빅데이터로 지능적 처리에 날개를 달다

Overview

- Big data
- Text classification
- How to handle dark data?

Big data

Big data ≈ 3V

[Volume]
parallel computing을 활용해서 처리
- data partitioning (*****)
- process partitioned data in parallel
- merge processed data (*)

Making big data smaller (sampling)
- approximate computing

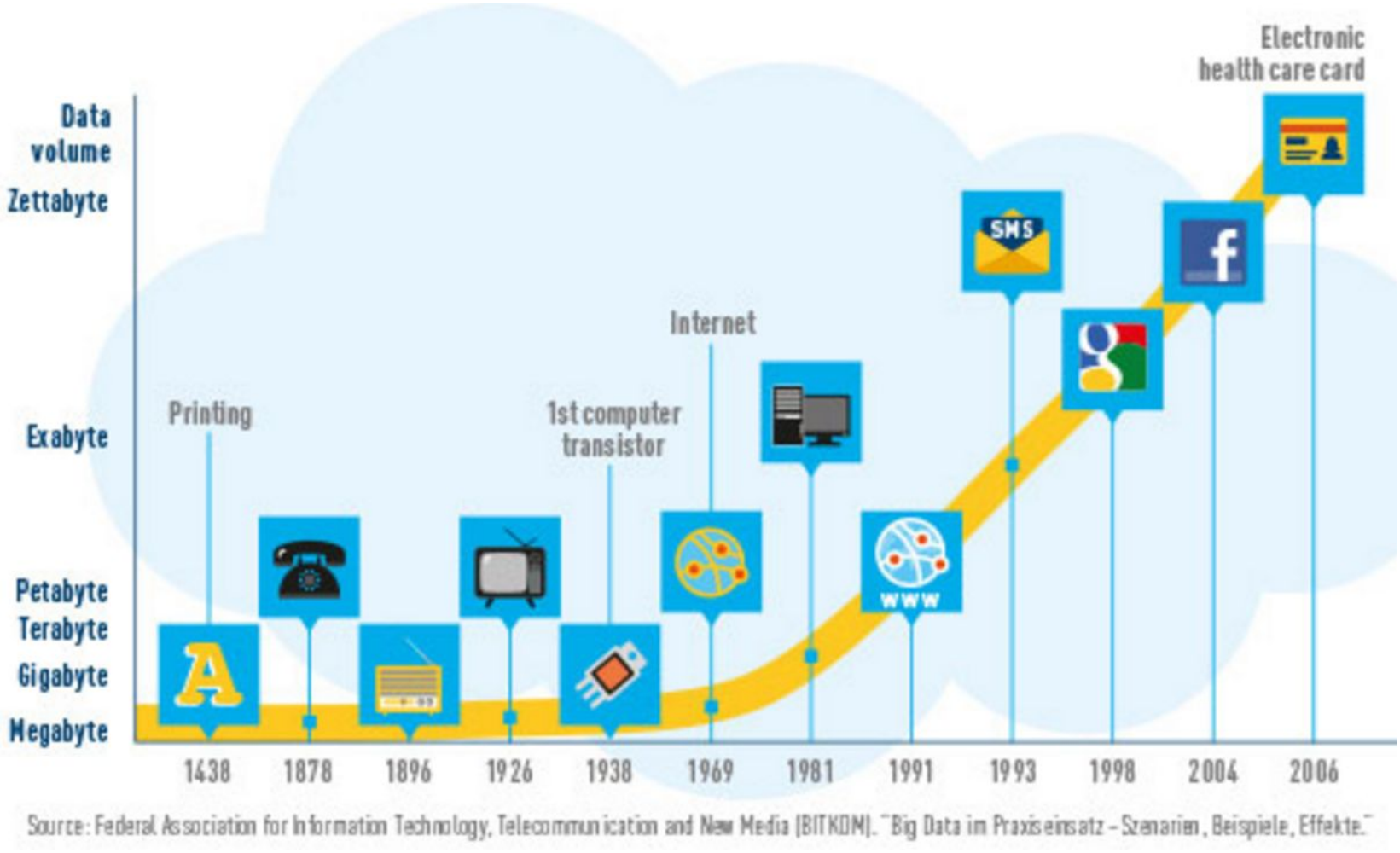
when computation cost is extremely high and data is not that big
- replicate data for each machine
- making computation fine-granule

[Velocity]
결국 volume과 함께 parallelism과 관련한 것
얼마나 적은 비용으로 큰 volume과 velocity를 관리하는 것

[Variety]
AI technology (knowledge graph, large language model)

Volume

하나의 컴퓨터로 처리할 수 없는 데이터
과거의 무어의 법칙은 더이상 적용되지 않음
(무어의 법칙: 18~24개월 마다 처리할 수 있는 데이터 분량이 두배씩 증가한다는 법칙, 최근에는 병렬화가 중요해짐)



What if data does not fit in memory?

sw, hw 모두 개발되어야 함.

hw로 병렬화 하기 위한 두 가지 방법
macro: 컴퓨터 대수를 늘리는 것
micro: 한 대의 컴퓨터로 병렬화

- Make algorithms scalable!
- Exploit all hardware resources at the same time
  - Macro parallelism (Scale-out)
    - Cloud computing
    - Infiniband 고속 네트워킹
  - Micro parallelism (Scale-up)
    - Multi-core (Multi-core CPUs and GPUs)
    - NUMA
    - Parallel I/Os

non uniform memory access: 메모리 액세스 비용이 다름 → CPU에서 코어가 있는 공간을 독립적인 구획으로 나눠 메모리 할당. 메모리 설계 방법. s/w도 numa architecture에 잘 맞게 디자인되어야 함.

Latency Numbers를 알면 어느 정도 latency 이내에 sw를 제공할 수 있는지 계산할 수 있음
scalable 알고리즘으로 sw를 고도화하고, 병렬화한 hw를 사용할수록 점점 성능이 높아질 수 밖에 없음.
캐시와 메모리를 어떻게 활용하느냐에 따라 속도 차이가 매우 큼.

Scalable matrix factorization

	Avatar	*The Matrix*	*Up*
Alice	?	4	2
Bob	3	2	?
Charlie	5	?	3

Matrix Factorization (MF)

