hotel and transport to buying connected services, e.g., ski tickets, all becomes much simpler and efficient by using natural language.
 - Experiences phase, Chatbots and Voice Assistants can also announce special offers or events.
 - All requested information and processes are available 24/7/365 and instantly
-

Tourist Chatbot Use Case

- Need structures that encode the knowledge about the tourism domain, in terms of entities and relations between them, in a machine processable form.
- Knowledge Graphs are such structure providing the technical means to integrate various heterogeneous touristic information sources
- Integrates multiple sources of content, data, and services from various providers, both
 - Closed sources: feratel, General Solutions, intermaps, Outdooractive etc
 - Open sources: DBpedia, GeoNames, OpenStreetMap, and Wikidata

Tourist Chatbot Use Case

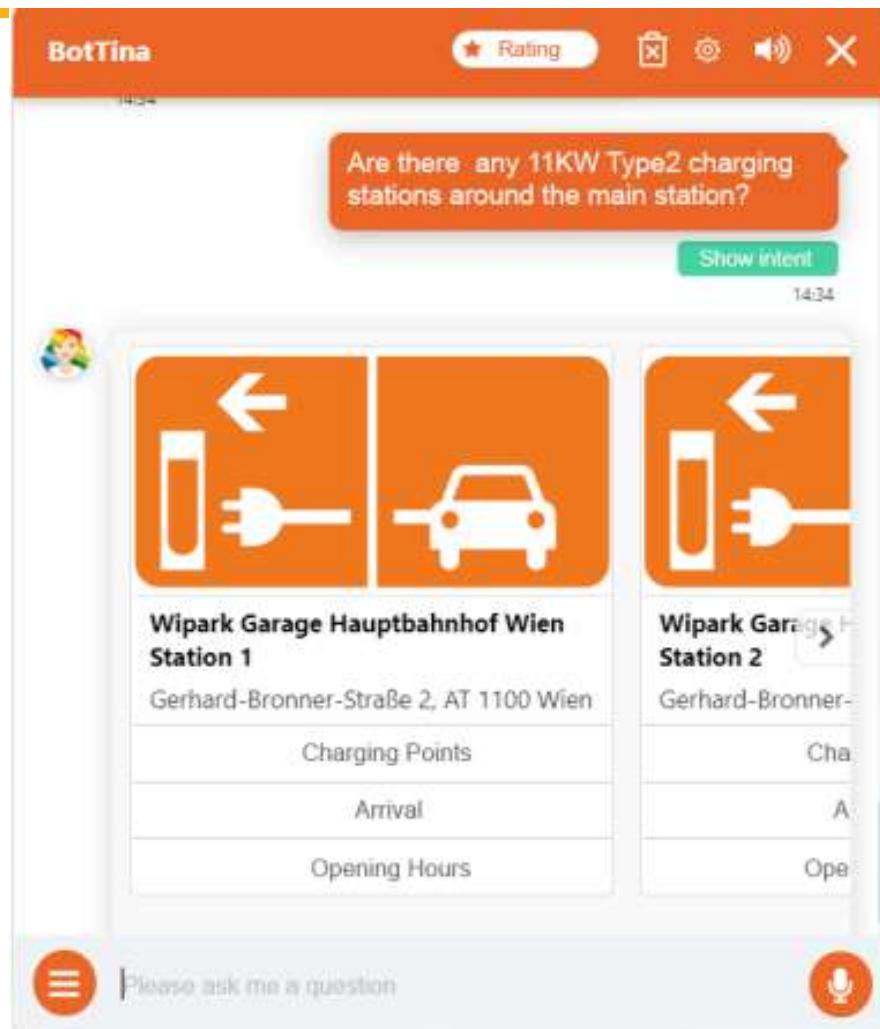
"Where can I have a traditional Tyrolean food when going cross country skiing?"



Tourist Chatbot Use Case

- Need to integrate data from multiple heterogeneous static and dynamic sources for which we need to track provenance and maintain one shared evolving schema
- Ensured a level of quality of the knowledge.
- Optimize conversational interfaces based on Knowledge Graphs by providing a rich intent and entity management (e.g., automated NLU training), question answering over the Knowledge Graph
- Supporting advanced dialogues such as guiding a user through actions or recommendations or follow up conversations

Energy Use Cases



Wien Energy Knowledge Graph

- Includes information about the electric vehicle charging stations in the city of Vienna.
- Extended the schema.org vocabulary with additional types and properties.
- Can model information such as the different types of plugs available at an EV charging station, its voltage and power, its address and geo-coordinates, as well as accepted payment methods.
- KG populated with data received from Wien Energie EV charging station API.
- Static data and dynamic data considered. Ex no. of available plugs, waiting times and offers
- Using the Knowledge Graph, the chatbot can answer questions such as: "Are there any 11KW Type 2 charging stations around the main station?"

Wien Energy Knowledge Graph

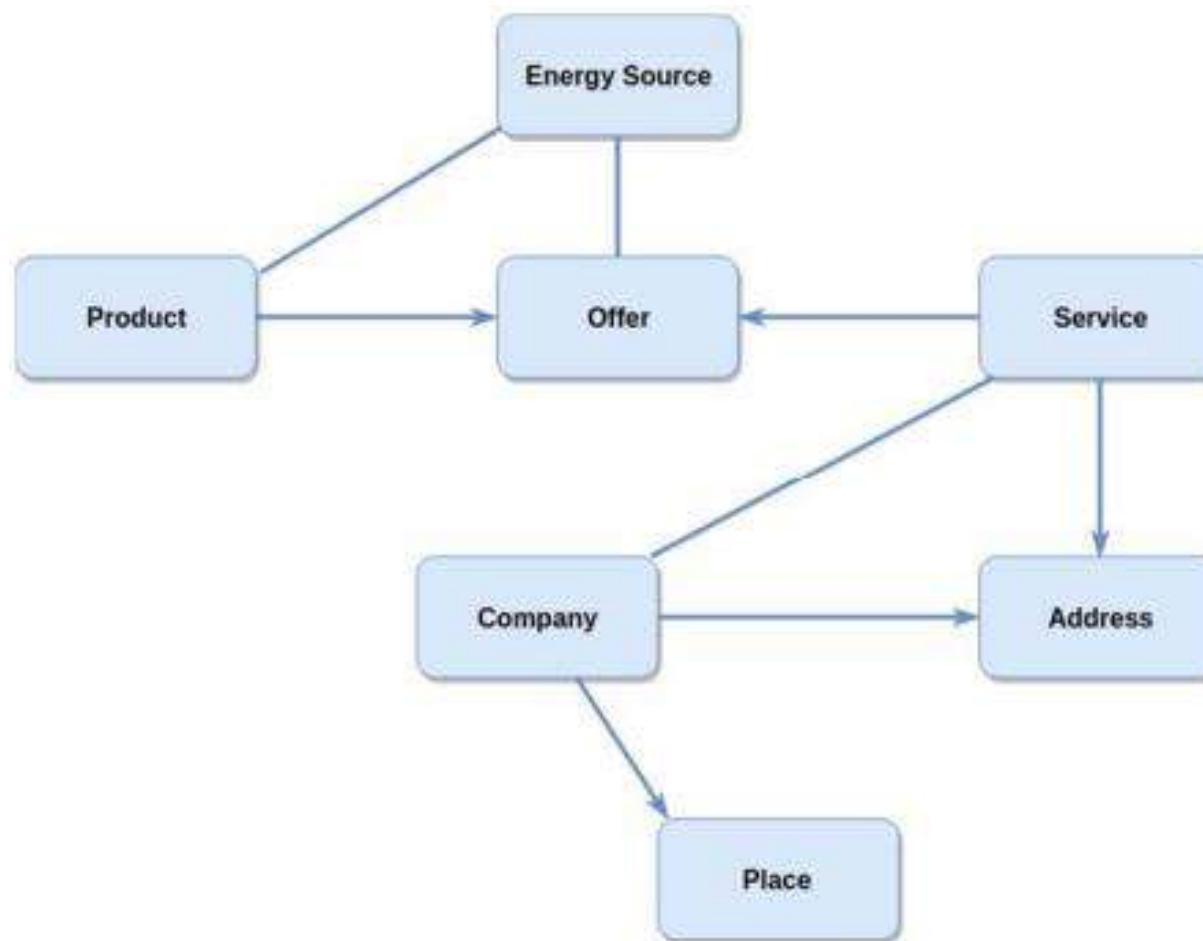
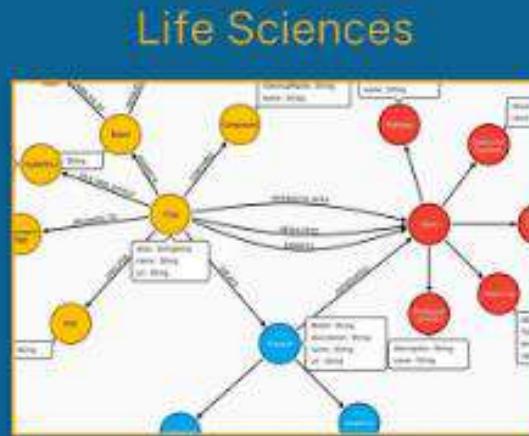


Fig. 4.8. Energienetze Steiermark domain model top-level types

Knowledge Graph use in different domains



Finance



Life Sciences

Marketing & Recommendations



How many flagged accounts are in the applicant's network **4+ hops out**?

