**Dublin City University**

**School of Computing**

# CA4009: Search Technologies

# Section 6: Semantic Search

Gareth Jones

October 2019

# Introduction

Information retrieval by term matching is a simple approach which makes no attempt to “understand” the contents of the items to be retrieved.

- it doesn’t attempt to understand the words, phrases, sentences, documents or queries.

Given its simplicity, it is remarkably effective - and often very suitable when the information need is not or cannot be clearly specified - consider the issues of ASK again.

But this approach can fail due to mismatch between the terms in the query and the relevant documents.

e.g. “n y times” does NOT match “new york times”

# Introduction

Successful query and document matching often relies on the principle of *redundancy* of information in the document collection.

Redundancy here essentially means that relevant information will appear in more than one document, ...

and that at least one of them will use the same vocabulary as the user’s query.

Search for relevant documents fails when this assumption fails.

Semantic matching in search seeks to overcome this mismatch problem by better understanding or representing features of the query and documents.

The methods for improving ranking of relevant documents are no use if we can’t retrieve relevant documents in the first place due to term mismatch!

## Introduction

Semantic matching in search seeks to overcome this mismatch problem by better understanding or representing features of the query and documents.

Text can be interpreted and understood at multiple levels of completeness.

We can interpret and represent: words, phrases, sentences, text structure, without trying to fully understand the text.

This partial understanding of the text can be used to better satisfy user information needs:

- improve user experience - answer question type queries directly, more informative summaries,
- make search more reliable - address term mismatch problems

# Question Answering

Information retrieval systems are concerned with searching for and retrieving documents relevant to a user's information need.

The searcher is required to extract information from the retrieved documents in order to satisfy their information need.

In the case of text documents, they do this by reading, while in the case of multimedia content they do this by listening to audio or watching a video.

However, many user requests are essentially questions looking for a single fact or piece of information that can be made in a short statement.

Addressing these information needs by analysing whole retrieved items can be time consuming and highly inefficient.

# Question Answering

Retrieved text documents can be augmented with simple features such as highlighting of query terms present in the text to aid visual scanning for relevant content, but this still requires the user to read and interpret the text which takes time.

For question type requests, the efficiency of accessing relevant information can be improved by processing the retrieved items to concentrate on information relevant to the question.

This of course will only work for question type requests - but analysis of query logs shows that there are a lot of these!

For broad exploratory information needs, standard IR systems must still be used to explore the topic.

# Question Answering

Summarization is a means of reducing the user effort in analysing a document,

e.g. the use of documents snippets in a ranked list to enable to users rapidly identify relevant documents without needing to read the whole document.

Sometimes the information necessary to address the information need can be captured in a summary.

A step beyond this for reducing user effort is *question answering (QA)* systems.

Instead of retrieving potentially relevant documents for the user to read or otherwise access, a QA system seeks to provide the answer or answers to the user's question.

## Question Answering

Requests which can be addressed by QA systems have differing levels of difficulty. They may have:

- simple answers to simple questions, e.g. an easily identified name of a person, place, organisation, etc.
- simple answers to difficult questions, e,g. where an amount of money must be totalled from a number of different sources, e.g. how much money did Manchester United spend on players last season?
- difficult answers to difficult questions (sometimes apparently easy ones), e.g. composing a list from multiple documents, or contrasting information in multiple documents.

## Question Answering

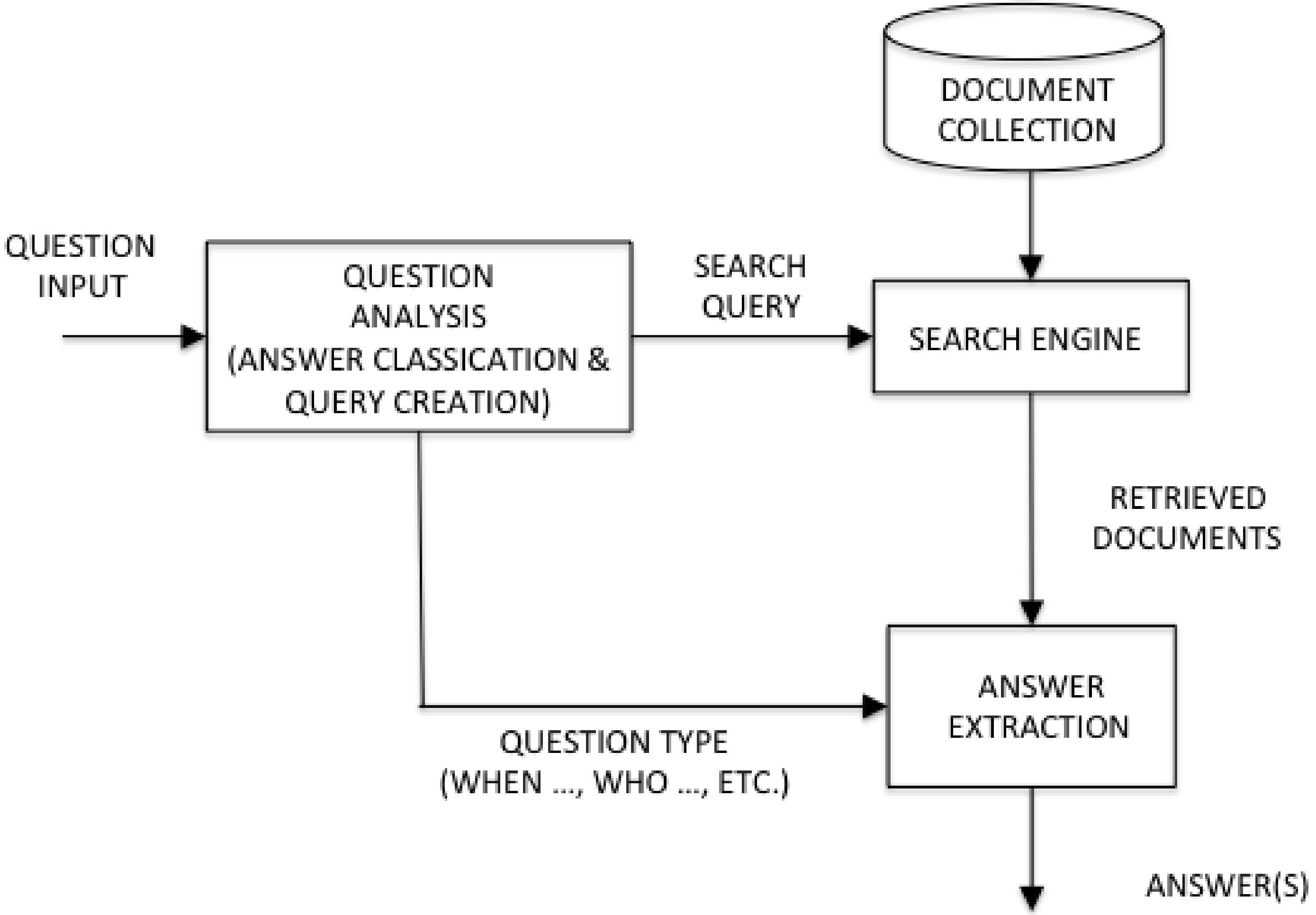
A general purpose QA system or even one for a broad specific domain, e.g. medicine, needs to access a large set of information or knowledge in order to have a reasonable prospect of it containing the answer.