	Avatar	*The Matrix*	*Up*
Alice	?	4	2
Bob	3	2	?
Charlie	5	?	3

Rating matrix R

$\mathbb{R}^{n \times m}$

$\mathbb{R}^{k \times n}$

User factor matrix P^T

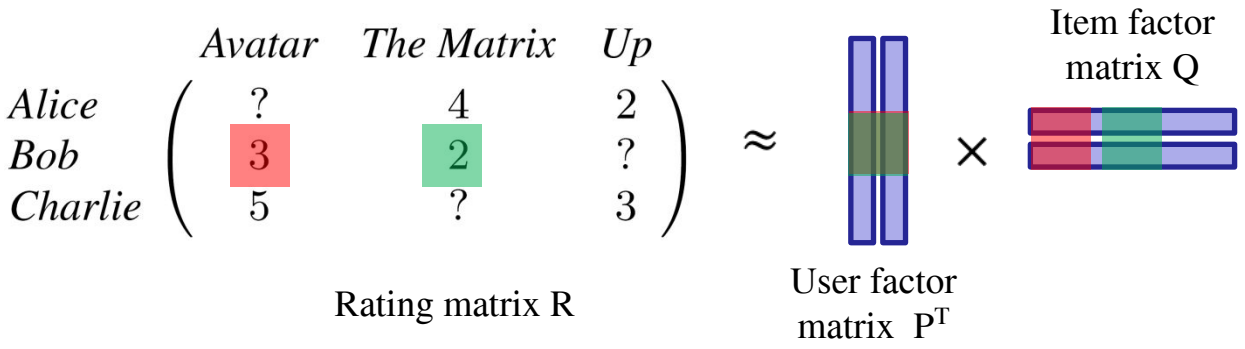
$\mathbb{R}^{k \times m}$

Item factor matrix Q

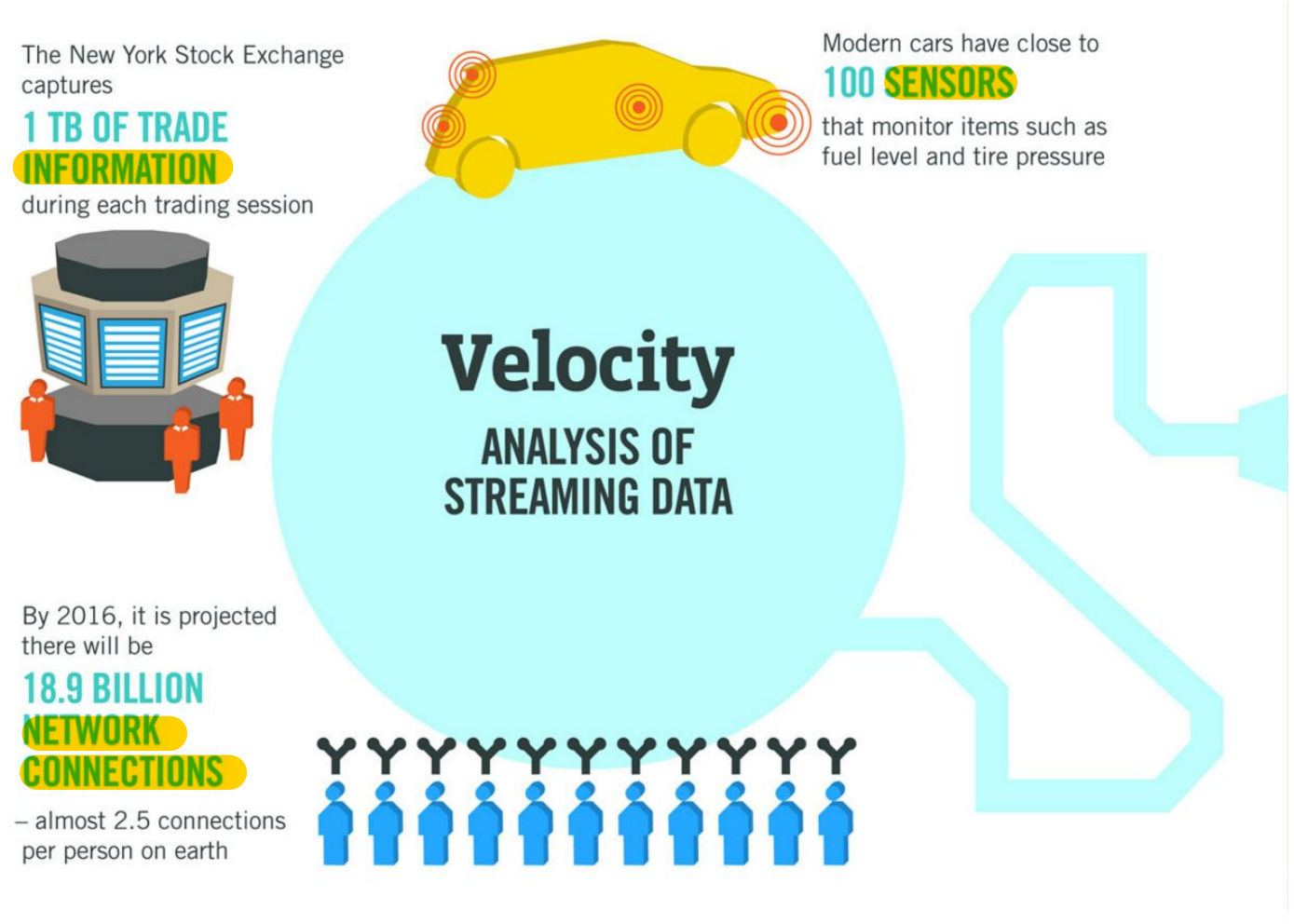
Oh, J., Han, W., Yu, H., and Jiang, X., "Fast and Robust Parallel SGD Matrix Factorization," In *KDD 2015*.

SGD for MF

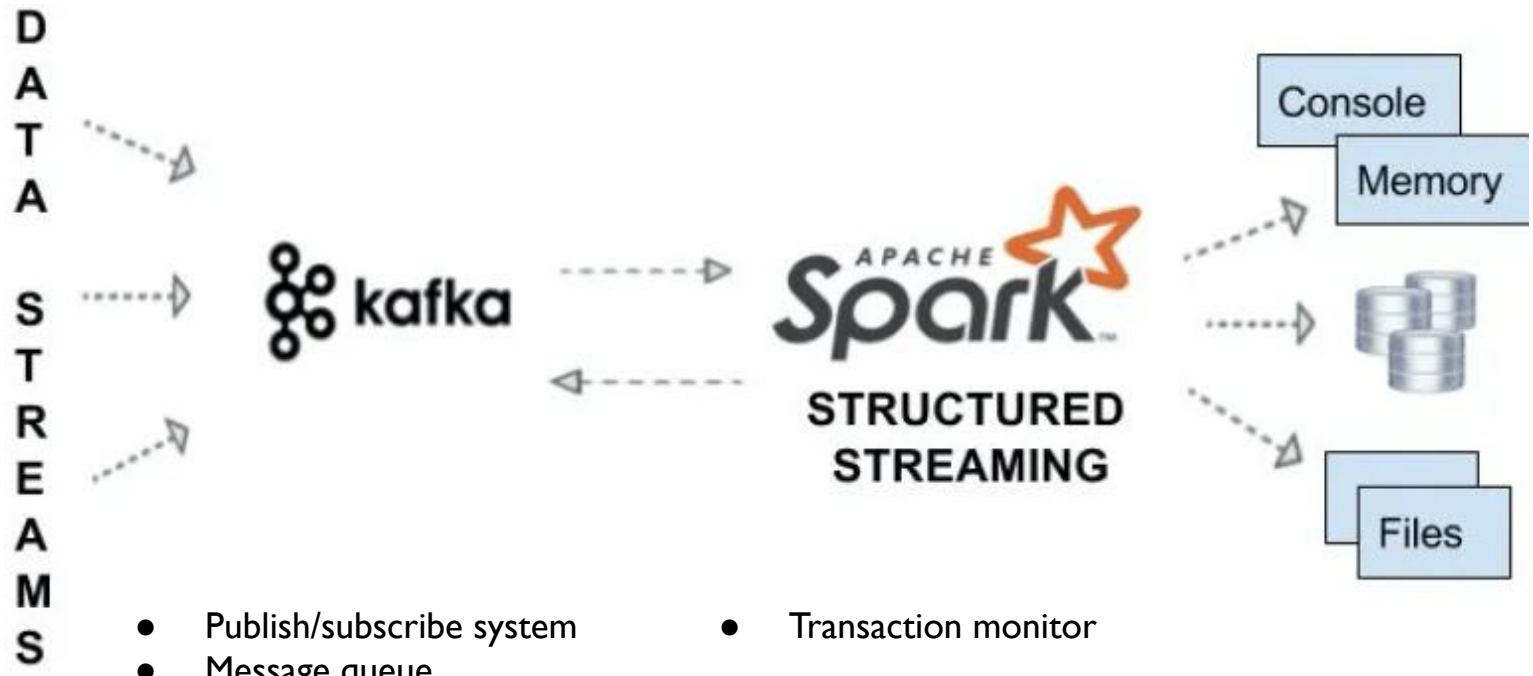
- SGD gradually updates factor matrices by repeating following two steps
  1. Randomly pick a rating data
  2. Update corresponding vectors in factor matrices



Velocity


The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session
Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure
Velocity
ANALYSIS OF STREAMING DATA
By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth
Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

stateless vs. stateful
스트리밍 프로세스에서는 데이터를 기억해야 하는지, 삭제해도 되는지에 따라
관리 방법이 달라짐



- Publish/subscribe system
- Message queue

데이터 피드의 분산 스트리밍, 파이프 라이닝,
실시간 데이터 처리 등을 가능하게 하는 분산형
publish/subscribe system
고속으로 데이터를 처리하기 위해 활용

- Transaction monitor

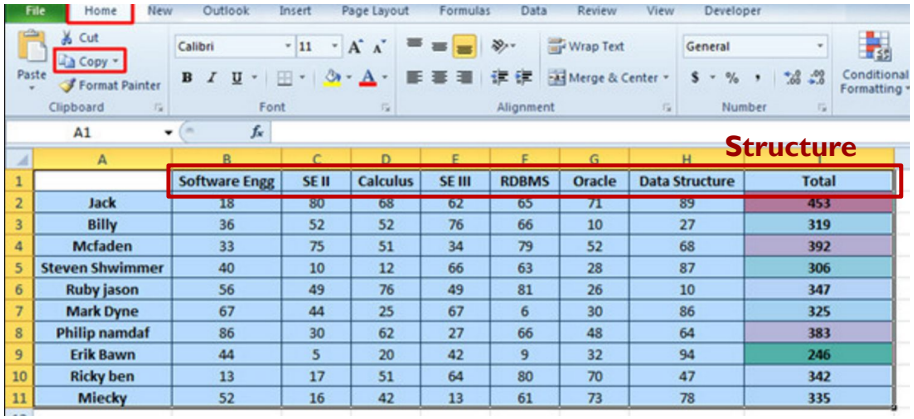
DBMS(데이터 베이스 관리 시스템)로 보내기 전에
데이터를 구조적으로 관리