How many **login / account variables in common**?

Add these metrics to your approval process

What **completes the connections** from genes to diseases to targets?

What genes can be reached **4+ hops out** from a known drug target?

What **mechanisms in common** are there between two drugs?

Collaborative filtering: users who bought X, also bought Y

What items make you more likely to buy additional items **in subsequent transactions**?

Traverse hierarchies - what items are similar **4+ hops out**?

Other Verticals: Education

The University of Innsbruck pilot

- Constructing a Knowledge Graph about study programs offered
- Used to improve conversations related to the different study programs, subjects, and examinations.
- Based on the entities and relations modeled in the Knowledge Graph, dynamic intents are created to query the content from the Knowledge Graph
- Answer questions such as “Which study programs are offered by the Faculty of Biology?”, or ‘How many semesters do I need to study for the Master’s Programme Business Law?’

Other Verticals: Finance

Wiener Börse pilot

- focuses on modeling knowledge about companies, equities, bonds, indexes available on the Vienna Stock Exchange
- Accessing data in real time
- High-frequency data are essential to enable the Chatbot and Intelligent Personal Assistant to provide accurate, up-to-date information
- Chatbot answers questions such as "What is the stock price of X?", "Which stocks have lost the most?"

Other Verticals: Retail

- Radically change the e-marketing and ecommerce processes.
 - Customers will be able to search for products that fit their needs and then may buy these products, by directly engaging in natural dialogues with the retail Chatbots and Intelligent Personal Assistants.
 - Access to clothing and footwear products, models and integrates information about product catalogs, product stocks, shops, and connected services into a retail Knowledge Graph
 - Answering complex questions about products or e-commerce processes
-

Other Verticals: Healthcare

- **Integrating medical knowledge:** Healthcare data is the most diverse dataset of all. It is an endless pile of rich information from biomedical, genomic, and protein to molecular data.
 - **Mapping relationships between medical data:** By mapping the correlations between heterogeneous medical data
 - **Drug discovery:** The potential to cure life-threatening diseases
 - **Patient Care:** Monitoring patient information and predicting risks and anomalies in their data.
-

Other Verticals: Media & Entertainment

- **Social Graphs:** Social media platforms use knowledge graphs to build social graphs of their users to help them maintain and view their users in the context of their connections with other users.
- **Recommender Systems:** Predictive knowledge graphs enable media platforms to recommend the most relevant items
- **Content Targeting:** Using the structural relations between a user and its likes and dislikes through a behavioral knowledge graph, media platforms infer user preferences

Other Verticals: Cybersecurity

- **Mapping historical cyber attacks:** Historical cyber attack data is filled with insights that allow security professionals to detect patterns
- **Filling security gaps:** A network of cybersecurity data enables organizations to holistically view their security structure to find any gaps in their security implementations.

References

- Berners-Lee, Hendler, Lassila (May 17, 2001). The Semantic Web. Scientific American Magazine. @ <http://bit.ly/2z2LZIE>
- Berners-Lee. A roadmap to the Semantic Web. Tim Berners-Lee's Design Issues @ <http://bit.ly/2z29fGP>
- Berners-Lee. Linked Data. Tim Berners-Lee's Design Issues @ <http://bit.ly/21MR3Zt>
- Heath, Bizer. Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology @ <http://bit.ly/2xIWKou>
- <https://wiki.dbpedia.org/services-resources/ontology>
- <https://www.emse.fr/~zimmermann/Teaching/SemWeb/semwebintro.pdf>
- <https://www.youtube.com/watch?v=C9M7N979pIU>
- <https://enterprise-knowledge.com/whats-the-difference-between-an-ontology-and-a-knowledge-graph/>
- <https://www.w3.org/Consortium/facts>
- <https://pages.semanticscholar.org/coronavirus-research>

References

- <https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/>
- [Google Knowledge Graph](#)
- [Building Knowledge graph](#)
- <https://www.youtube.com/watch?v=lUc0woFX16M>
- https://www2.slideshare.net/jeffpan_sw/linked-data-and-knowledge-graphs-constructing-and-understanding-knowledge-graphs?qid=ab5124c7-a15a-4f77-9fc9-914feac00a6a&v=&b=&from_search=1
- https://www2.slideshare.net/phaase/getting-started-with-knowledge-graphs?from_action=save
- <https://lod-cloud.net/>
- <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>
- https://www.slideshare.net/slideshow/embed_code/key/dJaP9LmSd1uPXq
- <https://www.youtube.com/watch?v=hILYM9oBI0M>
- <https://www.youtube.com/watch?v=sYejXeuZaeY>



Natural Language Processing Applications



BITS Pilani
Pilani Campus

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



Session 7: Knowledge Graph Applications and Retrieval Augmented Generation (RAG)

These slides are prepared by the instructor, with grateful acknowledgement of Prof. Philip Cimiano , Johanna Völker , Paul Buitelaar and many others who made their course materials freely available online.

Session Content

- Knowledge Graph and LLM
- Vector RAG
- Graph RAG
- Agentic RAG
- Tools and Technologies used in RAG
- Multimodal RAG
- Real World Case studies
- How KG power Agentic AI
- Implementation Demo

The Power and Limits of LLMs

What LLMs Are Great At:

- Generating fluent, human-like text.
- Summarization, translation, and content creation.
- Answering questions based on their vast training data.

The Inherent Limitations:

- **Knowledge Cutoff:** Their knowledge is frozen at the time of training (e.g., GPT-4's knowledge largely ends in early 2023). They don't know about recent events or data.
- **Hallucinations:** When they don't know an answer, they often "make up" plausible but incorrect information.
- **Lack of Specificity:** They lack deep knowledge of proprietary, domain-specific, or private information (e.g., our company's internal policies, our latest product specs).

Challenges using LLM in Enterprise

Consumer	Enterprise
✗ Trained on Public Data	✓ Proprietary Data
✗ Probabilistic Answers/Hallucinations	✓ Accurate Answers
✗ Training Data Sources	✓ Data Provenance
✗ Public Access	✓ Controlled Access
✗ Prompts and Data recorded/re-used?	✓ Proprietary IP
✗ Legal and Regulatory Conundrums	✓ Fully Compliant

Integration Strategies

- 1. KG-Augmented Retrieval:** Using KGs to retrieve relevant information to inform LLM responses.
 - 2. KG-Aware Generation:** Incorporating KG facts directly into LLM prompts.
 - 3. KG Construction via LLMs:** Employing LLMs to extract entities and relationships from unstructured text to build KGs.
 - 4. Joint Reasoning:** Combining graph traversal with LLM-based generation for complex queries
-

The Strategy: Accessing External Knowledge Bases

- Ground the LLM in factual, up-to-date, and relevant information.

The Method: Retrieval-Augmented Generation (RAG)

- Don't retrain the model (which is expensive and slow).
- Instead, give the LLM access to a live, external knowledge base at the moment it needs to answer a question.
- This knowledge base can be anything:
 - Internal company documents (PDFs, Confluence, SharePoint)
 - Product manuals and technical specifications
 - Customer support ticket histories
 - Databases with real-time information

The Strategy: Accessing External Knowledge Bases

- Ground the LLM in factual, up-to-date, and relevant information.

The Method: Retrieval-Augmented Generation (RAG)

- Don't retrain the model (which is expensive and slow).
- Instead, give the LLM access to a live, external knowledge base at the moment it needs to answer a question.
- This knowledge base can be anything:
 - Internal company documents (PDFs, Confluence, SharePoint)
 - Product manuals and technical specifications
 - Customer support ticket histories
 - Databases with real-time information

RAG

Definition: RAG combines information retrieval with language generation, allowing models to fetch relevant data from external sources to enhance responses.

Components:

- **Retriever:** Fetches relevant documents based on the query.
- **Generator:** Generates responses using both the query and retrieved documents.

Benefits:

- Access to up-to-date information.
- Improved factual accuracy
- Reduced hallucinations

How RAG Works

- 1. Retrieve:** When a user asks a question, the RAG system first searches a trusted data source (e.g., internal research, regulatory filings, market data).
 - 2. Augment:** It takes the relevant, factual information it found and adds it to the user's original prompt.
 - 3. Generate:** The LLM then generates an answer based *specifically on the provided information*, citing the sources it used.
-