One approach to QA is to use an IR system to identify a small number of documents from within a large document collection which are potentially relevant to a request, and to attempt to extract the answer to the question from these documents.

Finding the answer within a retrieved item will require the application of an answer extraction component to the retrieved documents.

Standard IR-based QA pipeline: system typically combines IR to create a reduced answer space of retrieved documents and answer extraction to locate the answer within these documents.

# IR-based Question Answering



Typical Question Answering workflow

## IR-based Question Answering

Typical workflow of a QA system:

- Analyze question to determine what type of question it is.
- Form a search query from the question. The query type, e.g. the word “when”, is usually left out of the query since it is not related to the topic and identification of potentially useful documents.
- Enter the query into an IR system to retrieve a number documents which are potentially relevant and potentially contain the answer.
- Enter output of question analysis module and top ranked retrieved documents into answer extraction module.

## IR-based Question Answering

- The answer extraction module analyses retrieved documents to try to identify the answer or answers to the question.

The IR component is generally robust and computationally cheap to apply.

The answer extraction module is typically more expensive to apply.

The retrieved list is truncated either after a fixed number number of documents or based on the rate of decrease in the matching scores.

There is a tradeoff in precision and recall in deciding the length of the list.

- Too short: answers to the question might not be included (so the recall of the possible answer will be too low).
- Too long: too much irrelevant information may be included increasing the computation time and chance that the wrong answer might be selected.

## IR-based Question Answering

Two contrasting approaches to QA:

- knowledge-based: apply formal natural language processing (NLP) and extensive linguistic resources.
- data-based: use of large document collections with shallow informal language processing, and exploitation of information redundancy.

Both approaches adopt the same standard QA architecture.

The IR stage identifies a number of documents from which the answer might be extracted, but they differ fundamentally in the technology of the answer extraction module.

We will look at both of these approaches.

## Knowledge-Based Question-Answering

Retrieved documents are parsed to find the answers to the question.

- locate examples of *Expected Answer Type (EAT)* in the retrieved documents.
- examine relationship between words from the question and the possible answers to try to identify the answer to the question.

The NLP methods used here generally fall into the category of information extraction (IE) methods which are designed to extract specific pieces of information from a text, e.g. analyse a text to determine that Dublin is the capital of Ireland.

The approach can fail if the NLP methods used or the dictionaries are not sufficient to cover the details in the questions or the answers.

# Knowledge-Based Question-Answering

Process the question to determine the EAT.

The EAT is often some type of named entity, e.g. someone's name.

First, identify the word(s) in the question that determine the EAT:

*who* (easy - looking for a person)

```
Who is the president of Ireland?
```

*what* (much harder - may refer to many different entities).

```
What do most tourists visit in Dublin?
```

The EAT is “landmark”

# Knowledge-Based Question-Answering

When we know the expected type of the answer, we can analyze the documents to try to find potential answers which match this type requirement.

“How much did Manchester United spend on players last season?”

- EAT = money

A superficially easy, but possibly difficult question to answer.

What would the QA system need to be able to understand in a text in order to be able to answer this question?

What does “last” refer to? Possibly need to gather information on players from multiple documents. May be problems in identifying individual players.

# Knowledge-Based Question-Answering

To assist with answering the questions, the contents of the available documents can be augmented with knowledge contained in additional domain specific data structures. These include:

- simple lists of facts
- conventional structured databases
- purpose built knowledge structures - *taxonomies* and *ontologies*

# Knowledge Representation using Taxonomies and Ontologies

Taxonomies are a means of capturing information about a particular topic in an hierarchical structure.

A subcategory of a category inherits the properties of its parent.

For example,

motor vehicles → cars, vans, lorries, etc

cars → family, sports, mini, etc

Vehicles can be defined in different taxonomies for different applications.

## Knowledge Representation using Taxonomies and Ontologies

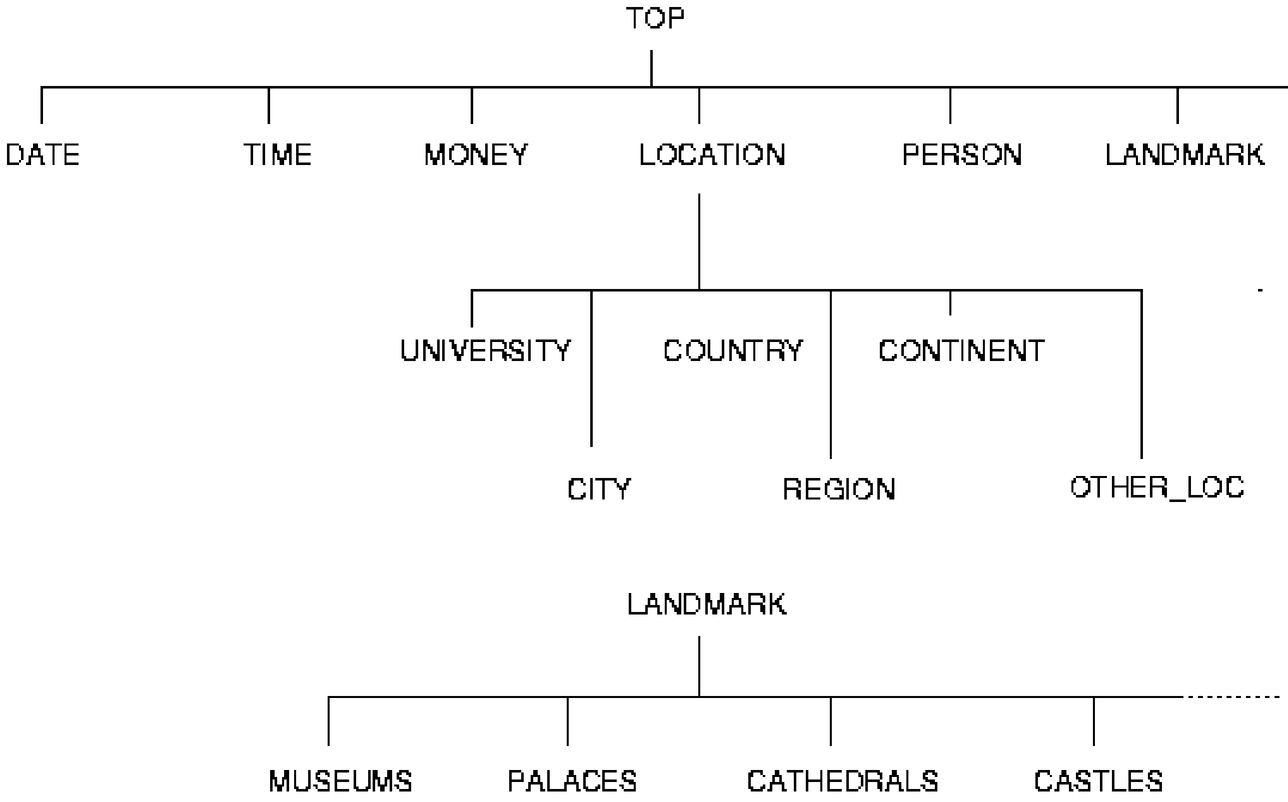
Ontologies generally include all the information captured in a taxonomy, but also details about rules and relationships between categories.

An ontology may include semantic information that can be used in problem solving and decision making.

e.g. relations between - “A is the father of B”

# Knowledge-Based Question-Answering

Using a taxonomy of places to identify location types which can be classified as landmarks.