Variety

- Excel sheets, relational data, csv files Structured data
- XML, JSON, graph Semi-structured data
- Text documents, web documents, images, audios, videos Unstructured data

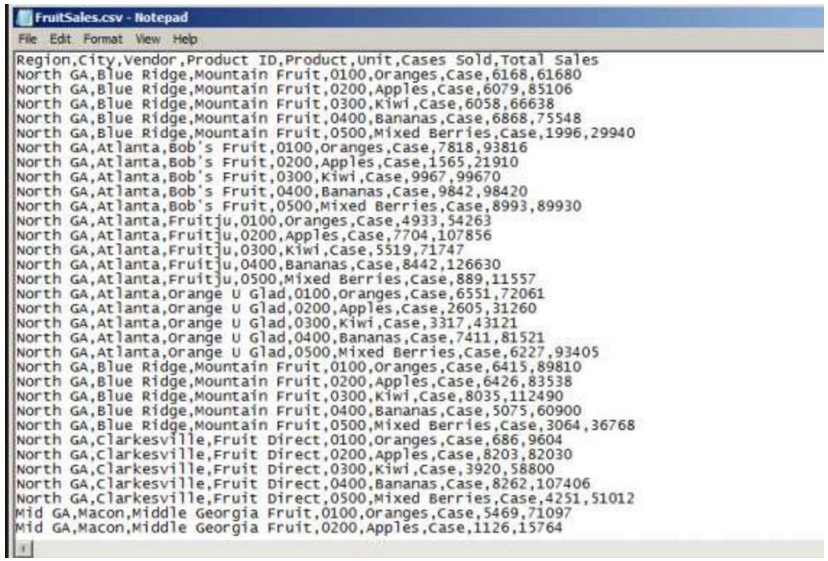
schema(데이터의 논리적 구조)가 있는 data
table에 있는 모든 데이터가 같은 schema를 가짐

Structured data
(= tables)



	A	B	C	D	E	F	G	H	I
1		Software Engg	SE II	Calculus	SE III	RDBMS	Oracle	Data Structure	Total
2	Jack	18	80	68	62	65	71	89	453
3	Billy	36	52	52	76	66	10	27	319
4	Mcfaden	33	75	51	34	79	52	68	392
5	Steven Shwimmer	40	10	12	66	63	28	87	306
6	Ruby jason	56	49	76	49	81	26	10	347
7	Mark Dyne	67	44	25	67	6	30	86	325
8	Philip namdaf	86	30	62	27	66	48	64	383
9	Erik Bawn	44	5	20	42	9	32	94	246
10	Ricky ben	13	17	51	64	80	70	47	342
11	Miecky	52	16	42	13	61	73	78	335

Excel file



CSV file

Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Participants Table

ID*	Activity*
084	Tennis
084	Swimming
100	Squash
100	Swimming
182	Tennis
219	Golf
219	Swimming
219	Squash

Activities Table

Activity*	Cost
Golf	$47
Sailing	$50
Squash	$40
Swimming	$15
Tennis	$36

Relational database

Semi-structured data

데이터마다(table 마다) schema를 가지고 있지만 schema가 다를 수 있음

```
<? XML VERSION = "1.0" STANDALONE = "yes" ?>
<STAR-MOVIE-DATA>
    <STAR><NAME>Carrie Fisher</NAME>
        <ADDRESS><STREET>123 Maple St.</STREET>
            <CITY>Hollywood</CITY></ADDRESS>
        <ADDRESS><STREET>5 Locust Ln.</STREET>
            <CITY>Malibu</CITY></ADDRESS>
    </STAR>
    <STAR><NAME>Mark Hamill</NAME>
        <STREET>456 Oak Rd.</STREET><CITY>Brentwood</CITY>
    </STAR>
    <MOVIE><TITLE>Star Wars</TITLE><YEAR>1977</YEAR>
    </MOVIE>
</STAR-MOVIE-DATA>
```

schema

```
{ "users":[
        {
          "firstName":"Ray",
          "lastName":"Villalobos",
          "joined": {
              "month":"January",
              "day":12,
              "year":2012
          }
        },
        {
          "firstName":"John",
          "lastName":"Jones",
          "joined": {
              "month":"April",
              "day":28,
              "year":2010
          }
        }
    ]}
```

XML JSON

(Mostly) tree structured data

CSV, JSON, XML

CSV

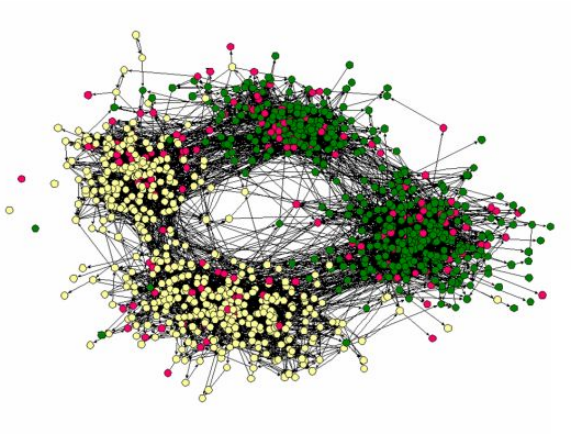
	A	B	C	D
1	ID	Gender	City	Monthly_I
2	ID000002C	Female	Delhi	20000
3	ID000004E	Male	Mumbai	35000
4	ID000007H	Male	Panchkula	22500
5	ID000008I	Male	Saharsa	35000
6	ID000009J	Male	Bengaluru	100000
7	ID000010K	Male	Bengaluru	45000
8	ID000011L	Female	Sindhudur	70000
9	ID000012M	Male	Bengaluru	20000
10	ID000013N	Male	Kochi	75000
11	ID000014C	Female	Mumbai	30000
12	ID000016C	Male	Mumbai	25000
13	ID000018S	Female	Surat	25000
14	ID000019T	Female	Pune	24000
15	ID000021V	Male	Bhubanes	27000
16	ID000022V	Female	Howrah	28000

JSON

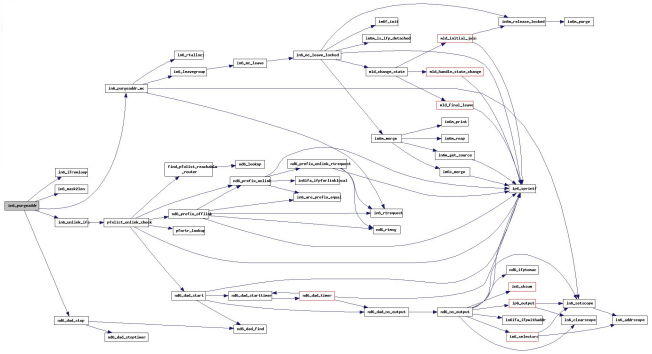
```
{
   "Employee": [

      {

         "id":"1",

         "Name": "Ankit",

         "Sal": "1000",

      },
      {

         "id":"2",

         "Name": "Faizy",
```

```
<?xml version="1.0"?>

<contact-info>

<name>Ankit</name>

<company>Anlytics Vidhya</company>

<phone>+9187654321</phone>

</contact-info>
```