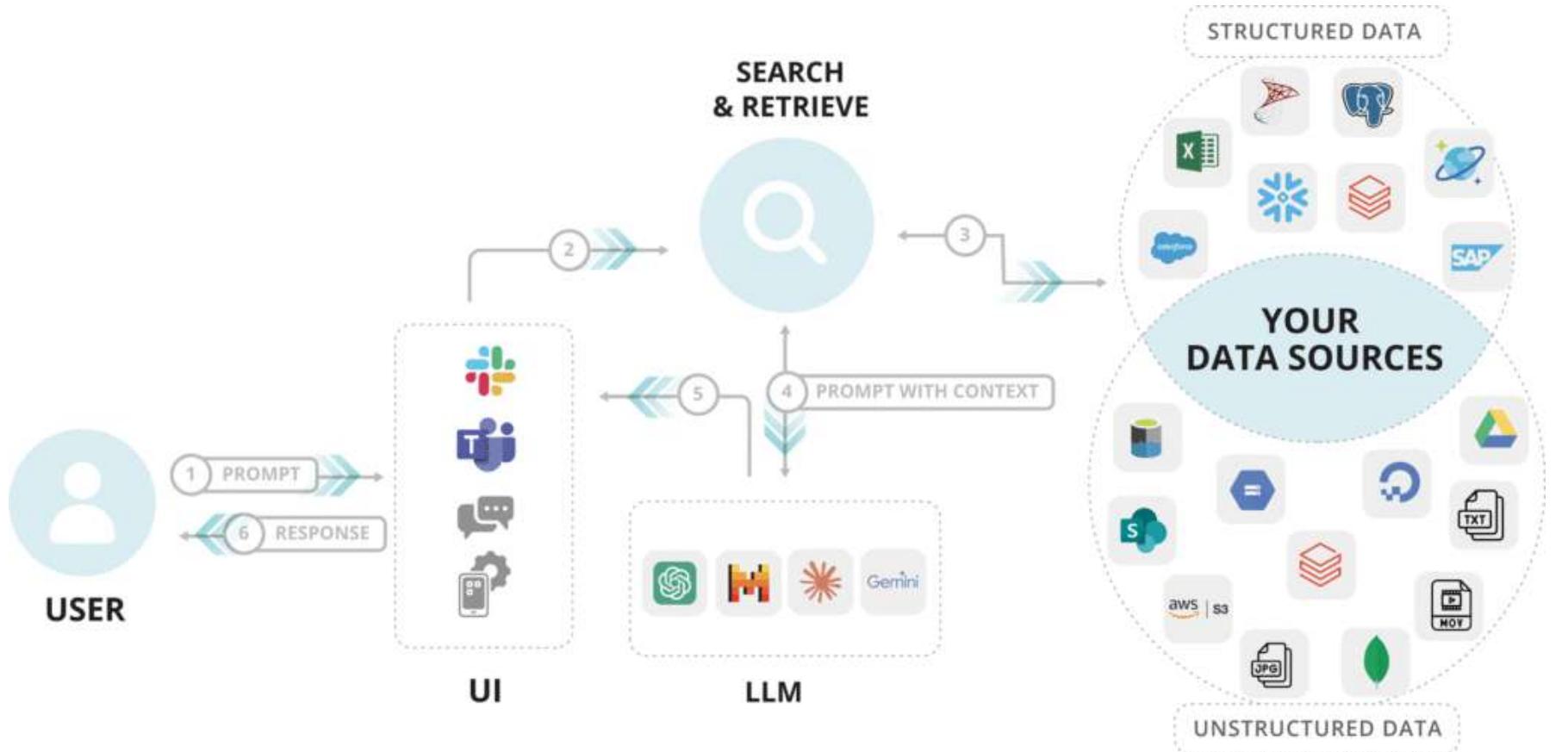
RAG Flow

- **User Query:** A user asks a question,
e.g., "What is our policy on international travel?"
 - **Semantic Search:** The system searches the external knowledge base (e.g., HR policy documents) for text chunks semantically related to the query.
 - **Context Retrieval:** It retrieves the most relevant paragraphs, such as the section on "Overseas Business Trips and Expense Approval."
 - **Prompt Enhancement:** It creates a new, detailed prompt for the LLM:
 - *"Using the following context: [insert retrieved text here], answer this question: What is our policy on international travel?"*
 - **LLM Generation:** The LLM generates a clear, concise answer based *only* on the provided context, preventing hallucination and ensuring accuracy.
-

RAG Flow

- **User Query:** A user asks a question,
e.g., "What is our policy on international travel?"
- **Semantic Search:** The system searches the external knowledge base (e.g., HR policy documents) for text chunks semantically related to the query.
- **Context Retrieval:** It retrieves the most relevant paragraphs, such as the section on "Overseas Business Trips and Expense Approval."
- **Prompt Enhancement:** It creates a new, detailed prompt for the LLM:
 - "*Using the following context: [insert retrieved text here], answer this question: What is our policy on international travel?*"
- **LLM Generation:** The LLM generates a clear, concise answer based *only* on the provided context, preventing hallucination and ensuring accuracy.

How RAG Works



<https://hatchworks.com/blog/gen-ai/rag-for-financial-services/>

Vector RAG

Ingestion: External data (documents, web pages, etc.) is processed, broken into chunks, and converted into numerical representations called **embeddings**.

Indexing: These embeddings are stored in a specialized database called a **Vector Database**.

Retrieval: When a user asks a query, it is also converted into an embedding. The vector database then searches for the most similar data chunks (based on the query embedding).

Augmentation: The original user query and the retrieved data chunks are combined into a comprehensive prompt for the LLM.

Generation: The LLM uses this augmented prompt to generate a response that is informed by the retrieved context.

Vector embedding Search

Text Embedding Vectors for Semantic Search

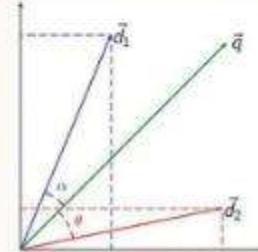
graphsun [

Given a question, find the most relevant documents based on a similarity metric (such as Cosine Similarity) between vector of the question and vectors of contents.

Moving from keyword search to similarity (semantic) search.

Q: what is a text embedding?

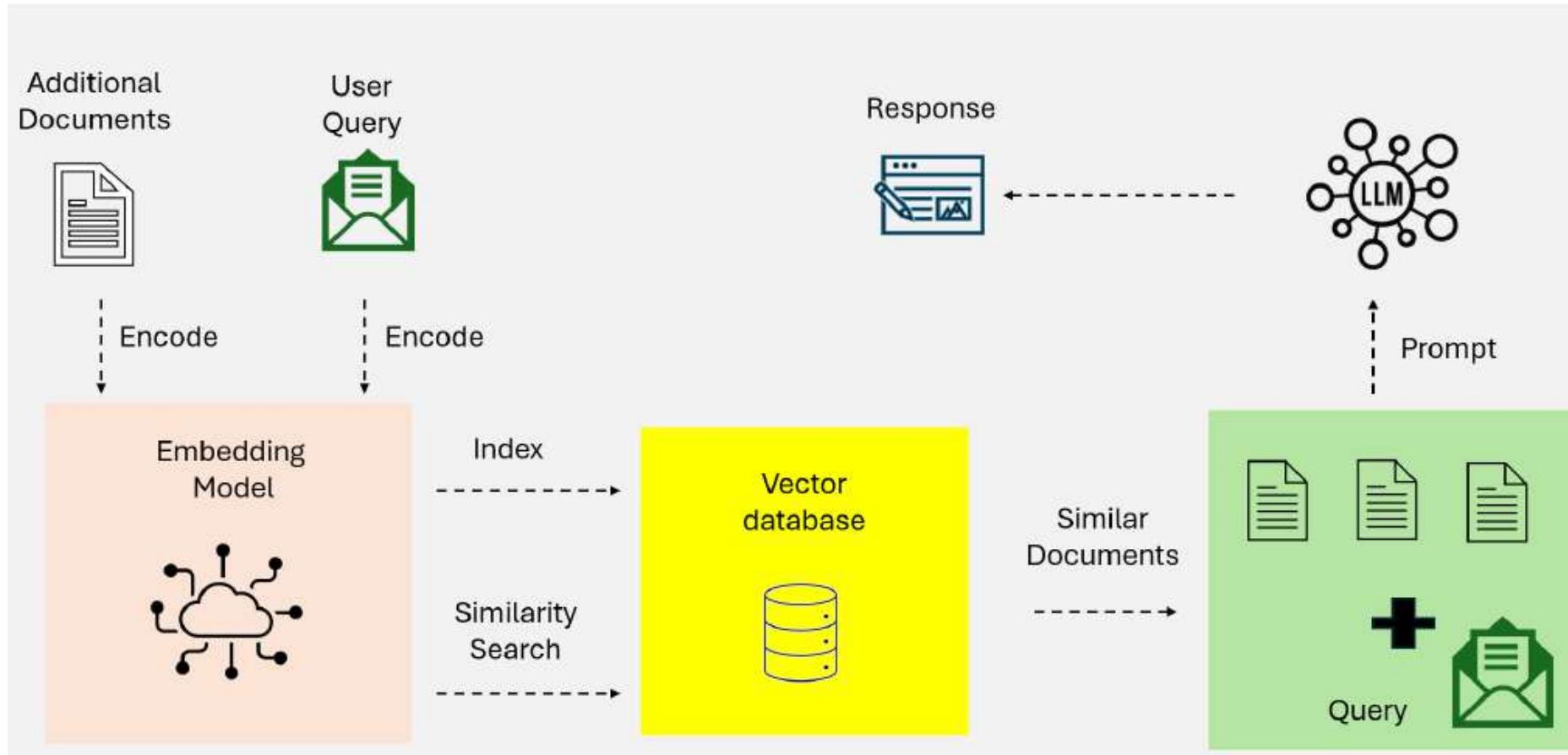
embedding [-0.019200351,0.0035306285,0.0
14364655,-0.009949144,0.00115
81815,0.017021084,-0.01998375,
-0.018986698,-0.033643346,-0.0
2841595,0.014685135,0.0141937
32,... Show all]



Top K by similarity

abstractId	similarity
456	0.923445
22	0.892114
...	...