# Data-Based Question-Answering

Within small document collections often only one document may answer the question. (of course, sometimes no document will answer the question!)

To answer a question from such a collection, the QA system must locate the answer in this specific document.

For small collections both the question and the documents often need to be analysed in detail to understand the question and to match it with the answer(s).

For very large document collections (e.g. the WWW) many documents may contain the answer to the question.

# Data-Based Question-Answering

The answer to the question: *Who killed Adraham Lincoln?*,

can be extracted from the text:

*John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.*

but it will be quite difficult.

Thus in small collections it is more likely that sophisticated NLP will be needed to locate and extract the answer, since it may only be stated once in a complex statement.

The QA system must be sophisticated and robust enough to handle any way in which the answer may appear in the text.

## Data-Based Question-Answering

In a large collection, particularly one such as the Web with many authors writing about the same topic, the answer to a question will often be stated in alternative ways in a number of different documents.

In some of these documents the answer may be written in a simple way which can be matched very easily to the question, e.g. one of the documents may include the statement: *John Wilkes Booth killed Abraham Lincoln.*

Much shallower linguistic processing can be applied in this case, as shown in the following example.

# Data-Based Question-Answering

Rather than attempting to “understand” the relevant contents of the question and the document, data-based QA systems rely on simple pattern matching.

For example, the question can be rewritten according to simple rules to form possible answer patterns.

For the question: *Where is the Louvre Museum located?*

The obvious answer pattern is: *The Louvre Museum is located in*, so we only need to find the pattern: *The Louvre Museum is located in Paris.*, in one document to be able to answer the question.

For the general question form *Where is* $w_1 w_2 \ldots w_n$, potential answer patterns can be generated by moving the verb to all possible position, e.g. $w_1$ is $w_2$ $w_3 \ldots w_n$, $w_1$ $w_2$ is $w_3 \ldots w_n$, etc.

## Data-Based Question-Answering

Some of these patterns will be nonsense, e.g. *The is Louvre Museum located in*, *The Louvre is Museum located in*, but these are unlikely to match with the content of any retrieved document.

We can reasonably expect to find the statement *The Louvre Museum is located in ...* in at least one document on the Web.

Having found the statement we are looking for in one of the documents, we can reasonably hypothesize that the word following the statement will be the answer to the question.

In this case of course the answer will mostly probably be “Paris”, although we might also find the answer “France” or the district of Paris where the Louvre is located. If we have multiple possible answers, then we could use the method in the next example to help choose the “best” answer.

## Data-Based Question-Answering

Even for very large collections, we may not find the answer in exactly the form that we looking using this exact string matching method.

When no obvious answer can be found in the text, redundancy can often provide a reasonable “guess” at the correct answer, or to help us choose from among multiple possible answers (France, Paris, ... - see previous example).

We can look for words from the question in close proximity to a word of the correct form for the answer, e.g. a number or a place.

# Data-Based Question-Answering

Question: *How many times has Roger Federer won Wimbledon?*

Assume that we are not able to find an answer of the form *Roger Federer has won Winbledon 8 times.*, but we can find a number of documents containing phrases of the form.

(1) **Roger Federer** blah blah **Wimbledon** blah blah **8** blah . . .

(2) **Wimbledon** blah blah blah **Roger Federer** blah **35** blah . . .

(3) blah **Roger Federer** blah blah **8** blah blah **Wimbledon** . . .

(4) **8** blah blah **Wimbledon** blah blah **Roger Federer** . . .

## Data-Based Question-Answering

In order to answer the question we could simply look for words in close proximity (remember Luhn’s hypothesis about word proximity from earlier in the module), i.e. in this case Roger Federer and Wimbledon, and look for possible answers nearby.

Some more analysis could show that “How many ...?” questions will have a numerical answer. So we could look for numbers in close proximity to the question words.

The majority answer is most likely to relate to both the concepts here, i.e. Roger Federer and Wimbledon.

# Data-Based Question-Answering

Since someone asks the question ”How many” in relation to Roger Federer and Wimbledon, we know that we are looking for a number.

If we find a number located close consistently to “Roger Federer” and “Wimbledon” - there is a reasonable chance that this is the answer to the question.

The number 35 probably relates to at least one of the entities (Federer or Wimbledon), and possibly both, but is probably less important since it only appears once. It may have been Federer’s age when the article was written, or may it relate to some specific incident involving him and Wimbledon, perhaps the aces served in a match.

But the most frequent number relating to both concepts is 8, so this can be proposed as the answer to the question.

# Data-Based Question-Answering

In an interactive system, what could we do if the user knows or thinks that the answer to the question proposed by the QA system may be wrong or wants to confirm that it is correct?

# Evaluation

As with information retrieval systems, evaluation is important in the development of QA systems.

What metrics should we use?

Correct: percentage of all questions posed which answered correctly

Precision: percentage of questions which are answered where the answer is correct.

Correct may equal Precision - but it may not! why?

Recall: percentage of questions that could be answered correctly from the document collection that have been answered correctly.

No answer: percentage of questions that cannot be answered from the document collection, that have been identified as such. Identifying questions which cannot be answered is an important feature of QA systems.

# Evaluation

Evaluation needs:

- Set of documents which will be used to answer the questions.
- Set of questions[a].
- The answers to the questions.

As with evaluation of IR systems, the documents and questions should be representative of the QA task for which the system is being developed.

[a] It may not be possible to answer all the questions from the document set.

## Question Answering using Knowledge Graphs

A question answering system using documents as its information is only able to address questions where the answer is expressed in at least a retrieved document.

Question answering system using information extraction from documents (see the Summarisation section)

- extracting facts from documents into a database using NLP

can be used to combine information from multiple documents a into database entries.

But this cannot be done on the fly at search time, and not all useful information will be contained in retrieved documents.

## Question Answering using Knowledge Graphs

A more powerful and useful way to capture interrelated information is a *knowledge graph*.

A knowledge graphs aims to encode information from text sources, e.g. web pages, news, books, etc. in a formal representation.

It is essentially an extension of the ontologies which encode information on a topic and its relationships.

Knowledge graphs underlie new services offered by web search engines, and services provided by Apple Siri, Microsoft Cortana, Amazon Echo and Google Now.

Facebook and Twitter use knowledge graphs to recommend new contacts, posts or adverts.

## Question Answering using Knowledge Graphs

Information contained in a knowledge graph is referred as *structured data* since it is encoded in a form which can easily be accessed by computers.

There is a trade off in construction of a knowledge graphs between the complexity of the stored information, and the efficiency with which it can be accessed by a computer.

The most popular form of knowledge graph used in search focuses on:

- entities: persons, organisations, locations, products - usually real world objects
- relations: join entities
- facts: combination of entities and relations

## Question Answering using Knowledge Graphs

Types of entities will have typically attributes, e.g. people will have:

- first name, last name
- date of birth
- place of brith
- mother, father, often spouse and children
- occupation

Once a person has been identified as an entity, the knowledge graph construction process looks to assign values to the standard attributes of a person.