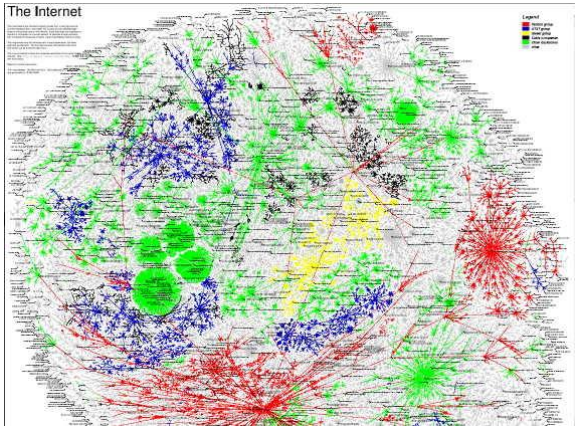
Graph data



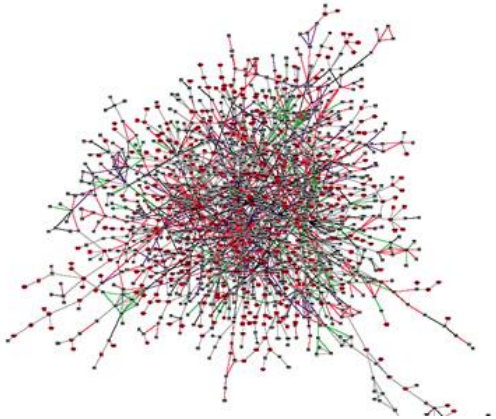
Friendship network



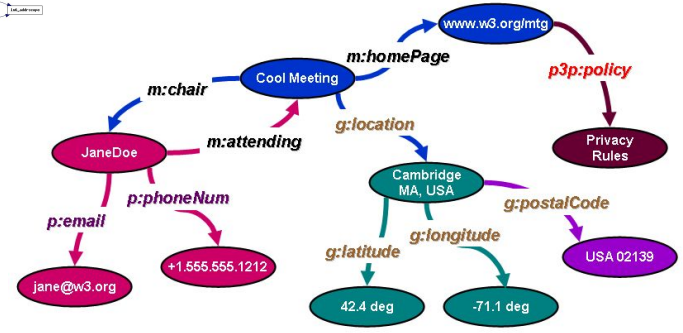
Call Graph



Internet map



Protein interactions



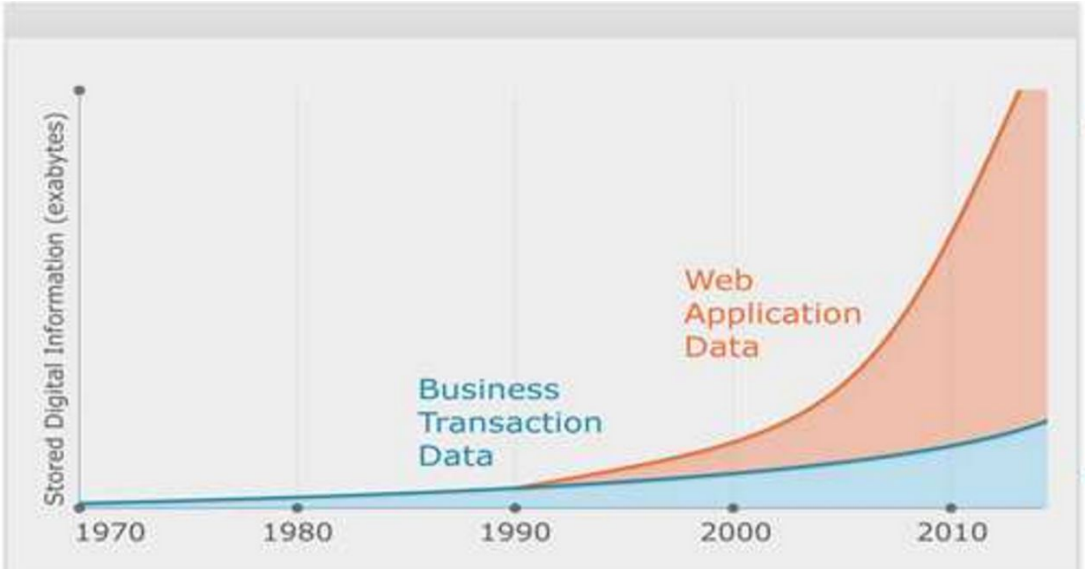
Semantic web

Everything can be easily represented as graphs

- Right!
- All structured and semi-structured data can be viewed as graphs

Unstructured data (≈ dark data)

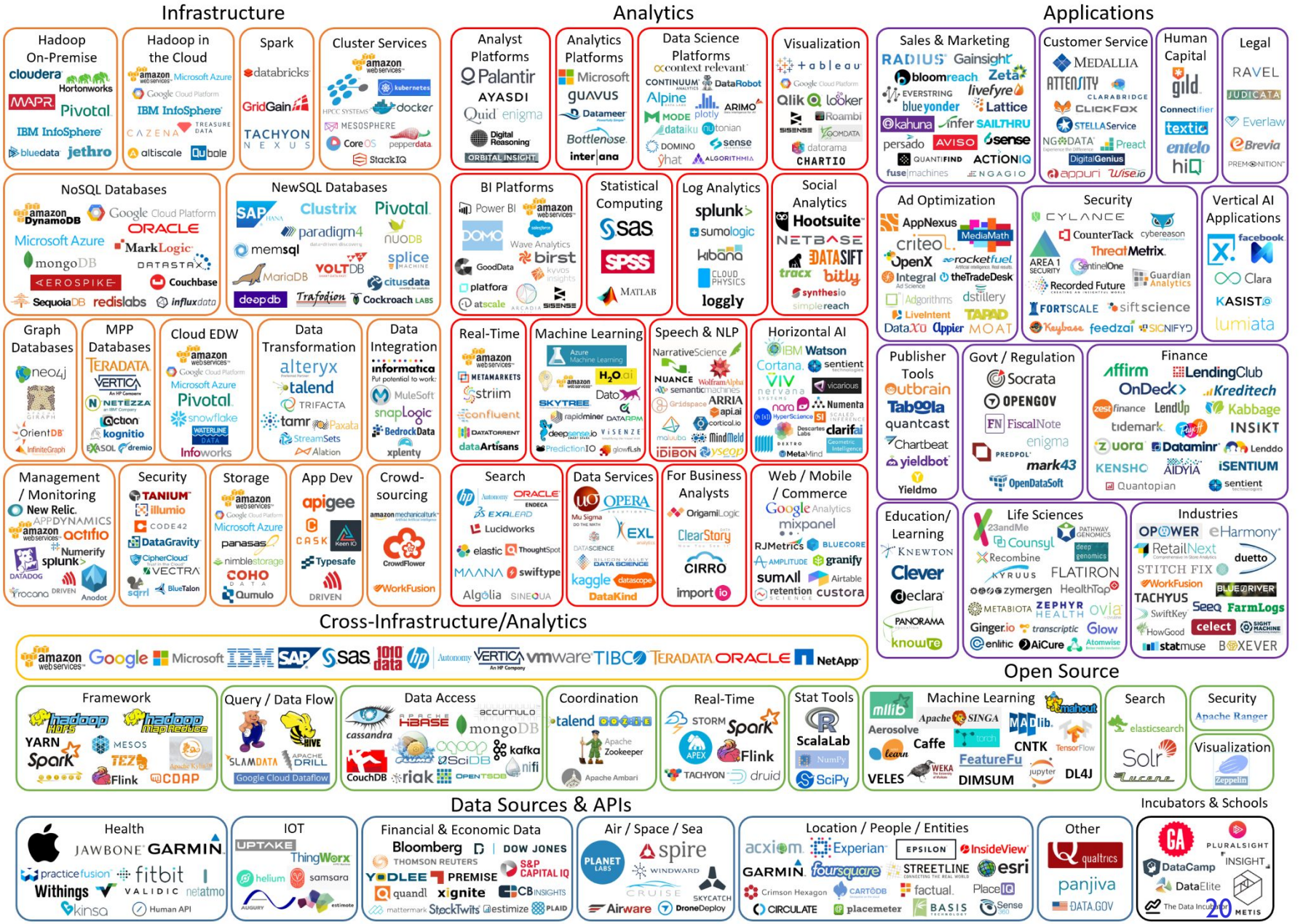
데이터는 존재하지만 활용되기 어려움. 형태가 너무 다양하기 때문.