Vector RAG



<https://arxiv.org/abs/2501.09136>

Advanced RAG

While Vector RAG is powerful, it can fall short with multi-faceted or complex questions. This has led to the development of more sophisticated RAG architectures:

Graph RAG: Leverages the relationships within structured data (knowledge graphs) for more precise and context-aware retrieval.

Agentic RAG: Introduces autonomous "agents" that can reason, plan, and use tools to improve the retrieval process.

GraphRAG

Graph RAG utilizes knowledge graphs—databases that store information as nodes (entities) and edges (relationships)—to enhance retrieval. This is ideal for domains with highly interconnected data, such as finance.

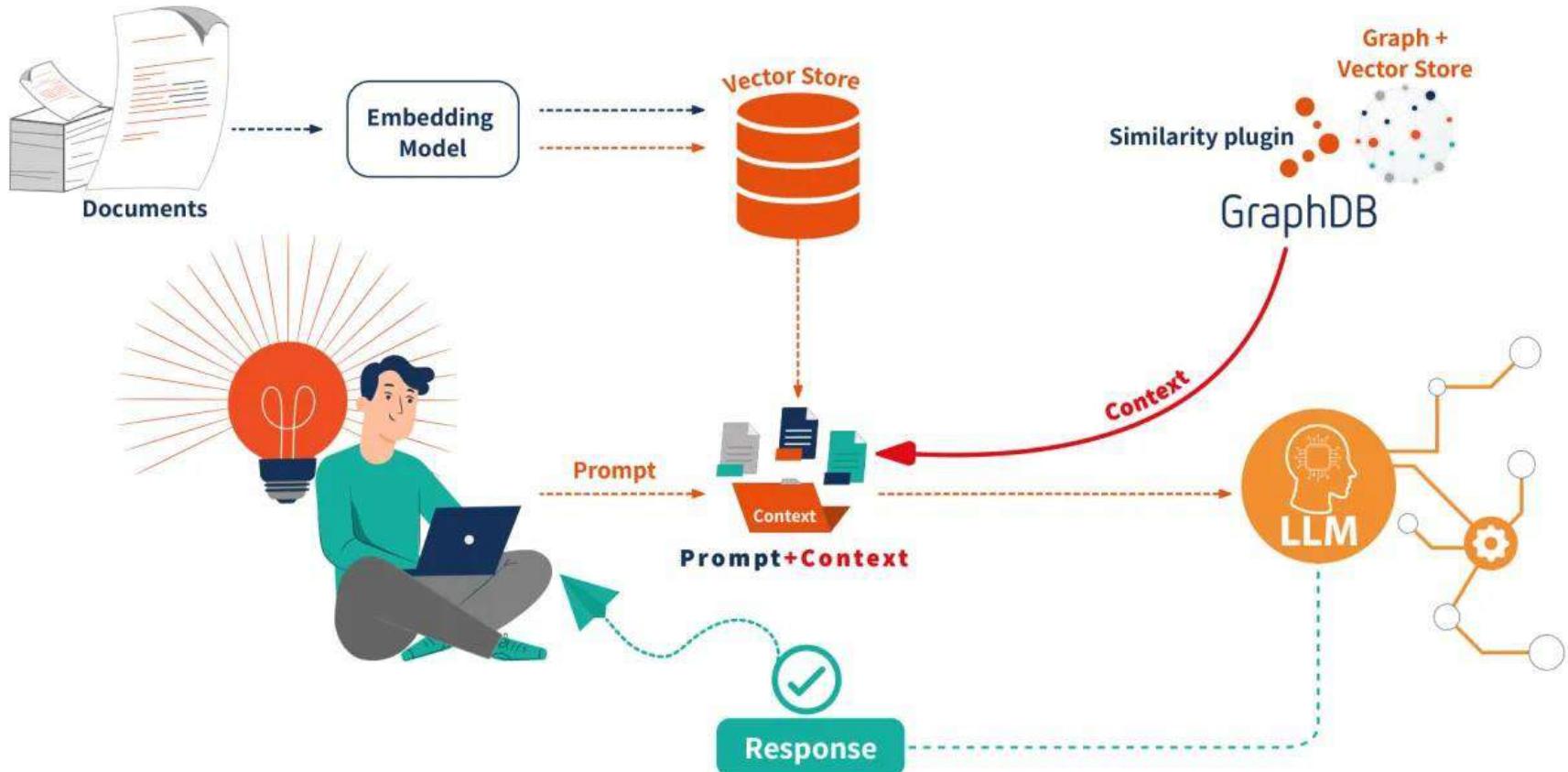
How it Works:

Entity and Relationship Extraction: Key entities and their relationships are identified in the user query.

Graph Traversal: The system navigates the knowledge graph to find relevant nodes and their connections.

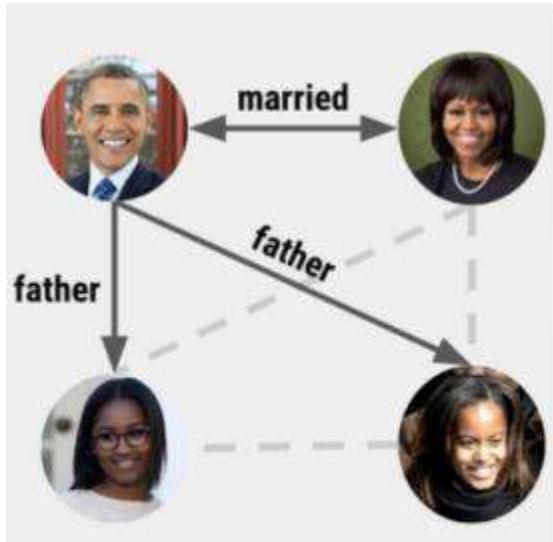
Contextualized Retrieval: Retrieves not just individual facts, but also the surrounding context and relationships.

Graph RAG



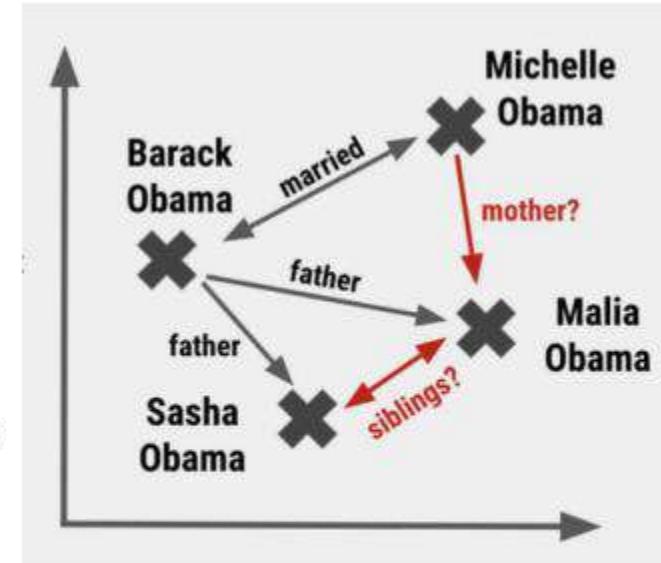
<https://www.ontotext.com/blog/matching-skills-and-candidates-with-graph-rag/>

Knowledge Graph Embedding

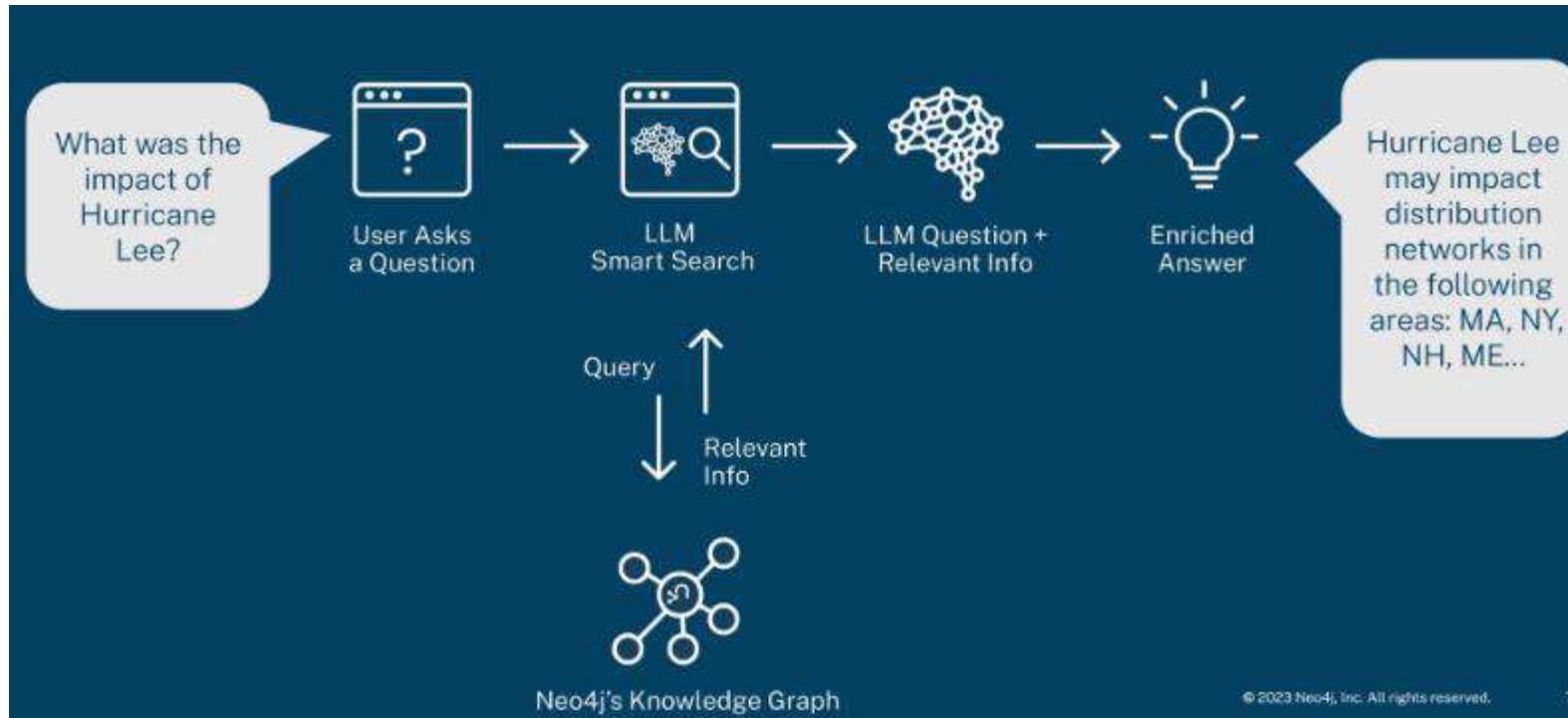


Knowledge graph embeddings:

- TransE, H...
- Neural tensor networks
- Graph convolutional networks (or their variants)
- Matrix factorization
- ...



Knowledge Graph use to ground LLM



© 2023 Neo4j, Inc. All rights reserved.

7

Knowledge Graph use to ground LLM



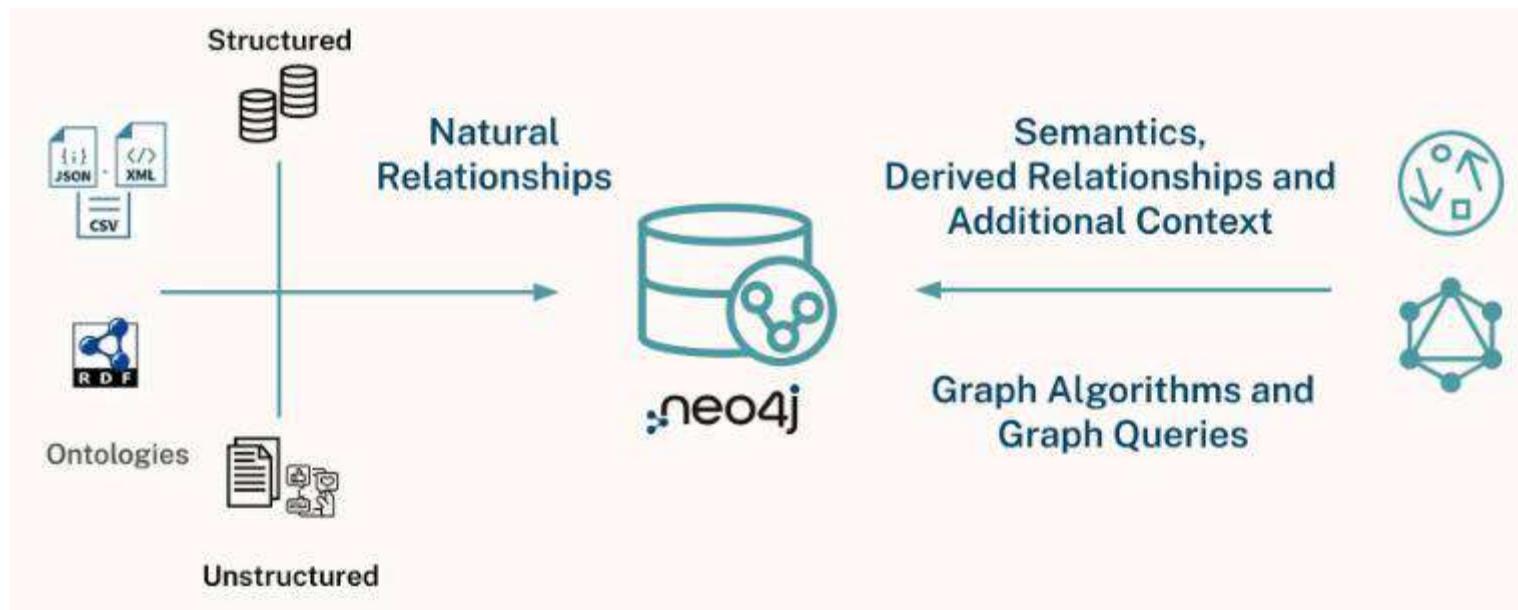
Context rich,
connected view of
your data that
enables easier
decision making

Enhance your data
with **graph data
science**, text
embeddings, and
additional derived
context

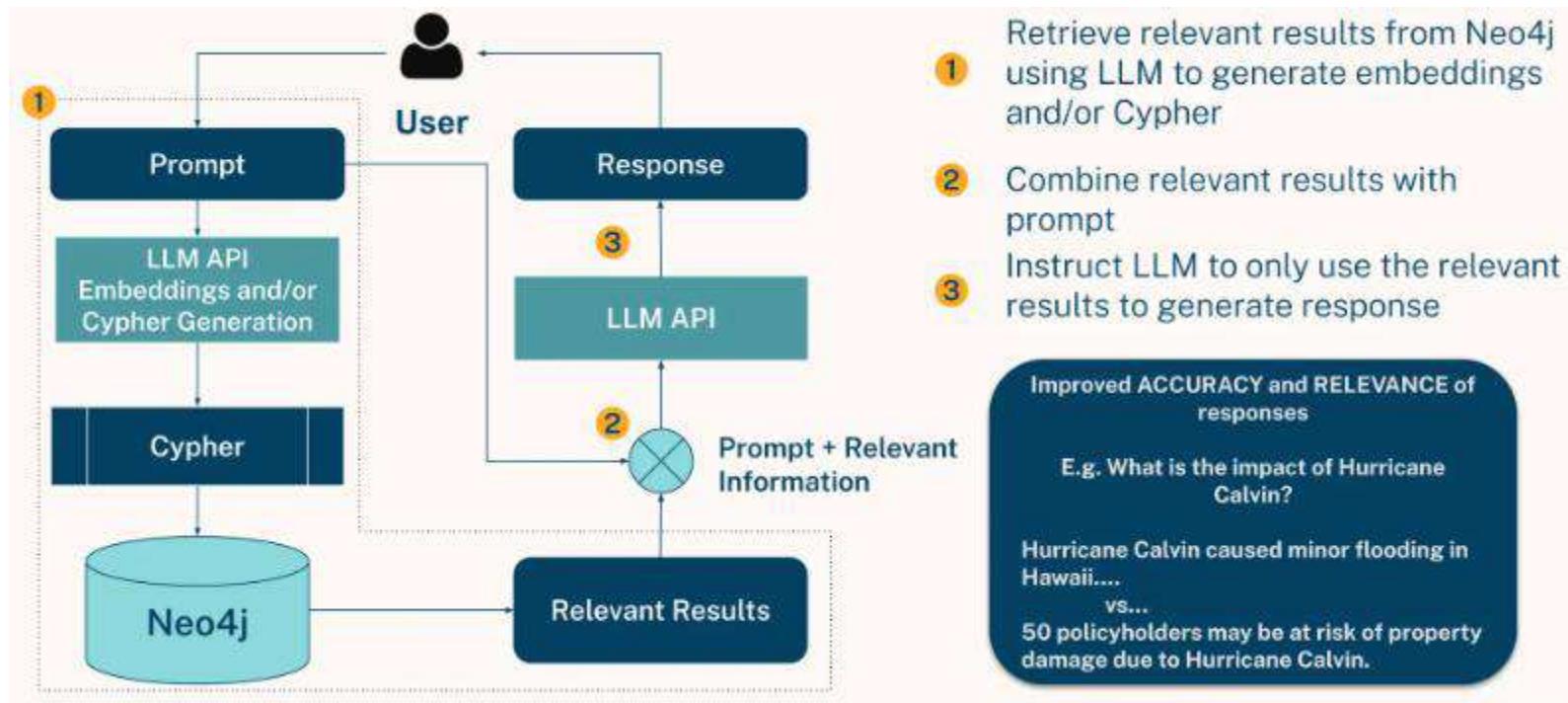
Ground responses
with information and
context in the graph