# Question Answering using Knowledge Graphs

A simple example:

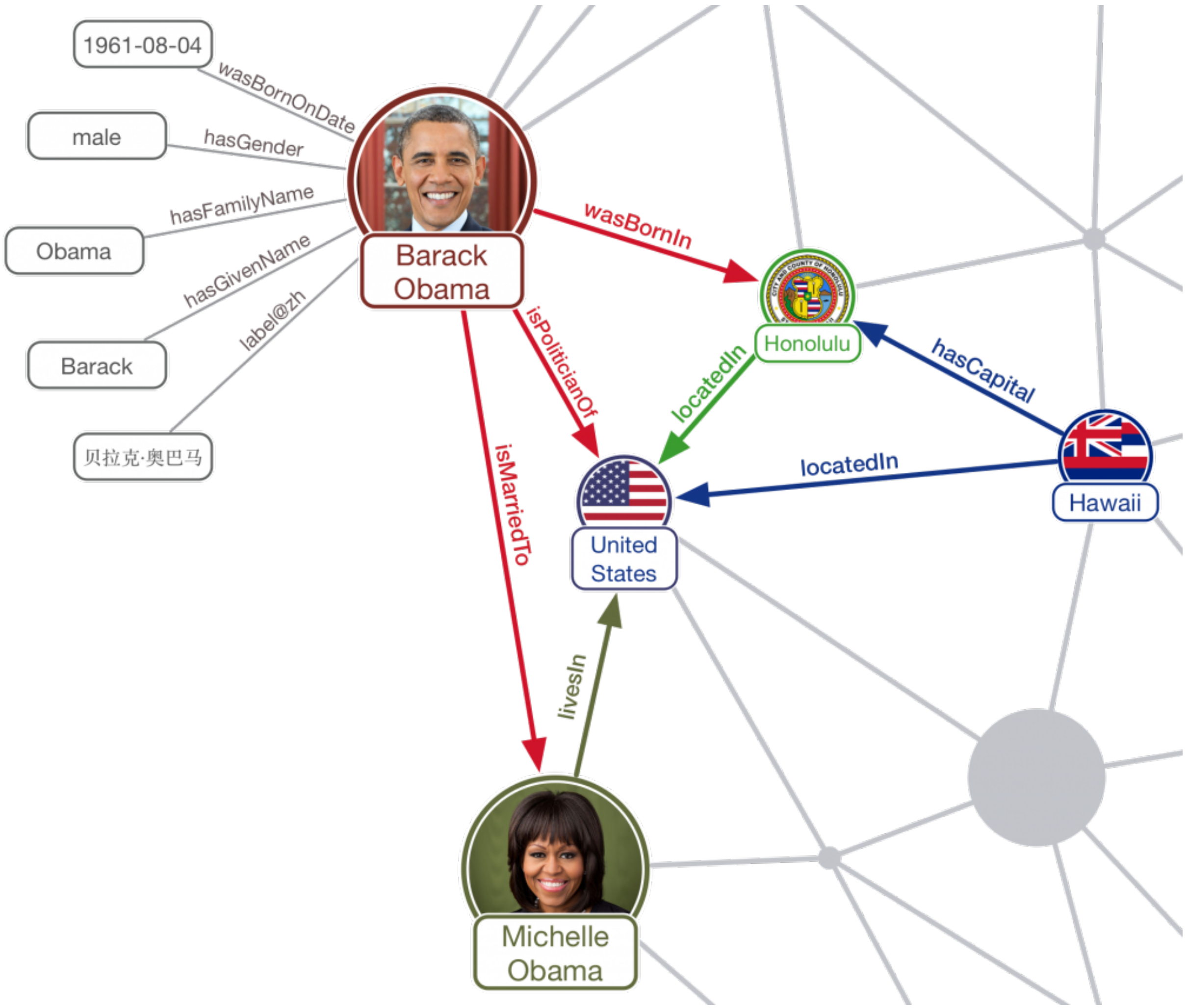
- entities: `Lionel Messi, Argentina`
- relations: `<plays_for>, <was born_in>`
- facts: combine the previous two

  `<Messi, plays_for, Argentina>`

A more complex example feature Barack Obama follows.

This shows the relations between Barack Obama and other entities, and attributes such as his first name, last name and date of birth.

# Question Answering using Knowledge Graphs



## Question Answering using Knowledge Graphs

Workflow of a question answering system based on a knowledge graph:

- Use *semantic parsing* method to map natural language question into a formal query statement in a form which can be efficiently executed against a knowledge graph.
- Attempt to answer the question from the knowledge graph using the formatted question.

Knowledge graphs have a generative capacity to represent an infinite number of facts never seen in an individual text.

# Question Answering using Knowledge Graphs

But:

- knowledge graphs can be incomplete
- the relations captured in the knowledge graph may not be expressive enough to represent the answer to the question

Hybrid combinations between document based question answering using information retrieval and question answering based on knowledge graphs can offer:

- potential for improved efficiency of answering “easy” questions using the knowledge graph,
- more specialist questions using documents.

# IBM Watson

IBM developed *Watson* as an example of a system which aggregates large amounts of data of multiple types from many different sources to achieve an objective.

The development of Watson is an example of a “Grand Challenge” in computing which sets out to promote rapid development in a new area

IBM set out to develop a system to beat human champions at the TV game show “Jeopardy”!

Watson combines multiple established technologies - all based on research in AI related topics - including IR, NLP, machine learning, etc.

These work together to leverage a variety of unstructured and structured data and information sources in combination.

## Cognitive Computing

Watson is an example of what IBM refers to as a *cognitive computing* application.

Cognitive computing systems are defined as incorporating the following fundamental principles:

- *Learn* - use data to make inference about something, e.g a domain, topic, person, ..., based on training and observation.
- *Model* - create a model of something and dynamically incorporate data into the model as it is observed.
- *Generate hypotheses* - assume that there is not a single correct answer, the system is probabilistic and uses data to score multiple hypotheses which potentially address a particular problem.

# IBM Watson

From IBM's perspective Watson is a means to improve business processes through natural human interaction with machines. In Watson's case by using natural language.

Users have become proficient in obtaining information using standard IR systems, e.g. web search engines, Watson seeks to support users in a different way via QA.

Watson promotes dialogue in information search - it can either give a potential answer or answers, or ask a follow up question seeking more information, in order to identify potential answers.

The cognitive computing approach seeks to ensure that the accuracy of results continues to improve through offline and online iterative training. Watson becomes "'smarter" each time new data is ingested.

# IBM Watson

Exploiting all the available information for a task or domain requires the availability of diverse information processing technologies.

Watson combines both shallow and deep information analysis and processing technologies.

Watson can integrate information from multiple sources and integrate them into a knowledge graph.

In QA terms, Watson combines elements of knowledge-based and data-based type QA methods with interrogation of knowledge graphs.

All available technologies within Watson can potentially produce hypotheses from which the answers(s) might be selected.

## Jeopardy: the rules of game

Jeopardy is a TV quiz show in which contestants are required to answer general knowledge questions.

The game proceeds as follows:

- The host reads the question.
- When the question has been read in its entirety, a “ready” signal light is illuminated.
- Contestants are now allowed to active their buzzer to answer.
- The first contestant to active their buzzer has the chance to respond.

# Jeopardy: the rules of game

Watson received the questions as electronic texts at the same moment they they were made visible to the human players.

Watson first had to determine whether it was sufficiently confident in its potential answer to activate the buzzer to get the right to answer.

For each question, Watson’s three most probable answers were displayed.

After it had activated the buzzer, Watson gave its answer using speech synthesis.

Watson won the competition, but had problems with short questions containing only a few words.

# Planning to Win at Jeopardy

To win Jeopardy by beating the best human contestants, IBM determined:

- You need to answer 70% of the questions, and get the answer right 80% of the time.
- You need to be able to answer in 3 seconds.