→ Complex, Unstructured
- Text
- Images
- Audio
- Video …

→ Relational

Big Data Landscape 2016 (Version 3.0)



Last Updated 3/23/2016

FIRSTMARK

Infrastructure for Big data is very important

- But, we skip this part completely
  - Students who are interested in building systems love this part
- The rest of the talk focuses on handling variety of data

Text classification

Text classification

- Pop vs Heavy Metal
  - Given verse from lyrics, recognize genre

> "I'm a rolling thunder, a pouring rain
> I'm comin' on like a hurricane
> My lightning's flashing across the sky
> You're only young but you're gonna die
> I won't take no prisoners, won't spare no lives
> Nobody's putting up a fight
> I got my bell, I'm gonna take you to hell
> I'm gonna get you, Satan get you"

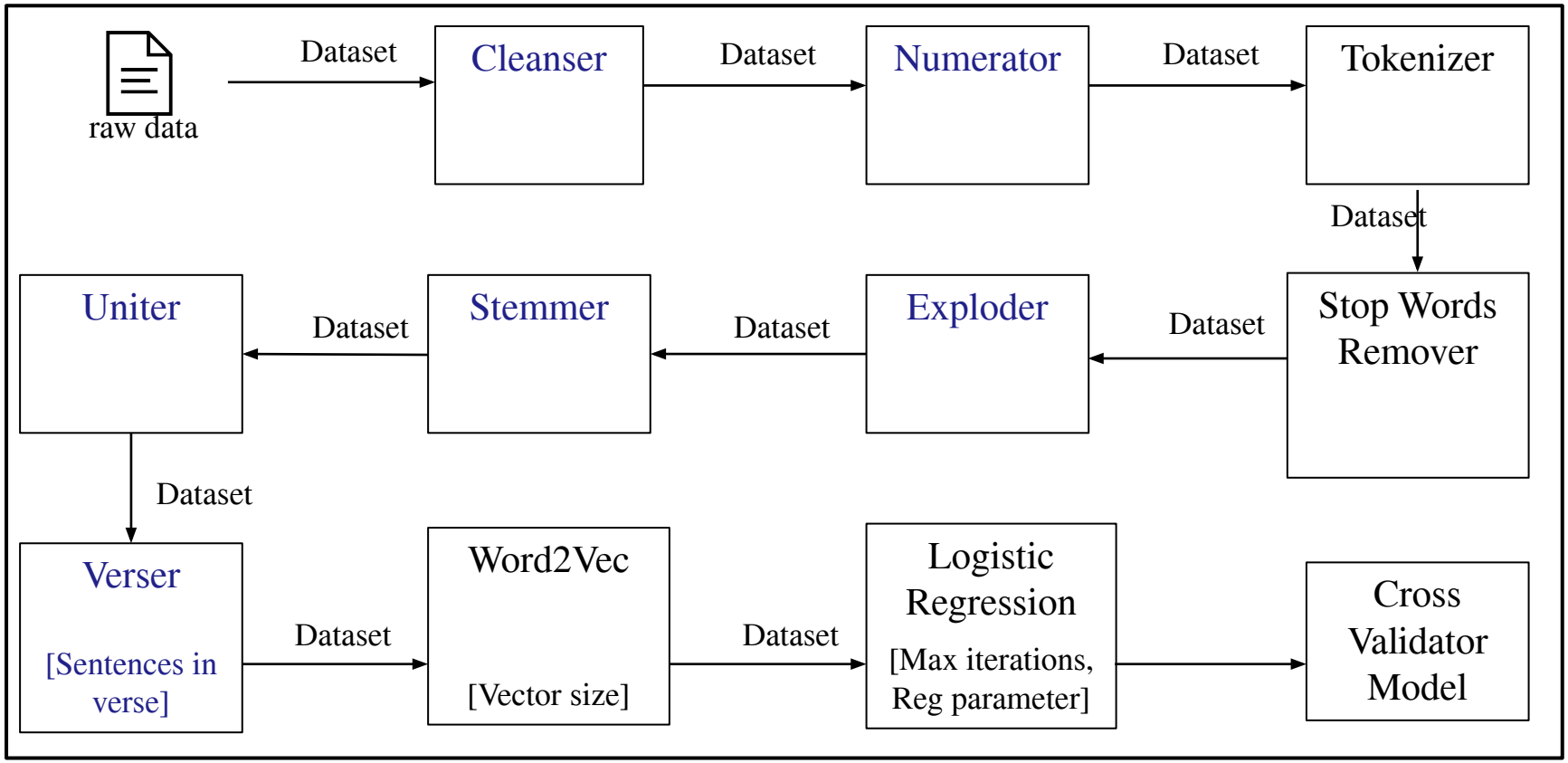
  - Collect raw data set of lyrics (~65k sentences)
    - [Pop] Abba, Ace of base, Backstreet Boys, Britney Spears, Christina Auilera, Madonna, etc.
    - [Heavy Metal] Black Sabbath, In Flames, Iron Maiden, Metallica, Moonspell, Nightwish, Sentences, etc.
  - Create training set, i.e. label (0|1) + features
  - Train logistic regression

Wait a minute!

- ML takes only numeric vectors as input!
- How about text documents?

Word2Vec

ML pipeline


raw data
Dataset
Cleanser
Dataset
Numerator
Dataset
Tokenizer
Dataset
Stop Words Remover
Dataset
Exploder
Dataset
Stemmer
Dataset
Uniter
Dataset
Verser
[Sentences in verse]
Dataset
Word2Vec
[Vector size]
Dataset
Logistic Regression
[Max iterations, Reg parameter]
Cross Validator Model

Cleanser

I'm a rolling thunder, a pouring rain
I'm comin' on like a hurricane
My lightning's flashing across the sky
You're only young but you're gonna die
I won't take no prisoners, won't spare no lives
Nobody's putting up a fight
I got my bell, I'm gonna take you to hell
I'm gonna get you, Satan get you

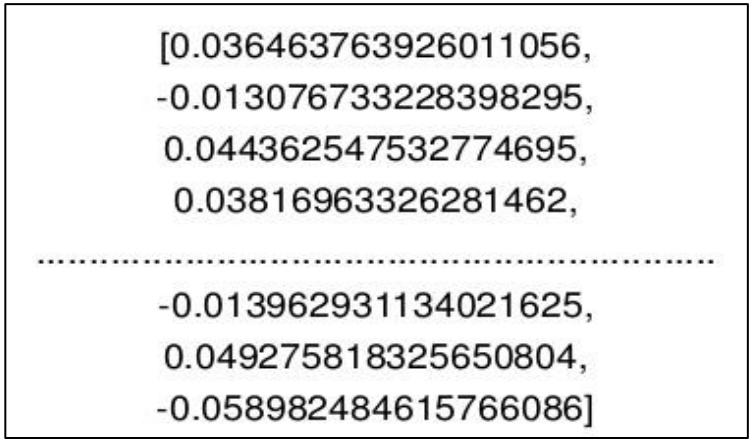
Im a rolling thunder a pouring rain
Im comin on like a hurricane
My lightnings flashing across the sky
Youre only young but youre gonna die
I wont take no prisoners wont spare no lives
Nobodys putting up a fight
I got my bell Im gonna take you to hell
Im gonna get you Satan get you

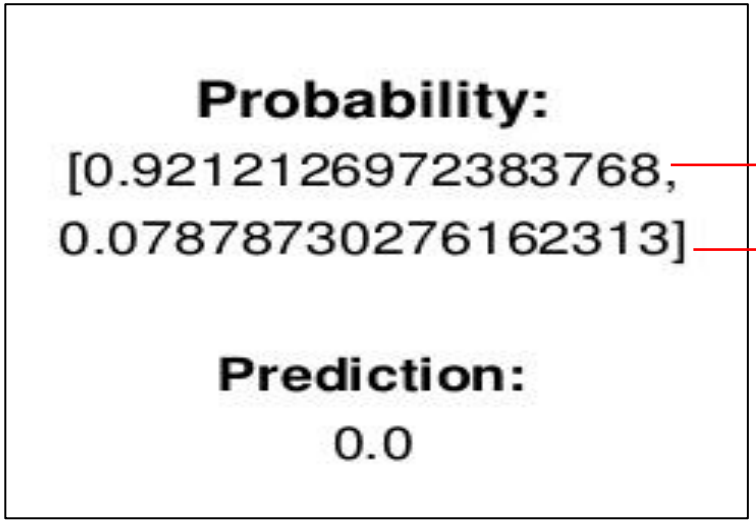
Stemmer

im rolling thunder pouring rain
im comin like hurricane
lightnings flashing across sky
youre young youre gonna die
wont take prisoners wont spare lives
nobodys putting fight
got bell im gonna take hell
im gonna get satan get

im roll thunder pour rain
im comin like hurrican
lightn flash across sky
your young your gonna die
wont take prison wont spare live
nobodi put fight
got bell im gonna take hell
im gonna get satan get

Word2Vec & Logistic regression





P(Heavy Metal)

P(Pop)

How to handle dark data?

Which Personal Assistant is the Smartest?

Here are the results of our research:

Personal Assistant	% Questions Answered	100% Complete & Correct
The Google Assistant on Google Home	68.1%	90.6%
Cortana **(MS)**	56.5%	81.9%
Siri	21.7%	62.2%
Alexa on the Amazon Echo	20.7%	87.0%

https://www.stonetemple.com/digital-personal-assistants-test

Do you know the secret of success?