**Improve search
relevance** combining
vector search and
graph traversals

Knowledge Graph



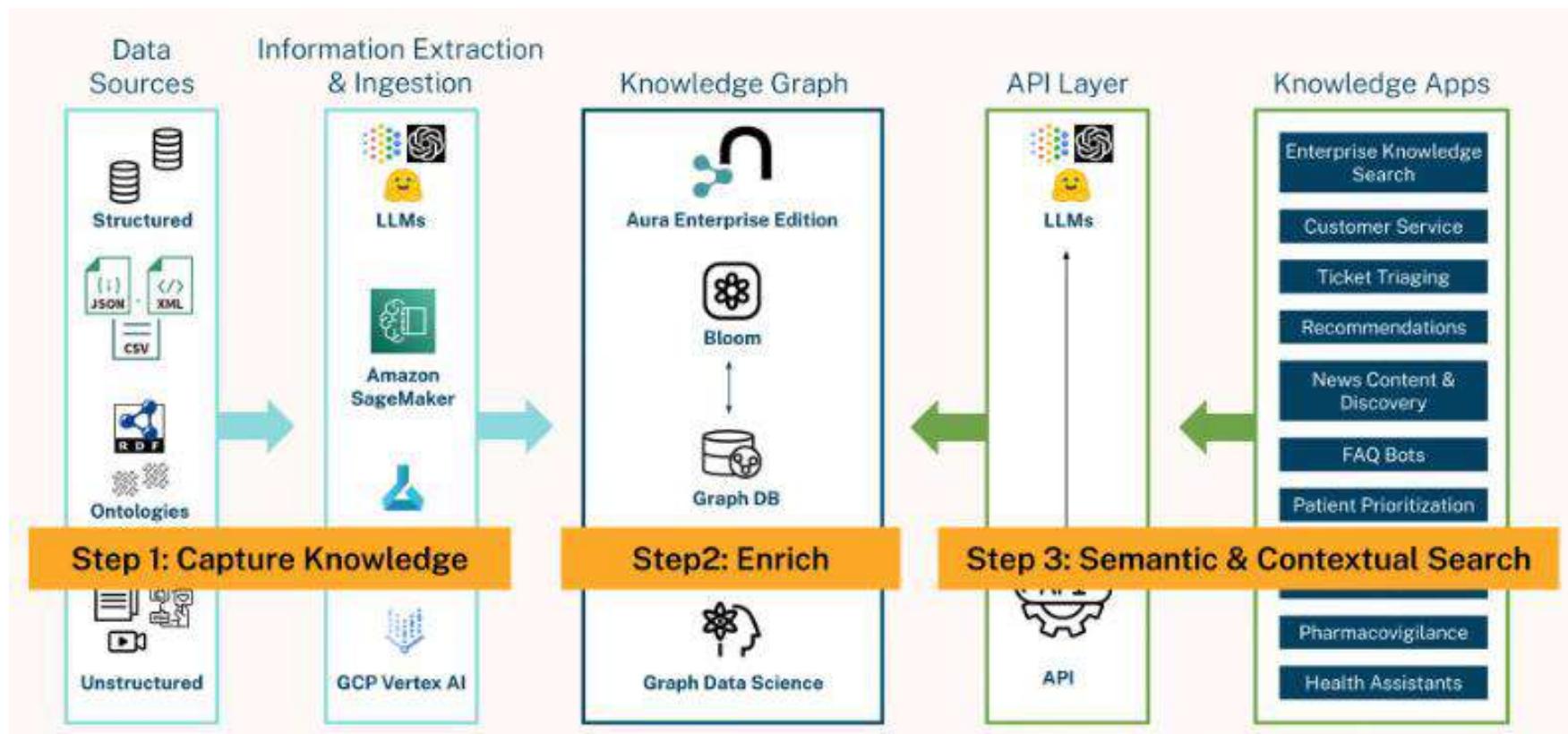
Improve results with knowledge graph using Neo4j



Vector RAG vs. Graph RAG

Feature	Naïve RAG	Graph RAG
Data Format	Unstructured Text	Structured Graph
Retrieval Method	Dense Vector Search	Graph Traversal/Querying
Reasoning Capability	Limited	Enhanced (Multi-hop)
Explainability	Low	High
Integration Complexity	Moderate	High
Use Cases	General QA	Domain-Specific Applications

Use of KG



Agentic RAG

Agentic RAG utilizes AI agents to make intelligent decisions about how to best retrieve information. Instead of a linear pipeline, agents can:

- **Decompose Complex Queries:** Break down a broad question into smaller, manageable sub-queries.
- **Iterative Retrieval:** Refine search queries based on initial results to find more relevant information.
- **Tool Use:** Interact with various tools, such as APIs, vector database, graph, or code interpreters, to gather information from diverse sources.
- **Self-Correction:** Analyze the retrieved information for relevance and quality, and re-attempt retrieval if necessary.

This approach is particularly effective for complex research, planning, and problem-solving tasks.

Agentic RAG

A framework where the **LLM acts as an agent**, not just a passive generator.

The agent:

- Sets subgoals.
- Plans retrieval steps.
- Executes tool use (e.g., search, APIs).
- Iteratively refines answers.

Core Idea: Make RAG interactive, iterative, and autonomous.

Agentic RAG

Key Components:

- 1. LLM Agent** – orchestrates reasoning and control.
- 2. Memory / Scratchpad** – stores context, subgoals, past actions.
- 3. Retrievers / Tools** – for search, API calls, calculations.
- 4. Planner / Executor** – decomposes queries, sequences actions.

Workflow Example:

User query → Plan → Retrieve → Reason → Iterate
→ Final answer

Agentic RAG Example

Query: “Compare the annual energy output of nuclear vs solar in 2024 globally.”

Agentic RAG Flow:

Decomposes into sub-queries.

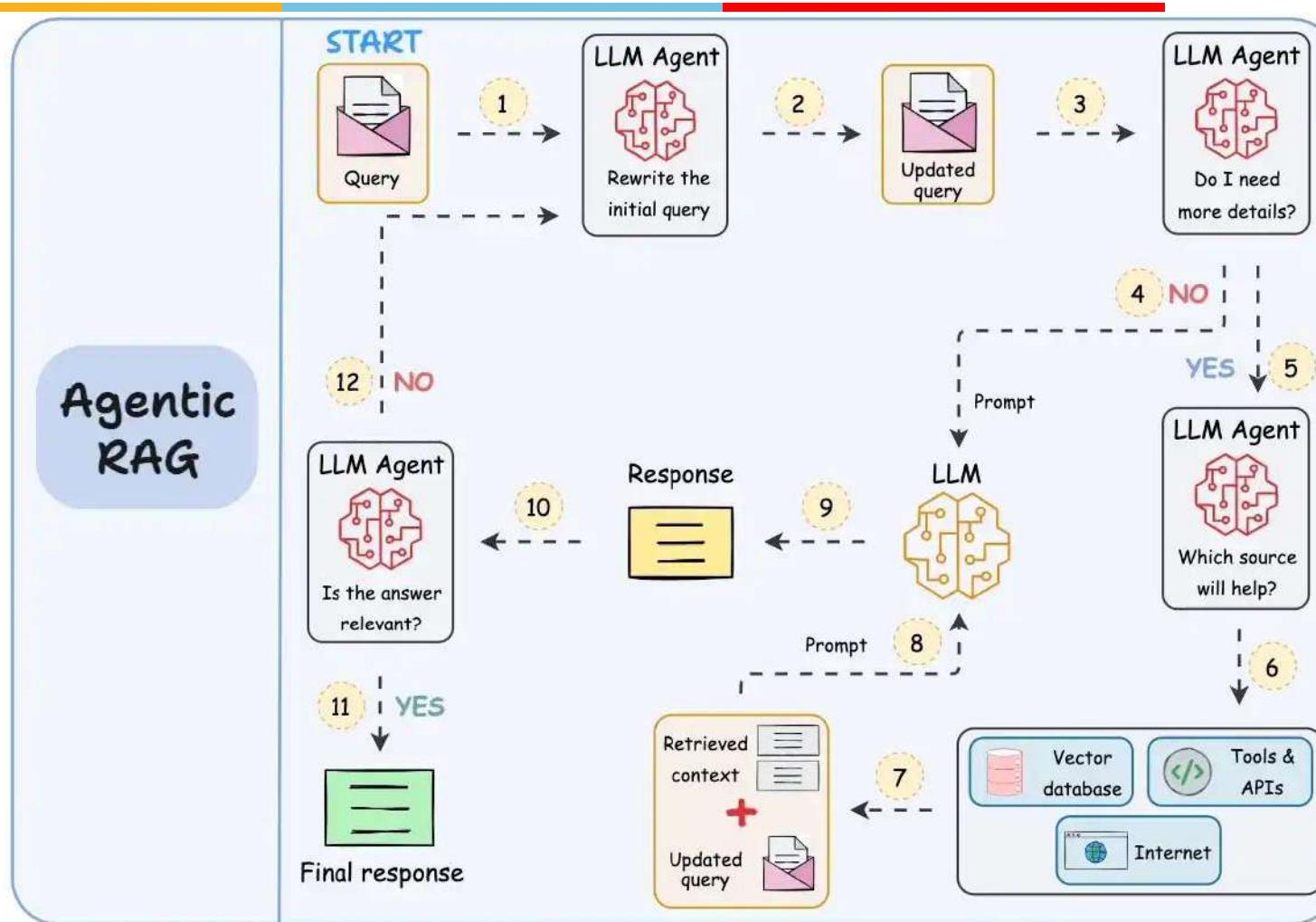
Searches for energy reports.

Extracts numbers, units, timeframes.

Performs calculations.