There is thus likely to be a tradeoff in accuracy and speed.

The correct answer is more likely to be identified using more sophisticated analysis and NLP methods, but the answer may come too late to be counted.

# Planning to Win at Jeopardy

The main challenge to answering the questions in Jeopardy is their diversity.

Watson needed to ingest a wide range of information in order to be able to answer questions on a wide range of topics.

The data sources ingested by Watson included encyclopaedias, Wikipedia, dictionaries, historical documents, textbooks, news articles, music databases, literature, etc.

Watson had access to 200 million pages of structured and unstructured content while playing the game.

It was not connected to the internet during the game!

# Planning to Win at Jeopardy

The general Watson strategy was:

- Question classification: fact-based, puzzle, pun - different answering approach used depending on question class.
- Identify many possible answers in parallel. These possible answers need to be broad, but not so broad as to be confusing.
- Determine a confidence level for each answer. Combine different sources of evidence and scoring to produce confidence score - similar idea to learning-to-rank in web retrieval.
- Choose the best answer.

# IBM Watson

Strategy to make Watson fast enough to meet the 3 second response deadline.

- Operate many machines in parallel.
- Use powerful individual computers.
- Store knowledge base in RAM to minmize access time.
- Use really fast networking!

# Commercial Application of Watson

IBM didn’t ultimately build Watson as a cool toy to win Jeopardy!

The Jeopardy application was an important technical grand challenge in development of Watson technology.

Winning at Jeopardy was great publicity for IBM and Watson.

IBM see Watson technology as an important direction for development of commercial cognitive computing applications.

## Commercial Application of Watson

The questions posed to Watson are likely to be much more complex in commercial applications, often without a specific correct answer.

Watson needs to be able to identify many potential answers, and will sometimes need to return alternative answers ranked by confidence, possibly with supporting evidence made available.

Providing evidence for returned information in this way is often referred to as “explanation”, and is a topic of increasing importance both search applications and recommender systems.

## Commercial Application of Watson

In any application Watson can make use of all available information sources, e.g. an application to support medical diagnostics can include published reports and papers together with knowledge structures such as taxonomies and ontologies.

In general, include domain specific databases, ontologies and taxonomies, as well as domain specific unstructured sources, books, articles, specialist web content, etc.

Its knowledge-base can be continuously updated during operation by ingesting new content as it appears.

# Google Knowledge Graph

Google has build an extensive knowledge graph based on content harvested from the web.

The Google knowledge graph contains Information about and relationships between hundreds of millions of objects or “entities’, and contains billions of facts.

Many search queries focus on entities, and Google uses its knowledge graph to enrich the information returned to the searcher for these queries.

The standard SERP including ranked document snippets is preserved, and augmented with additional information.

# Google Knowledge Graph

The size of the Google knowledge graph is such that its construction is automated automated. This includes:

- information extraction from web documents
- integration of information into the knowledge graph

Automated web-based question answering is used to perform targeted completion of missing facts.

What questions to ask? Identify suitable questions for standard attributes of objects.

e.g. What is/was name of x’s father, etc.

# Google Knowledge Graph

The automation of this process means that it can be incomplete and/or make mistakes.

“prior” knowledge in manually constructed taxonomies and ontologies and existing smaller knowledge graphs can be used to can be used to reduce potential error rate.

Google “Wherever we can get our hands on structured data, we add it!”

Additionally ask people for missing information.

- use *crowdsourcing*.

## Google Information “Cards”

Google use their knowledge graph to construct summary ‘cards’ about searched for entities.

Cards are displayed when a significant entity appears in the search query.

Google includes facts for each entities which are of most interest for that object.

“People also search for” are on the card - lists related people, places, etc. - related entities that people search for when searching for the current entity.

Knowledge cards have “report a problem” - the data may be incorrect.

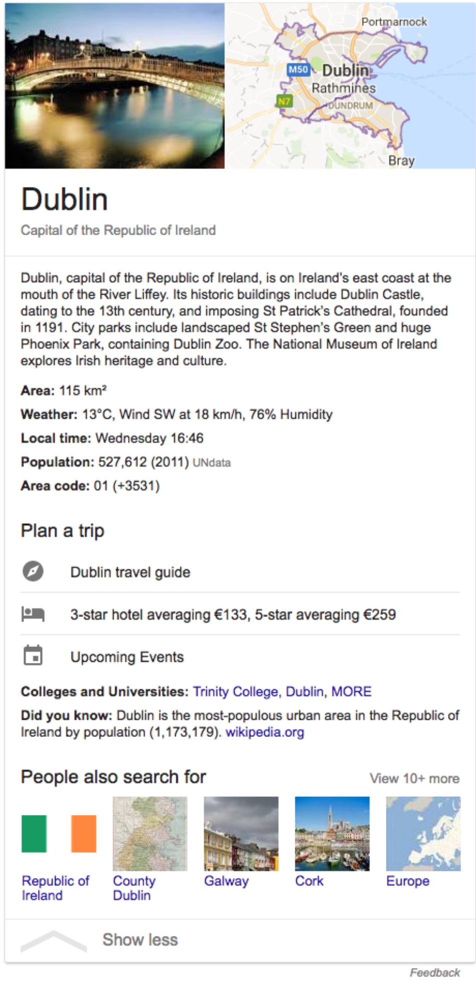
Do cards reduce searching, and therefore click based revenue? Interestingly, they find that the cards encourage more searching, and thus maintain clicks through to websites.

# Example Cards

Google

query:

“Dublin”



Dublin

Capital of the Republic of Ireland

Dublin, capital of the Republic of Ireland, is on Ireland's east coast at the mouth of the River Liffey. Its historic buildings include Dublin Castle, dating to the 13th century, and imposing St Patrick's Cathedral, founded in 1191. City parks include landscaped St Stephen's Green and huge Phoenix Park, containing Dublin Zoo. The National Museum of Ireland explores Irish heritage and culture.

**Area:** 115 km²

**Weather:** 13°C, Wind SW at 18 km/h, 76% Humidity

**Local time:** Wednesday 16:46

**Population:** 527,612 (2011) UNdata

**Area code:** 01 (+3531)

Plan a trip

Dublin travel guide

3-star hotel averaging €133, 5-star averaging €259

Upcoming Events

**Colleges and Universities:** Trinity College, Dublin, MORE

**Did you know:** Dublin is the most-populous urban area in the Republic of Ireland by population (1,173,179). wikipedia.org