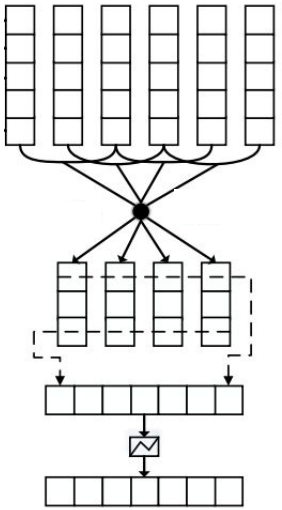
Job posting information extraction

unstructured data를 structured data로 바꾸는 작업이 필요함
> 일관된 방식으로 컨트롤하고 관리할 수 있음.

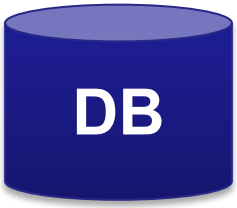
Title: Web Development Engineer
Location: Beaverton, Oregon
2017.04.21

This individual is responsible for design and implementation of the web-interfacing components of the AccessBase server, and general back-end development duties.
A successful candidate should have experience that includes:
One or more of: Solaris, Linux, IBM AIX, plus Windows/NT Programming in C/C++, Java
Database access and integration: Oracle, ODBC
CGI and scripting: one or more of Javascript, VBScript, Perl, PHP, ASP
Exposure to the following is a plus: JDBC, Flash/ Shockwave, FrontPage and/or Cold Fusion.
A BSCS and 2+ years experience (or equivalent) is required.

⇒



⇓ 정형화

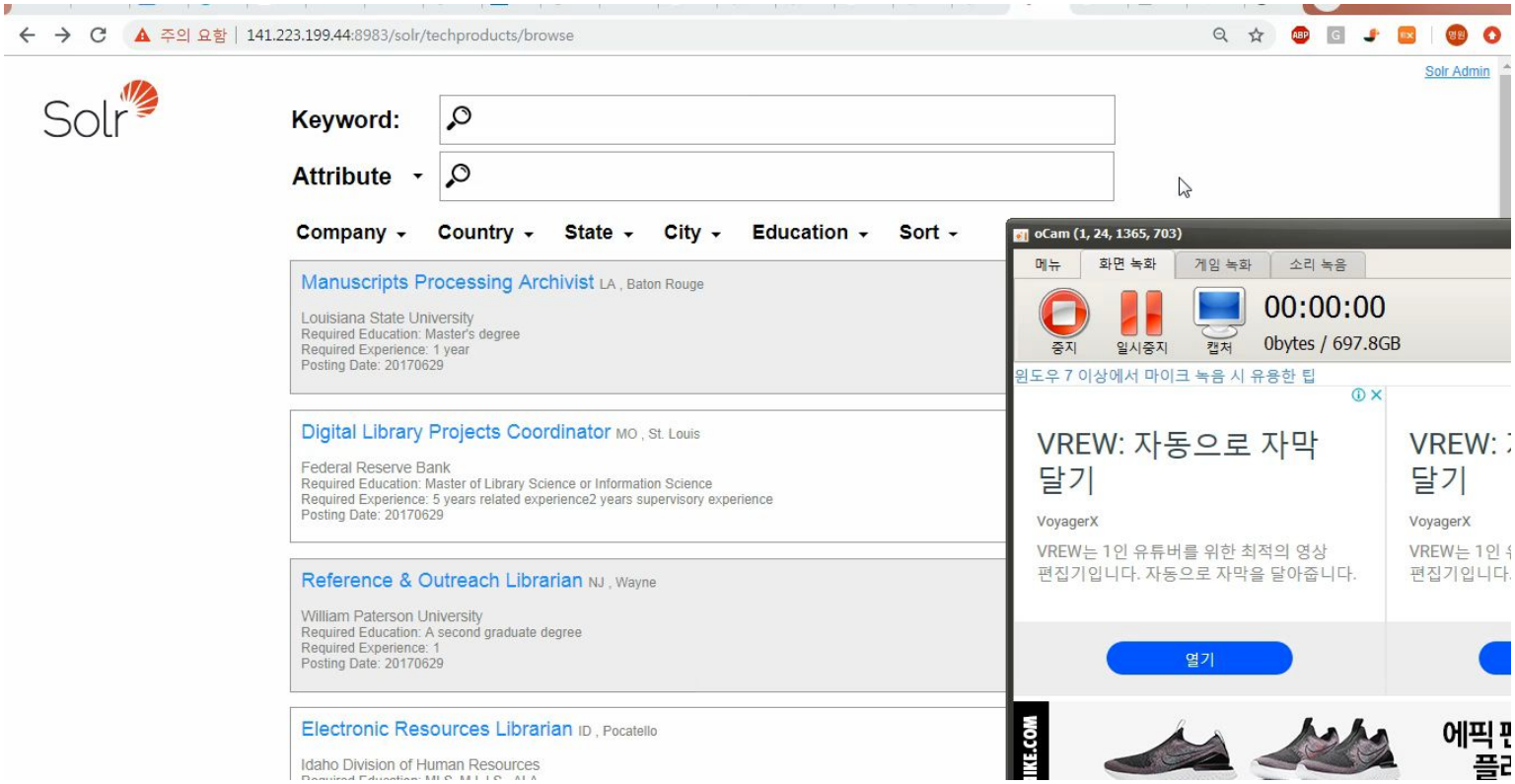


⇐

Title	Date	Location	Skills	…
Web Development Engineer	04-21-2017	Beaverton, Oragon	Windows/NT Programming in C/C++, Java	…

Demo

- Is being developed in our lab



Structured data in search results

Recognizing Set-Oriented Queries

Association Queries

Behind the search results

- Knowledge graph

knowledge → a set of facts
a fact: a triple <subject, predicate, object>

  - Large-scale extraction of facts from Web text
  - Incorporate it into an IR-style search engine

Resource Description Framework (RDF) data

<Bob> <is a> <person>.

<Bob> <is a friend of> <Alice>.

<Bob> <is born on> <the 4th of July 1990>.

<Bob> <is interested in> <the Mona Lisa>.

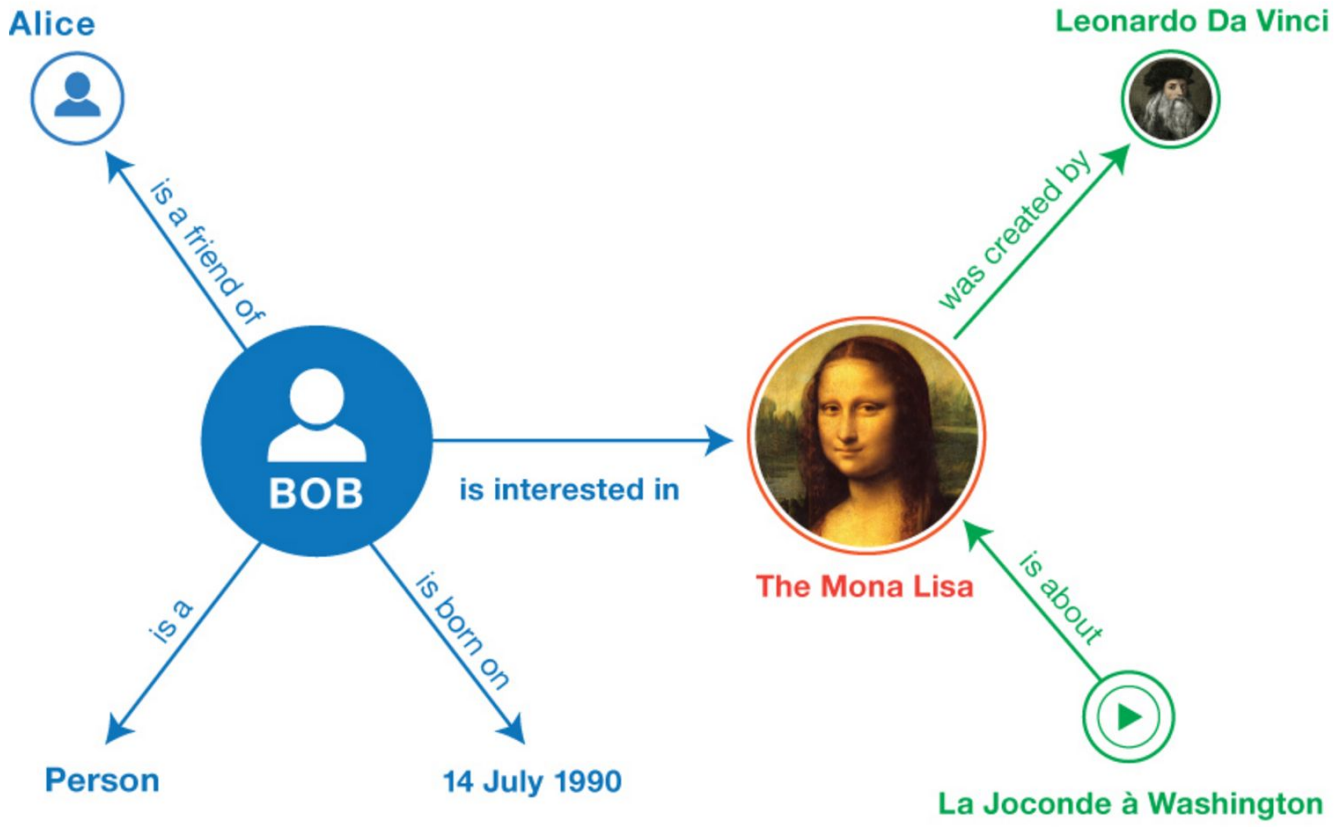
<the Mona Lisa> <was created by> <Leonardo da Vinci>.