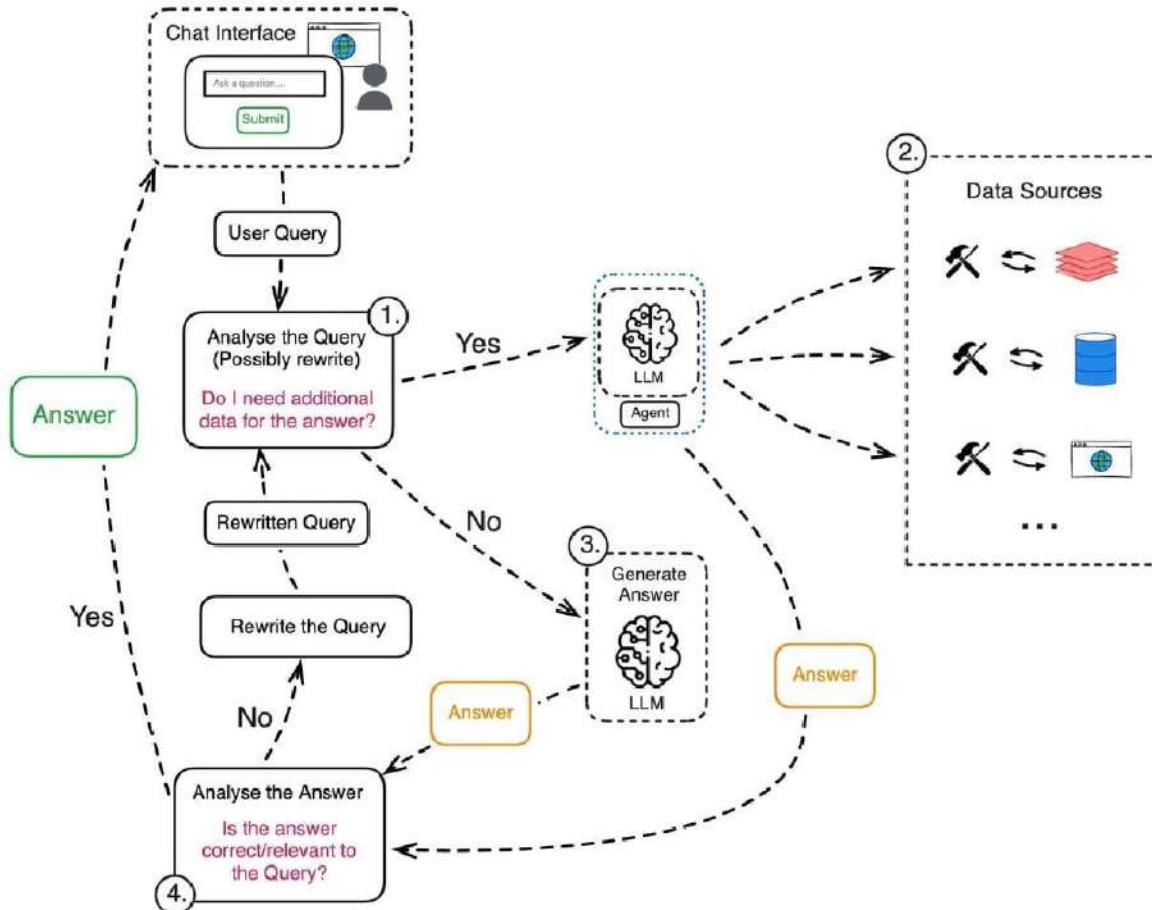
Returns synthesized answer with citations.

Agentic RAG



<https://blog.dailydoseofds.com/p/rag-vs-agentic-rag>

AI Agentic RAG - Example



Case Study

Analyze business impact of

- software & OS vulnerabilities,
- hardware & software upgrades,
- building/geographic disasters
- changes to business data formats

...across mission critical applications and business locations



Cybersecurity Agent Systems

- Companies like CrowdStrike & Palantir build **threat intelligence graphs**:
 - IPs
 - Malware signatures
 - Actors
 - Campaign timelines
- **Agents**:
 - Detect new threat patterns
 - Trigger automated alerts
 - Suggest mitigation steps

Case Study: RAG in the Financial Sector

Problem: Financial analysts need to process vast amounts of unstructured data (e.g., earnings call transcripts, SEC filings, news articles) to make informed investment decisions. This process is time-consuming and prone to human error.

Solution: An investment firm implemented an **Agentic RAG** system to create an "AI Research Assistant."

Key Goals:

Accelerate the analysis of financial documents.

Uncover hidden insights and relationships within the data.

Provide analysts with concise, data-driven summaries and answers to complex financial questions.

Hybrid RAG Architecture

Vector Search for Semantic Retrieval: Financial documents ingested into a **Weaviate** vector database for broad semantic search.

Graph RAG for Relationship Analysis: A **Neo4j** knowledge graph created to map relationships between companies, executives, and key financial events.

Agentic Layer: An AI agent, built with **LangChain**, to:

- Decompose complex queries like, "How has Company X's sentiment in earnings calls changed over the past year, and what are the key drivers mentioned in relation to their main competitors?"
- Query both the vector database and the knowledge graph to gather relevant information.
- Synthesize the findings into a coherent and actionable summary for the analyst.

Challenges and Future Directions

Challenges:

- Scalability of integrating large KGs with LLMs
- Ensuring data quality and consistency
- Balancing between structured and unstructured data
- Handling incomplete or noisy data.
- Integration complexity with existing systems.

Future Directions:

- Integration with agents and workflows.
- RAG + fine-tuning + agents = powerful hybrid systems.
- Retrieval from multimodal sources (text, code, images, video).
- Enhanced tools for automated KG creation.
- Developing unified frameworks for KG-LLM integration
- Advancing techniques for real-time KG updates using LLMs
- Development of hybrid models combining vector and graph retrieval.

Tools and Technologies Powering RAG



Vector databases: Pinecone, FAISS, Weaviate.

Embedding models: OpenAI, Cohere, Hugging Face.

LLMs: GPT-4, Claude, Mistral, etc.

State of art

DRAGON:

- Jointly pre-trains on text and KGs.
- Achieves notable performance on complex reasoning tasks.

KG-RAG:

- Integrates KGs with RAG for improved QA.
- Focuses on financial data applications.

GFM-RAG:

- Introduces a graph foundation model for RAG.
- Demonstrates state-of-the-art performance on multi-hop QA datasets.

GNN-RAG:

- Combines Graph Neural Networks with RAG for enhanced reasoning.

RAG

- **RAG is a powerful technique** for making LLMs more accurate, timely, and trustworthy.
- **Advanced architectures like Agentic, Graph, and Multimodal RAG** are pushing the boundaries of what's possible with AI.
- **Vector databases are a cornerstone technology** that enables the core functionality of RAG.

RAG Strategies Applications

- **Basic RAG**

- if your data is small and questions are simple

- **Hybrid + reranker RAG**

- if accuracy matters

- **Multi-step agentic RAG**

- if questions need reasoning over several documents

- **Structured RAG**

- if data → tables or APIs

- **Graph RAG**

- if relationships are important

RAG Impact

Increased Efficiency: Reduced the time spent on preliminary research

Deeper Insights: The ability to query across both unstructured text and a structured knowledge graph enable analysts to identify trends and risks that were previously missed.

Improved Accuracy: Grounding the LLM's responses in specific financial documents significantly reduces the risk of factual errors in reports.

Competitive Advantage: Able to react more quickly to market-moving information, leading to more timely and profitable investment decisions.

Key Challenges

Data Quality and Cleaning: The performance of any RAG system is highly dependent on the quality of the source data.

Chunking and Embedding Strategy: Finding the optimal way to break down and represent data is crucial for effective retrieval.

Evaluation and Monitoring: Continuously evaluating the relevance of retrieved documents and the accuracy of generated responses is a complex but essential task.

Technology and Tooling Ecosystem

Orchestration Frameworks:

- **LangChain & LlamalIndex**: Provide the core components and abstractions for building RAG pipelines.

Embedding Models:

- **OpenAI, Cohere, Hugging Face**: Provide pre-trained models for converting text into high-quality embeddings.

Large Language Models (LLMs):

- **GPT series (OpenAI), Claude series (Anthropic), Gemini (Google)**: The generative powerhouses that produce the final response.

Monitoring and Evaluation:

- **Arize AI, TruEra**: Platforms for monitoring the performance and accuracy of RAG systems in production.

Technology and Tooling Ecosystem

Vector RAG:

Open-Source: LangChain, LlamaIndex, with vector stores like Milvus, Qdrant, Chroma, FAISS and Weaviate.

Commercial: Pinecone, Cohere's RAG, and managed services on clouds like Vertex AI Search.

Graph RAG:

Open-Source: Neo4j with LangChain/LlamaIndex integrations, Diffbot, and custom knowledge graphs.

Commercial: NebulaGraph, Microsoft's GraphRAG, and knowledge graph platforms like Kensho.

Agentic RAG:

Open-Source: LangGraph, CrewAI, and AutoGen, which orchestrate complex RAG workflows.

Commercial: Cognosys, Glean, and various "AI agent" platforms

Understanding the Bottlenecks for Performance Optimization



Embedding Model Speed: The time it takes to convert the user query into an embedding.

Vector Search Latency: The speed of the retrieval from the vector database. This is influenced by the number of vectors, the dimensionality of the embeddings, and the indexing strategy.

LLM Generation Time: The time taken by the LLM to process the prompt and generate a response. This is heavily dependent on the size of the model and the length of the generated output.

Network Latency: The time for data to travel between the different components of the system.

Complexity of Advanced RAG: Agentic and Graph RAG can introduce additional computational steps (e.g., query decomposition, graph traversal), increasing overall latency.

Multimodal RAG

Our knowledge isn't just in .txt files. It's in:

- **Images:** Product photos, diagrams in manuals, satellite imagery.
- **Documents:** PDFs containing charts, tables, and complex layouts.
- **Videos:** How-to tutorials, product demos, security footage.
- **Audio:** Customer support calls, meeting recordings.