People also search for View 10+ more

Republic of Ireland
County Dublin
Galway
Cork
Europe

Show less

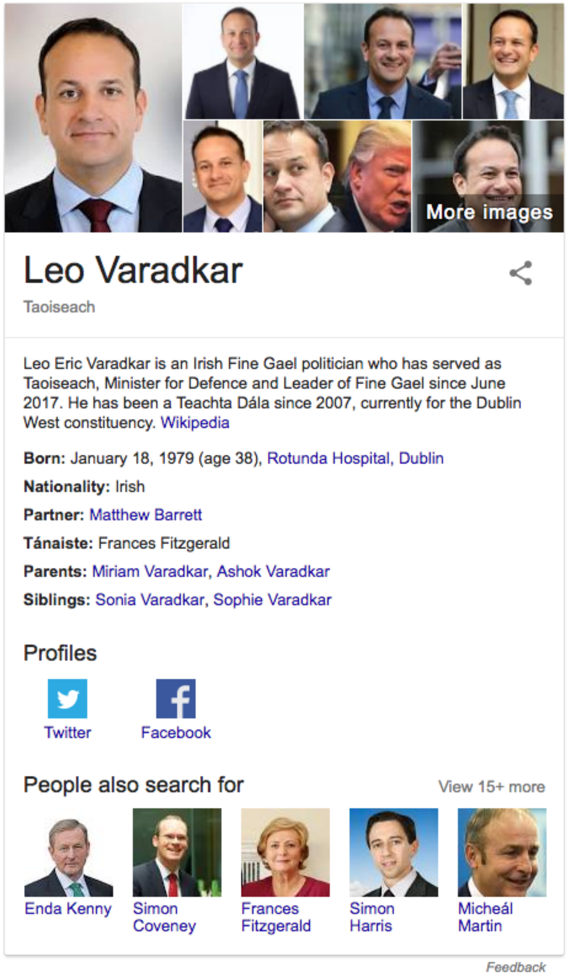
Feedback

Google

query:

“Leo

Varadkar”



Leo Varadkar

Taoiseach

Leo Eric Varadkar is an Irish Fine Gael politician who has served as Taoiseach, Minister for Defence and Leader of Fine Gael since June 2017. He has been a Teachta Dála since 2007, currently for the Dublin West constituency. Wikipedia

**Born:** January 18, 1979 (age 38), Rotunda Hospital, Dublin

**Nationality:** Irish

**Partner:** Matthew Barrett

**Tánaiste:** Frances Fitzgerald

**Parents:** Miriam Varadkar, Ashok Varadkar

**Siblings:** Sonia Varadkar, Sophie Varadkar

Profiles

Twitter
Facebook

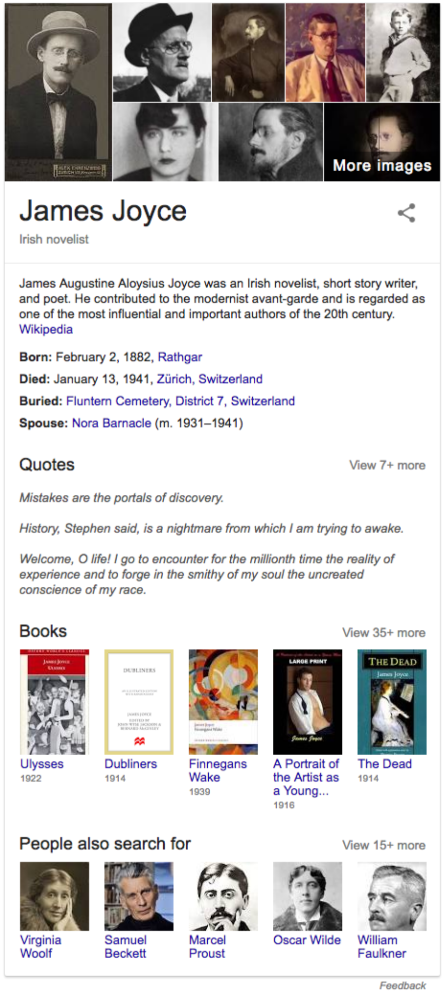
People also search for View 15+ more

Enda Kenny
Simon Coveney
Frances Fitzgerald
Simon Harris
Micheál Martin

Feedback

# Example Cards

Google query: “James Joyce”



James Joyce

Irish novelist

James Augustine Aloysius Joyce was an Irish novelist, short story writer, and poet. He contributed to the modernist avant-garde and is regarded as one of the most influential and important authors of the 20th century. Wikipedia

**Born:** February 2, 1882, Rathgar

**Died:** January 13, 1941, Zürich, Switzerland

**Buried:** Fluntern Cemetery, District 7, Switzerland

**Spouse:** Nora Barnacle (m. 1931–1941)

Quotes

*Mistakes are the portals of discovery.*

*History, Stephen said, is a nightmare from which I am trying to awake.*

*Welcome, O life! I go to encounter for the millionth time the reality of experience and to forge in the smithy of my soul the uncreated conscience of my race.*

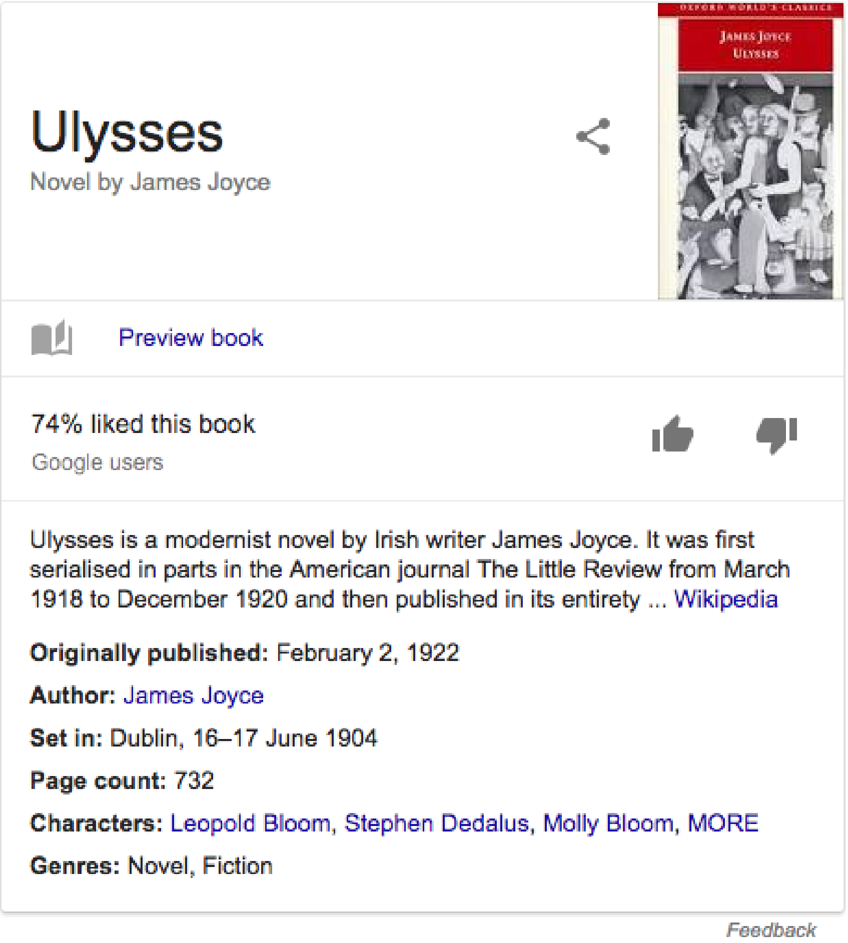
Books

- Ulysses 1922
- Dubliners 1914
- Finnegans Wake 1939
- A Portrait of the Artist as a Young... 1916
- The Dead 1914

People also search for

- Virginia Woolf
- Samuel Beckett
- Marcel Proust
- Oscar Wilde
- William Faulkner

Google query: “Ulysses”



Ulysses

Novel by James Joyce

74% liked this book

Google users

Ulysses is a modernist novel by Irish writer James Joyce. It was first serialised in parts in the American journal The Little Review from March 1918 to December 1920 and then published in its entirety ... Wikipedia