<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>

Resource Description Framework (RDF) data (cont.)

- represent a knowledge
- A knowledge is a set of facts, each of which is represented by a triple (subject, predicate, object)

RDF data is a kind of graph



Freebase is a database of entities

One entity per thing in the world

- Stable, long-lived identifiers
- Inclusive policy

/en/sienna_miller

Practical data

- Focus on available data
- People, places, products, etc.

/en/sony_dsc_s750

Data to build apps

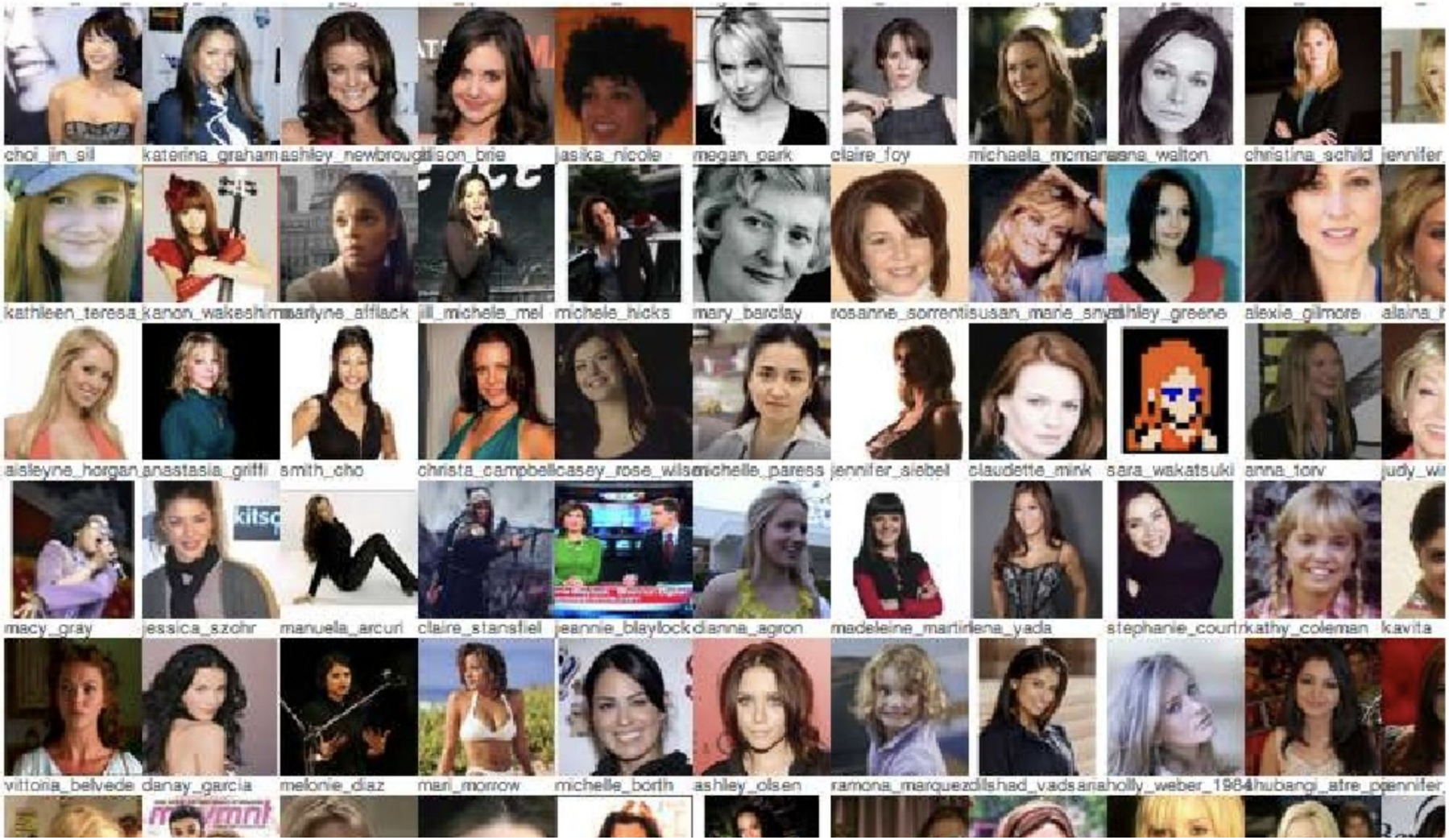
- Names, images, descriptions
- Dates, measurements and relationships

Frost/Nixon (2008)

Director:	Ron Howard
Released:	Oct 15, 2008
Runtime:	2 hr. 2 min.
Genres:	Drama
Box office:	$18,593,156

/en/frost_nixon_2008

Actresses (37,079)

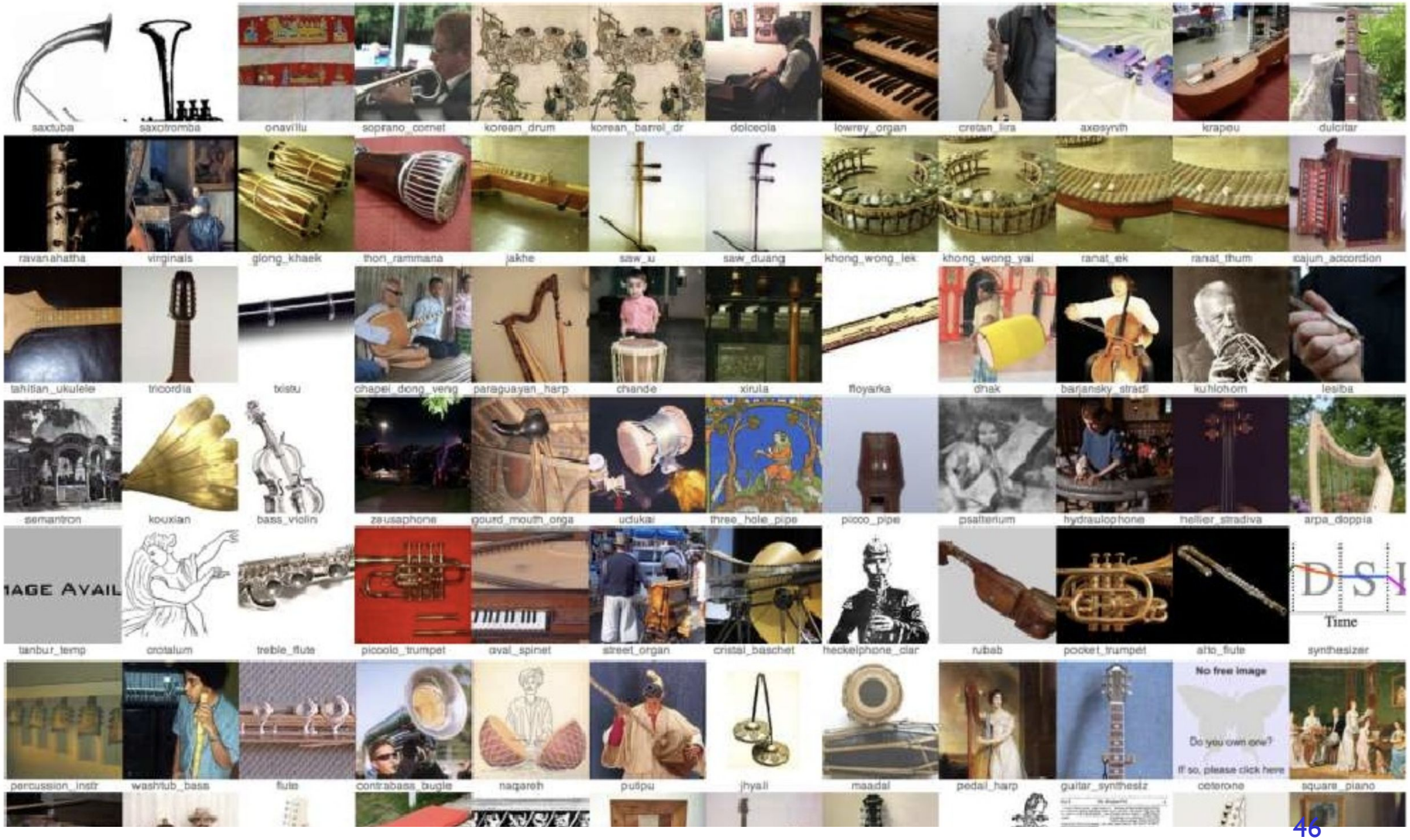


Football Players (16,568)