The Limitation of Standard RAG:

A text-only RAG system is blind to this rich, multimodal information. It can't answer questions like:

- *"Which of our products looks most like this picture?"*
- *"What were the Q3 revenue trends based on this chart in the annual report?"*
- *"Find the part in the video where the presenter explains the system architecture."*

Multimodal RAG

Multimodal RAG is an advanced AI framework that retrieves information from a knowledge base containing multiple data types (modalities) to generate a comprehensive answer.

It understands: Text, images, tables, and potentially audio/video.

It connects: The description of a product with its picture.
The text of a report with the data in its charts.

It enables: Users to ask questions using any modality (text or image) and get answers synthesized from all relevant data.

Multimodal RAG

Data Ingestion & Chunking: Documents (like PDFs) are broken down. Text is extracted, images are isolated, and tables are identified.

Multimodal Embeddings: A specialized model creates numerical representations (vectors) for all data types. Critically, it places the vector for an image of a dog near the vector for the word "dog" in a shared "meaning space."

Vector Database: These multimodal embeddings are stored and indexed for fast similarity search.

Multimodal RAG

Hybrid Retrieval: When a user asks a query (text or image), the system converts the query into an embedding and retrieves the most similar text chunks, images, or tables from the vector database.

Generation: The retrieved multimodal context (e.g., a paragraph of text and a relevant chart image) is passed to a powerful Multimodal LLM (like GPT-4o or Gemini) which generates the final, synthesized answer.

Multimodal RAG

Processes More Than Text: It integrates diverse data types like images, audio, and video, not just text, for a richer understanding.

Unified Data Representation: It uses embedding's to create a common language for different data formats, enabling cross-modal comparisons.

Improves Accuracy: By drawing from multiple data types, it gains better context, leading to more accurate and comprehensive answers.

Enables "Any-to-Any" Search: You can use one data type to find another, such as using a text query to retrieve relevant images or vice-versa.

Drives Advanced Applications: This technology powers innovative tools like visual search, interactive education, and complex document analysis.

Multimodal RAG Tools and Technology

Core Technology: Multimodal embedding models (like CLIP) create a unified search space for text, images, and audio.

Infrastructure: Vector databases (e.g., Milvus, Pinecone) are required to store and efficiently query these diverse embeddings.

Orchestration: Frameworks like LangChain and LlamaIndex are used to build and manage the entire RAG pipeline.

Specialized Tools: New open-source projects like Morphik (visual-first) and RAG-Anything (all-in-one) are emerging to solve specific multimodal challenges.

Generation Power: Advanced LLMs like GPT-4o and Gemini simplify the process by directly accepting multimodal inputs for the final response generation.

Enterprise Solutions: Cloud platforms from Google (Vertex AI), AWS, and Azure offer managed services for building scalable, production-grade applications.

KG role in real world applications

Domain	Application	KG Role
RAG	Graph-RAG, entity-guided retrieval	Structuring retrieval & reasoning
Search/QA	Semantic search, factual QA	Structuring answer space
Conversational AI	Context tracking, disambiguation	Tracking entities and relations
Recommendation	Product/movie suggestions	Semantic user-item modeling
Biomedical	Drug discovery, literature synthesis	Relationship modeling
Fraud/Finance	Anomaly detection	Graph-based behavior patterns
Manufacturing	Digital twins, predictive maintenance	Asset and sensor modeling
XAI	Explainable AI	Traceable reasoning via graph paths
Agentic AI	Planning and decision-making	Knowledge state and goal representation

Real World Applications

Biomedical Research: Integrating UMLS KGs with LLMs for accurate medical Q&A.

Legal Analysis: Combining legal KGs with LLMs to assist in case law research through structured data retrieval.

E-commerce: Enhancing product recommendations and customer support by merging product KGs with LLMs.

Education: Personalizing learning experiences using educational KGs and LLMs.

Financial Reporting: Analyzing annual financial statements using KG-RAG frameworks.

How Knowledge Graphs Power Agentic AI



- **World Modeling for Agents**
 - KGs give agents a structured understanding of the environment (people, objects, rules, relationships).
 - Agents use this to plan actions instead of just reacting to text.
- **Long-term Memory**
 - Agents store facts, tasks, outcomes, and preferences in a KG.
 - This becomes the agent's persistent, queryable memory.
- **Reasoning & Decision Making**
 - KGs enable **symbolic reasoning** (e.g., infer new facts, check constraints).
 - Agents combine LLM intuition with KG logic → **hybrid neuro-symbolic intelligence**.

How Knowledge Graphs Power Agentic AI

- **Multi-Agent Coordination**

Multiple agents share a common KG to:

- Avoid duplicating work
- Share discoveries
- Maintain consistent global state

The KG becomes a **shared blackboard**.

- **Tool Use & Action Planning**

Agents need knowledge about tools, APIs, and workflows.

KGs encode:

- Tool capabilities
- Preconditions/effects
- Dependencies

This lets agents plan **multi-step workflows** automatically..

How Knowledge Graphs Power Agentic AI



- **Grounding Agents in Facts**
 - Prevents hallucinations by grounding decisions in verified data.
 - Agents check KG facts before executing actions.
- **Context Retrieval for RAG/GraphRAG Agents**
 - Agents use KGs to fetch the right context, not random chunks.
 - GraphRAG gives agents structured, controllable retrieval.
- **Autonomy Feedback Loop**
 - Agents write back results →
 - KG updates →
 - Agent uses updated KG to improve future decisions
→ Self-improving, evolving agent system.

References

- Berners-Lee, Hendler, Lassila (May 17, 2001). The Semantic Web. Scientific American Magazine. @ <http://bit.ly/2z2LZIE>
- Berners-Lee. A roadmap to the Semantic Web. Tim Berners-Lee's Design Issues @ <http://bit.ly/2z29fGP>
- Berners-Lee. Linked Data. Tim Berners-Lee's Design Issues @ <http://bit.ly/21MR3Zt>
- Heath, Bizer. Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology @ <http://bit.ly/2xIWKou>
- <https://wiki.dbpedia.org/services-resources/ontology>
- <https://www.emse.fr/~zimmermann/Teaching/SemWeb/semwebintro.pdf>
- <https://www.youtube.com/watch?v=C9M7N979pIU>
- <https://enterprise-knowledge.com/whats-the-difference-between-an-ontology-and-a-knowledge-graph/>
- <https://www.w3.org/Consortium/facts>
- <https://pages.semanticscholar.org/coronavirus-research>

References

- <https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/>
- [Google Knowledge Graph](#)
- [Building Knowledge graph](#)
- <https://www.youtube.com/watch?v=lUc0woFX16M>
- https://www2.slideshare.net/jeffpan_sw/linked-data-and-knowledge-graphs-constructing-and-understanding-knowledge-graphs?qid=ab5124c7-a15a-4f77-9fc9-914feac00a6a&v=&b=&from_search=1
- https://www2.slideshare.net/phaase/getting-started-with-knowledge-graphs?from_action=save
- <https://lod-cloud.net/>
- <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>
- https://www.slideshare.net/slideshow/embed_code/key/dJaP9LmSd1uPXq
- <https://www.youtube.com/watch?v=hILYM9oBI0M>
- <https://www.youtube.com/watch?v=sYejXeuZaeY>

References

- Singh, Aditi, et al. "Agentic retrieval-augmented generation: A survey on agentic rag." *arXiv preprint arXiv:2501.09136* (2025).
- "**Retrieval-Augmented Generation (RAG).**" *Google Cloud Technical Documentation*.
- McKinsey & Company "The economic potential of generative AI: The next productivity frontier."
- **RAG for Finance: Automating Document Analysis with LLMs**
- "**Retrieval Augmented Generation (RAG).**" *Pinecone Learning Center*.
- Analysts Explore How Retrieval-Augmented Generation is Emerging as a Key Strategy for Deploying GenAI Applications at the Gartner Data & Analytics Summit
- Need for Contextualized, Reliable and Cost-Effective Solutions Is Driving the Shift Toward Small Task-Specific AI Models
- Financial Analysis: Intelligent Financial Data Analysis System Based on LLM-RAG
- [Future trends in retrieval augmented generation what to expect in 2025 and beyond/](#)
- [RAG Use Cases and Applications](#)
- [The Future of RAG is Agentic - Learn this Strategy NOW](#)
- [Multimodal RAG](#)

Code References

- https://colab.research.google.com/github/mistralai/cookbook/blob/main/mistral/rag/basic_RAG.ipynb
- <https://colab.research.google.com/github/mrdbourke/simple-local-rag/blob/main/00-simple-local-rag.ipynb#scrollTo=ELIOZ6KTWQga>
- https://colab.research.google.com/github/tomasonjo/blogs/blob/master/llm/enhancing_rag_with_graph.ipynb?ref=blog.premai.io#scrollTo=jPIRSGz4tHNV
- <https://colab.research.google.com/github/microsoft/LLMLingua/blob/main/examples/RAG.ipynb>
- https://colab.research.google.com/github/huggingface/cookbook/blob/main/notebooks/en/agent_rag.ipynb#scrollTo=5Dy-k3Eqxcrj
- <https://colab.research.google.com/github/tomasonjo/blogs/blob/master/bionlp/bioNLP2graph.ipynb>