**Originally published:** February 2, 1922

**Author:** James Joyce

**Set in:** Dublin, 16–17 June 1904

**Page count:** 732

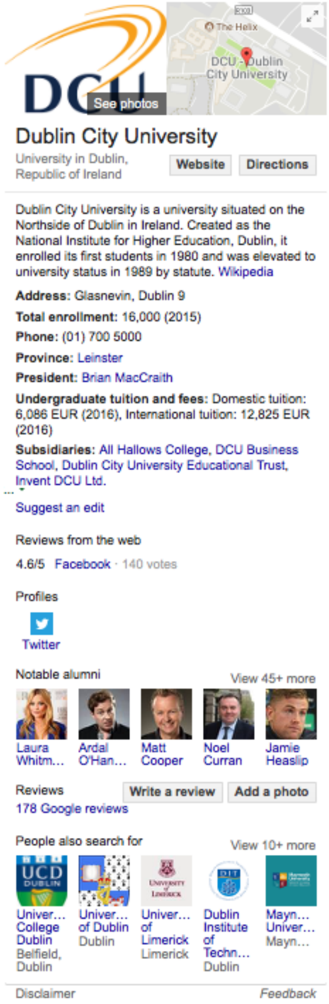
**Characters:** Leopold Bloom, Stephen Dedalus, Molly Bloom, MORE

**Genres:** Novel, Fiction

# Example Cards

Google query:

“Dublin City University”



Dublin City University

University in Dublin, Republic of Ireland

Dublin City University is a university situated on the Northside of Dublin in Ireland. Created as the National Institute for Higher Education, Dublin, it enrolled its first students in 1980 and was elevated to university status in 1989 by statute. Wikipedia

**Address:** Glasnevin, Dublin 9

**Total enrollment:** 16,000 (2015)

**Phone:** (01) 700 5000

**Province:** Leinster

**President:** Brian MacCraith

**Undergraduate tuition and fees:** Domestic tuition: 6,086 EUR (2016), International tuition: 12,825 EUR (2016)

**Subsidiaries:** All Hallows College, DCU Business School, Dublin City University Educational Trust, Invent DCU Ltd.

Suggest an edit

Reviews from the web

4.6/5 Facebook · 140 votes

Profiles

Twitter

Notable alumni — View 45+ more

Laura Whitm… · Ardal O'Han… · Matt Cooper · Noel Curran · Jamie Heaslip

Reviews

178 Google reviews

People also search for — View 10+ more

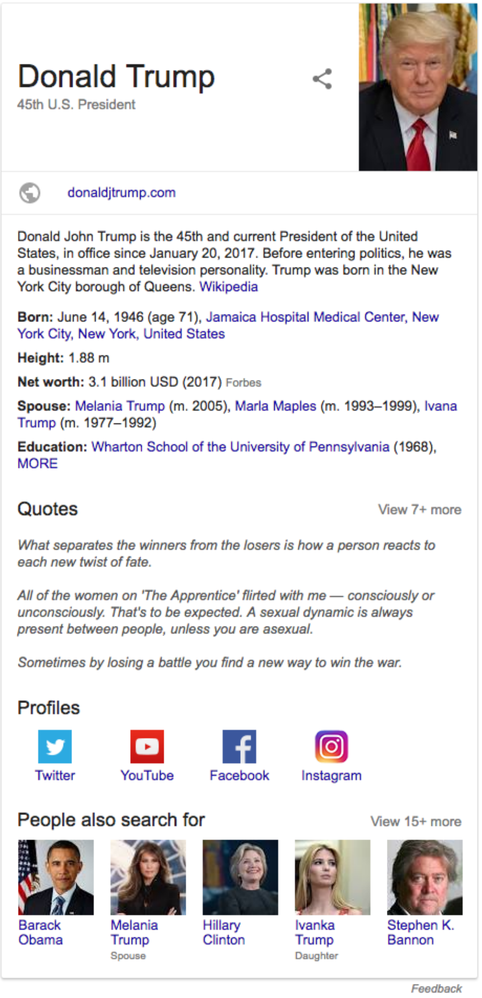
Univer… College Dublin, Belfield, Dublin · Univer… of Dublin, Dublin · Univer… of Limerick, Limerick · Dublin Institute of Techn…, Dublin · Mayn… Univer…, Mayn…

Disclaimer

*Feedback*

# Example Cards

Google query: “Donald Trump”



Donald Trump

45th U.S. President

donaldjtrump.com

Donald John Trump is the 45th and current President of the United States, in office since January 20, 2017. Before entering politics, he was a businessman and television personality. Trump was born in the New York City borough of Queens. Wikipedia

**Born:** June 14, 1946 (age 71), Jamaica Hospital Medical Center, New York City, New York, United States

**Height:** 1.88 m

**Net worth:** 3.1 billion USD (2017) Forbes

**Spouse:** Melania Trump (m. 2005), Marla Maples (m. 1993–1999), Ivana Trump (m. 1977–1992)

**Education:** Wharton School of the University of Pennsylvania (1968), MORE

Quotes View 7+ more

*What separates the winners from the losers is how a person reacts to each new twist of fate.*

*All of the women on 'The Apprentice' flirted with me — consciously or unconsciously. That's to be expected. A sexual dynamic is always present between people, unless you are asexual.*

*Sometimes by losing a battle you find a new way to win the war.*

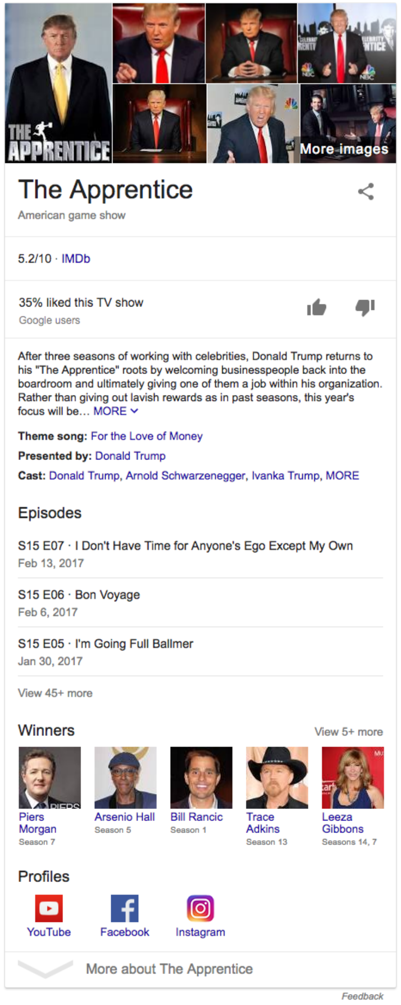
Profiles

People also search for View 15+ more

# Example Cards

But curiously for the Google query:

“Donald Trump apprentice”



The Apprentice

American game show

5.2/10 · IMDb

35% liked this TV show

Google users

After three seasons of working with celebrities, Donald Trump returns to his "The Apprentice" roots by welcoming businesspeople back into the boardroom and ultimately giving one of them a job within his organization. Rather than giving out lavish rewards as in past seasons, this year's focus will be… MORE

**Theme song:** For the Love of Money

**Presented by:** Donald Trump

**Cast:** Donald Trump, Arnold Schwarzenegger, Ivanka Trump, MORE

Episodes

S15 E07 · I Don't Have Time for Anyone's Ego Except My Own

Feb 13, 2017

S15 E06 · Bon Voyage

Feb 6, 2017

S15 E05 · I'm Going Full Ballmer

Jan 30, 2017

View 45+ more

Winners View 5+ more

| Piers Morgan | Arsenio Hall | Bill Rancic | Trace Adkins | Leeza Gibbons |
|---|---|---|---|---|
| Season 7 | Season 5 | Season 1 | Season 13 | Seasons 14, 7 |

Profiles

YouTube Facebook Instagram

More about The Apprentice

*Feedback*

# Addressing Query-Document Mismatch in Search

As we noted earlier, mismatch between words and phrases meaning the same thing appearing in queries and documents, can negatively impact on retrieval effectiveness.