Cheeses (488)


pleasant_ridge_r
gurt
brie_de_meaux
barbara_gilmour
aveyronnais
affidelice_au_ch
pelardon
edel_de_cleron
applewood_cheese
montasio
whey_cheese
rochebaron
fromager_d'affino
etorki
tourme_de_montbr
picodon
granular_cheese
roncal_cheese
chanco_cheese
bergamot
rokpol
golika
couronne_lochois
pouligny_saint_p
selles_sur_cher
maroilles
strained_yoghurt
pot_cheese
fine_fettle_york
morbier_charentai
la_serena_cheese
pasta_filata
tete_de_moine
cathare
bianca_bianca
big_woods_blue
beenleigh_blue
beaufort_alpage
bayley_hazen_blu
azeitao
abbaye_de_tamie
banon_de_provenc
pierce_pt
st_pat
mt_tam
marbon_cheese
bleu_du_vercors
bleu_de_gex
red_hawk
brevoline
saint_agur_blue
chabichou
babybel
stracciatella_di
green_cheese
scamorza_cheese
little_swiss
tillamook_cheddar
sainte-maure_de
beer_cheese
zamorano_cheese
itche_bai
boccoccini
serra_da_estrela
baguette_laonnai
blue_castello
bleu_des_causses
adelost
requeijao
cancoillotte
shropshire_blue_
pilgrims_choice

Musical Instruments (1,034)



Airports (11,556)

TV Programs (33,630)

Related entities are connected, forming a graph

Current stats:

- **~10M entities**
- **~1,800 "types"**
  - Celebrity
  - Movie
  - TV show
  - Book
  - Company
  - Location
  - Sports team
  - Product
  - Etc.
- **~275M facts**
- Continuous data input, cleanup, and syncing

DBPedia

- Extracting Structured Information from Wikipedia
- Wikipedia
  - 6.9 million articles
  - in 251 languages
  - monthly growth-rate: 4%

Structured information in Wikipedia

- Wikipedia articles contain infoboxes which use a template mechanism
- Images depicting the article's topic
- Categorization of the article
- Links to external webpages
- Intra-wiki links to other articles
- Inter-language links to articles about the same topic in different languages

province, in an area
"Calgary–Edmonton

pulation of 1,392,609,

alth and wellness,
ions.[5]

old" and "garden",
(pasture)"; or Gaelic

nhabited by the
mpson spent the
was the first

NWMP detachment
ois, after NWMP

an important
Montreal in 1996.[10]
ed as "The City of
municipal, local duties

200. Although no one
, to pre

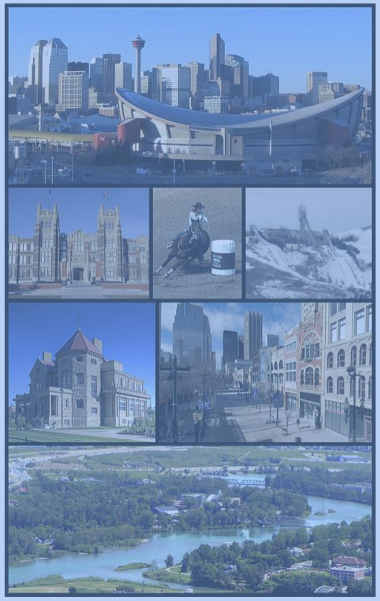
As a re
ntre of C

Infobox

Calgary

City

City of Calgary



From top left: Scotiabank Saddledome and Downtown Calgary, Southern Alberta Institute of Technology, Calgary Stampede, Canada Olympic Park, Lougheed House, Stephen Avenue, Calgary Zoo

Flag

Coat of arms

Nickname(s): Cowtown, Stampede City, more... [1][2]

Motto: *Onward*

Template | Edit

Generated from: Infobox settlement

Cancel | Transclusion | Apply changes

- image_skyline
- imagesize
- image_caption
- image_flag
- image_seal
- image_shield
- nickname
- motto
- image_map

Infobox settlement

an Infobox for human settlements (cities, towns, villages, communities) as well as other administrative districts, counties, provinces, etcetera

There might be some additional information about the "Infobox settlement" template on its page.

name: Calgary

official_name: City of Calgary

settlement_type: City

image_skyline: Calgarymontage5.jpg <!--Montage determined through past

Hide options

Extracting Infobox Data

```
http://en.wikipedia.org/wiki/Calgary
```

```
<http://dbpedia.org/resource/Calgary>
  dbpedia:native_name "Calgary" ;
  dbpedia:altitude "1048" ;
  dbpedia:population_city  "988193" ;
  dbpedia:population_metro "1079310" ;
  mayor_name
        dbpedia:Dave_Bronconnier ;
  governing_body
        dbpedia:Calgary_City_Council ;
 ...
```

- **Altogether 9,100,000 RDF triples extracted from 754,000 infoboxes**

Calgary

Downtown Calgary.

Government	
- Mayor	Dave Bronconnier (Past mayors)
- Governing body	Calgary City Council
- Manager	Owen A. Tobert
Area [1]	
- City	726.50 km² (280.5 sq mi)
- Metro	5,107.43 km² (1,972 sq mi)
Elevation	1,048 m (3,438.3 ft)
Population (2006)[1]	
- City	988,193
- Density	1,360.2/km² (3,522.9/sq mi)
- Metro	1,079,310
- Population rank	3rd
- Metro rank	5th

Christian Bizer et al: DBpedia – Querying Wikipedia Like a Database (May 11, 2007)

Extracting Other Article Data

- Short and long abstracts in 10 different languages

```
dbpedia:Calgary dbpedia:abstract "Calgary is the largest ..."@en ;
                dbpedia:abstract "Calgary ist eine Stadt ..."@de .
```

- Categorization information

```
dbpedia:Calgary skos:subject dbpedia:Category_Cities_in_Alberta ;
                skos:subject dbpedia:Host_cities_Olympic_Games .
```

- Links to the original Wikipedia articles, pictures and relevant external web pages

```
dbpedia:Calgary
      foaf:page <http://en.wikipedia.org/wiki/Calgary> ;
      dbpedia:wikipage-de <http://de.wikipedia.org/wiki/Calgary> ;
      foaf:depiction <http://upload.wikimedia.org/thumb/3/32> ;
      dbpedia:reference <http://www.calgary.ca> ;
      dbpedia:reference <http://www.tourismcalgary.com>.
```

Christian Bizer et al: DBpedia – Querying Wikipedia Like a Database (May 11, 2007)

The DBpedia Dataset

- 1,600,000 concepts
  - 58,000 persons
  - 70,000 places
  - 35,000 music albums
  - 12,000 films
- described by 91 million triples using 8,141 different properties.
- 557,000 links to pictures
- 1,300,000 links to relevant external web pages

Example queries

- People who were born in Berlin before 1900
- German musicians with German and English descriptions
- Musicians who were born in Berlin
- Soccer players, who are born in a country with more than 10 million inhabitants, who played as goalkeeper for a club that has a stadium with more than 30.000 seats and the club country is different from the birth country

Current research interests

- Parallel (/Distributed/Concurrent)
  - Effectively utilizing all hardware resources
- Graph
  - Flexible representation
- Intelligent
  - Incorporating the new wave of deep learning/AI for dark data

COMPUTATION


Thank you!