Representing words and phrases in a form that captures their semantic relationship could potentially overcome this mismatch problem.

*Word embedding* methods enable words and phrases to be described in terms of vectors representing their shared properties.

Comparing *word vectors* created using word embedding methods allows their semantic properties to be compared, potentially overcoming the mismatch problems.

## Word Embedding

Word embedding creates a fixed size vector representing the properties of each word in terms of the same vector features.

The vector representation of each word has size and direction.

The angle between the vector representations of the word represents the semantic similarity between the words.

The vector for each word is trained automatically using machine learning with a large corpus of text representative of the data for which the word vectors are to be used.

The elements of the vector for each word are trained using the contexts in which the word is observed in the training text, i.e. the words surrounding it.

- semantically similar words are typically found to be surrounded by the same words and thus will have similar vectors.

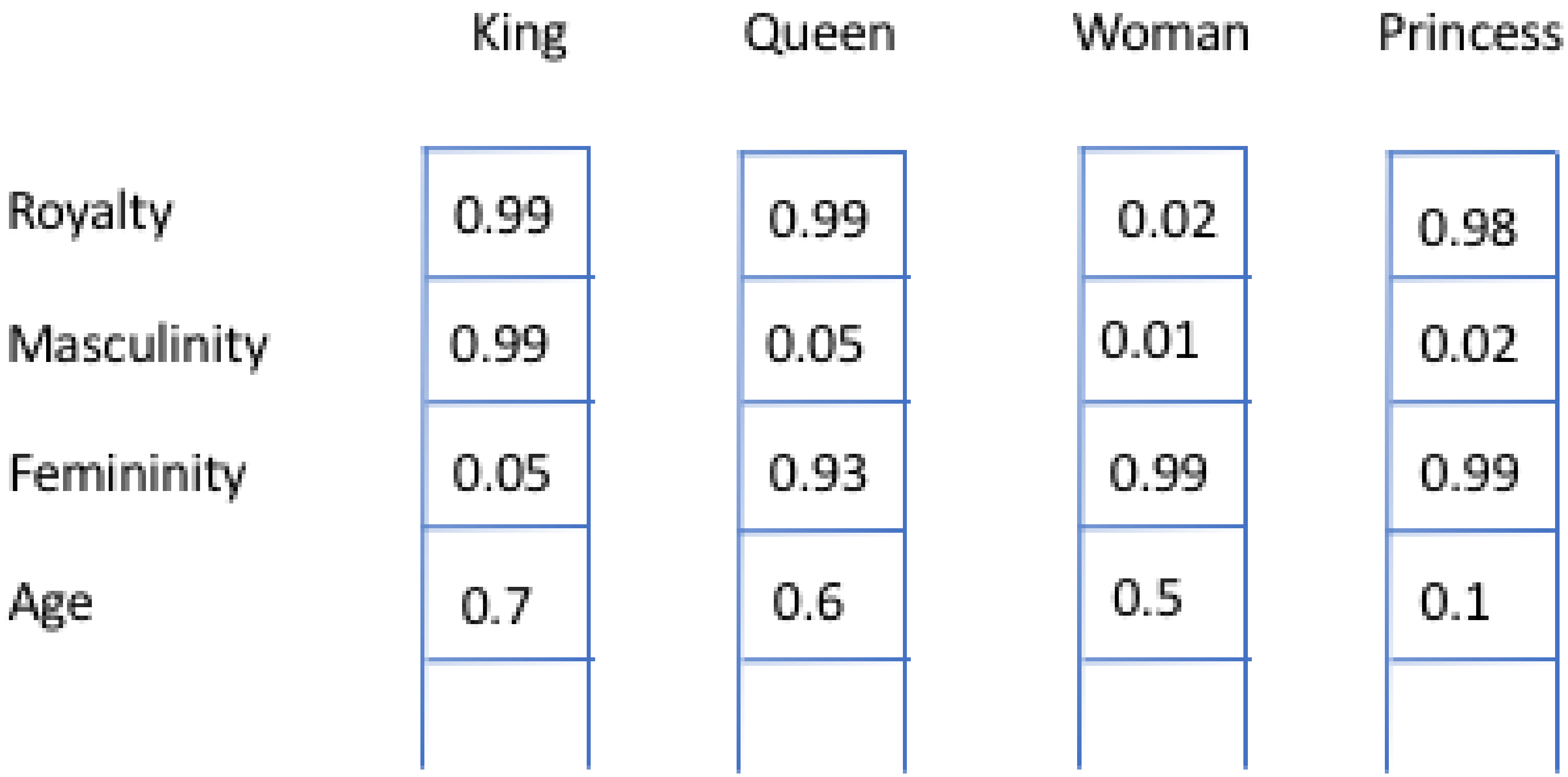
# Word Embedding

Each word is represented by real value weights in the vector.

The actual elements of the vector do not directly represent properties of the word since they are trained automatically from large collections of text corpora by the machine learning algorithm.

So, the example below is intended only to represent the principle of what is going on. We don’t actually label the elements of the vector “royalty”, “masculinity”, etc.

# Word Embedding



| | King | Queen | Woman | Princess |
|---|---|---|---|---|
| Royalty | 0.99 | 0.99 | 0.02 | 0.98 |
| Masculinity | 0.99 | 0.05 | 0.01 | 0.02 |
| Femininity | 0.05 | 0.93 | 0.99 | 0.99 |
| Age | 0.7 | 0.6 | 0.5 | 0.1 |

## Word Embedding in Information Retrieval

So how can this be applied to nformation retrieval and query-document matching?

The similarity between vectors for a word $w_i$ and a word $w_k$ can be computed using cosine similarity.

$$\cos\theta = \frac{\mathbf{w_i.w_k}}{|w_i||w_k|}$$

Thus, if $w_i$ appears in the query and $w_k$ appears in a document, we can incorporate the semantic similarity between $w_i$ and $w_k$ in the form of $\cos\theta$ in the query-document matching score.

Thus, by incorporating semantic representation at the word level, we are able to address the query-document mismatch problem when the query and document contain different words with strongly related semantic meanings.

## Word Embedding in Information Retrieval

If term $k$ appears in the query and term $i$ in a document, a revised term weight for term $i$ in document $j$ could be

$$w(i_{doc}, k_{query}) = w(i, j) \times \cos\theta$$

Here the term weight is effectively reduced according to the level of similarity between terms $i$ and $k$.

Implementing such an extended term matching scheme into an IR system would require some modifications and extensions to the inverted file based matching scheme described in the section on Text Retrieval.

## Word Embedding in NLP

As well as information retrieval, word embedding and other vector-based representation methods are currently very popular in many areas of natural language processing (NLP), including:

- natural language understanding
- machine translation
- automatic speech recognition
- conversation-based interfaces