Introduction to Information Retrieval

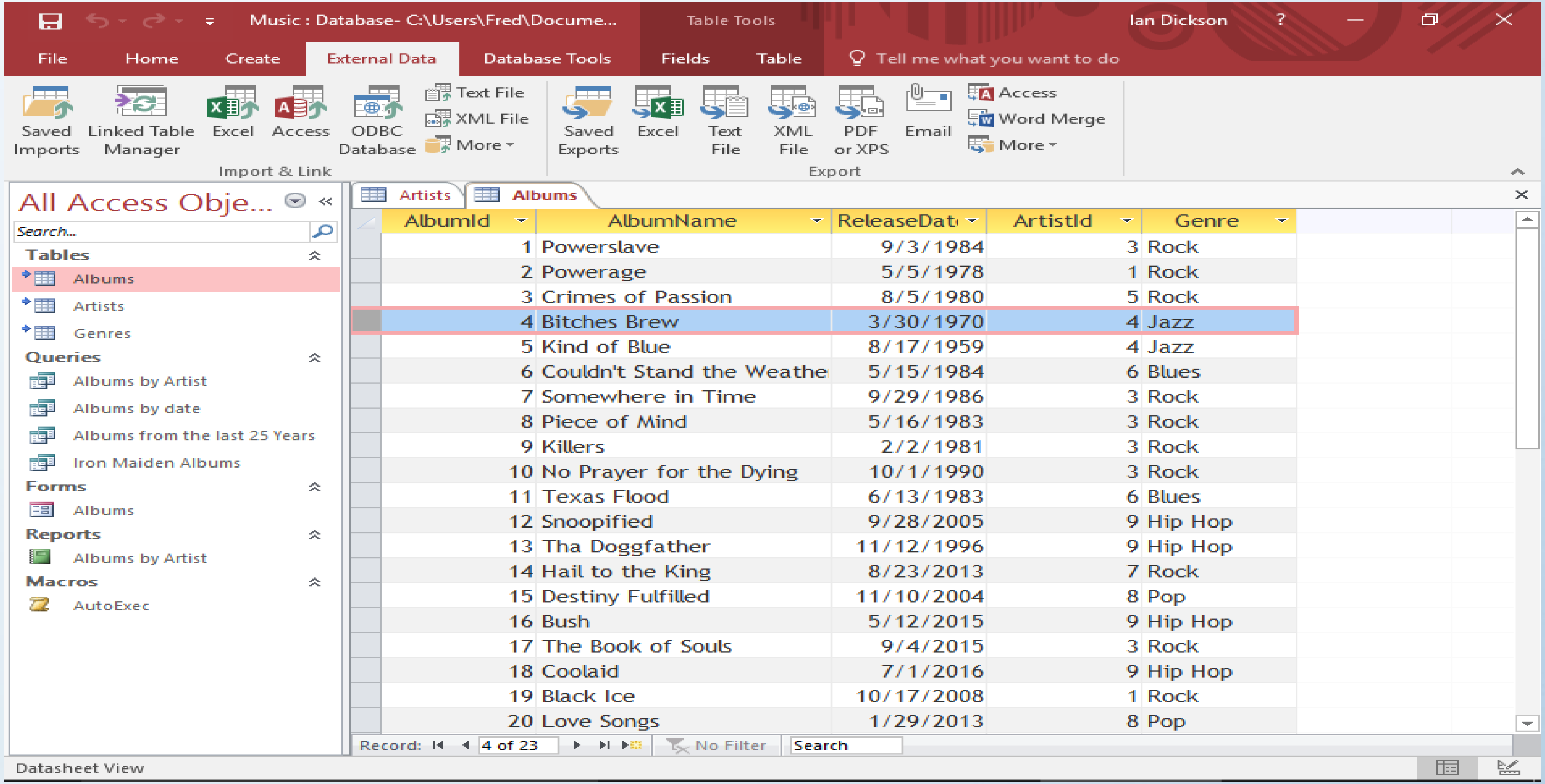
mireya.paredes@udlap.mx

Types of data

1. Structured
2. Unstructured
3. Semi-structured

Structured Data

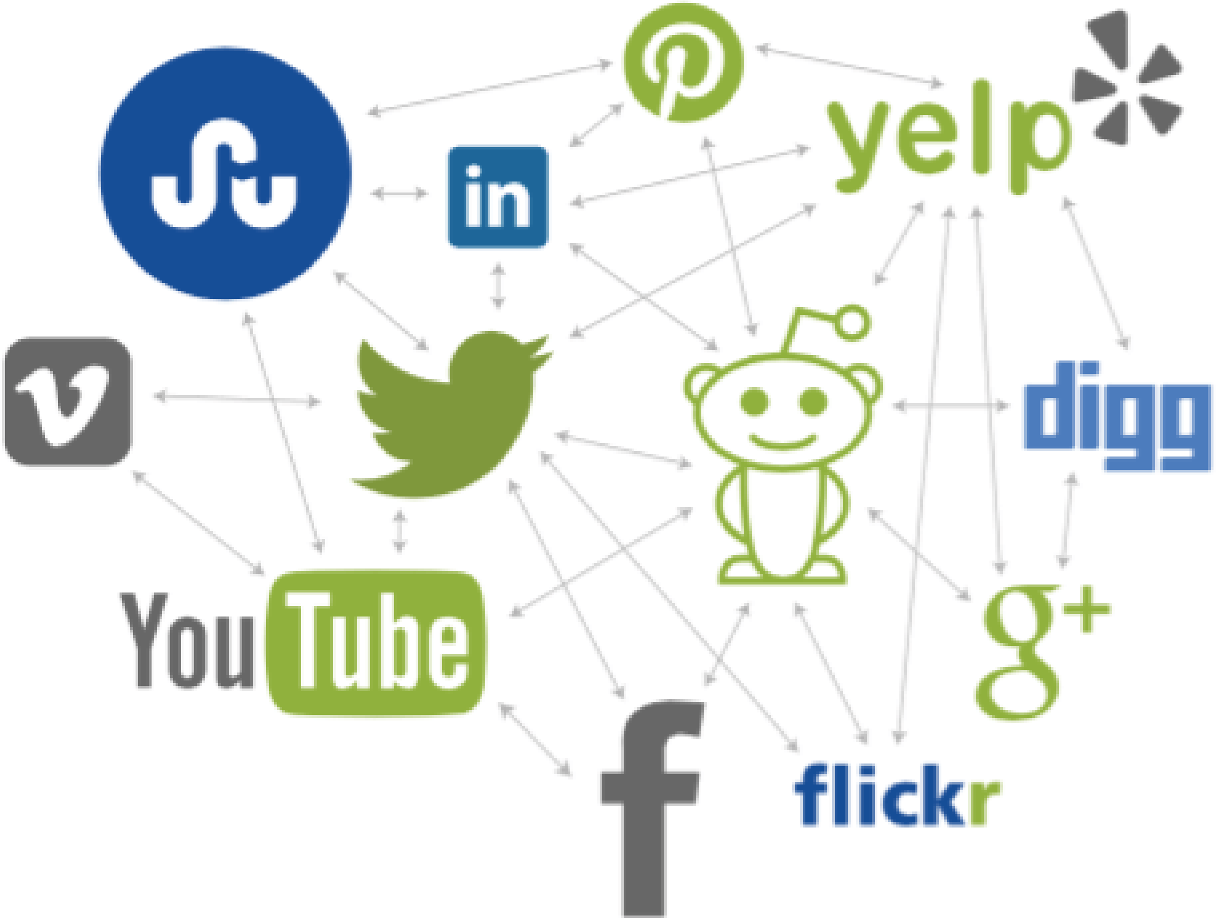
SQL



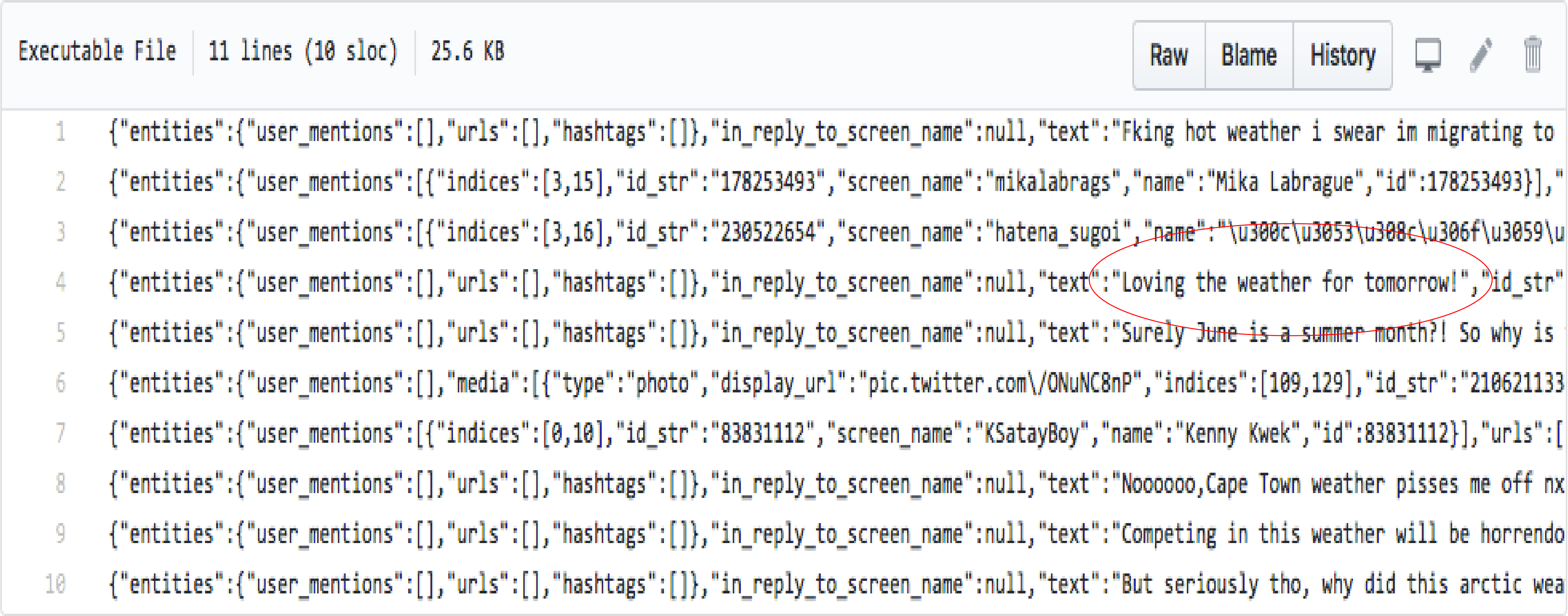
AlbumId	AlbumName	ReleaseDat	ArtistId	Genre
1	Powerslave	9/3/1984	3	Rock
2	Powerage	5/5/1978	1	Rock
3	Crimes of Passion	8/5/1980	5	Rock
4	Bitches Brew	3/30/1970	4	Jazz
5	Kind of Blue	8/17/1959	4	Jazz
6	Couldn't Stand the Weathe	5/15/1984	6	Blues
7	Somewhere in Time	9/29/1986	3	Rock
8	Piece of Mind	5/16/1983	3	Rock
9	Killers	2/2/1981	3	Rock
10	No Prayer for the Dying	10/1/1990	3	Rock
11	Texas Flood	6/13/1983	6	Blues
12	Snoopified	9/28/2005	9	Hip Hop
13	Tha Doggfather	11/12/1996	9	Hip Hop
14	Hail to the King	8/23/2013	7	Rock
15	Destiny Fulfilled	11/10/2004	8	Pop
16	Bush	5/12/2015	9	Hip Hop
17	The Book of Souls	9/4/2015	3	Rock
18	Coolaid	7/1/2016	9	Hip Hop
19	Black Ice	10/17/2008	1	Rock
20	Love Songs	1/29/2013	8	Pop

UNSTRUCTURED DATA

Social Media

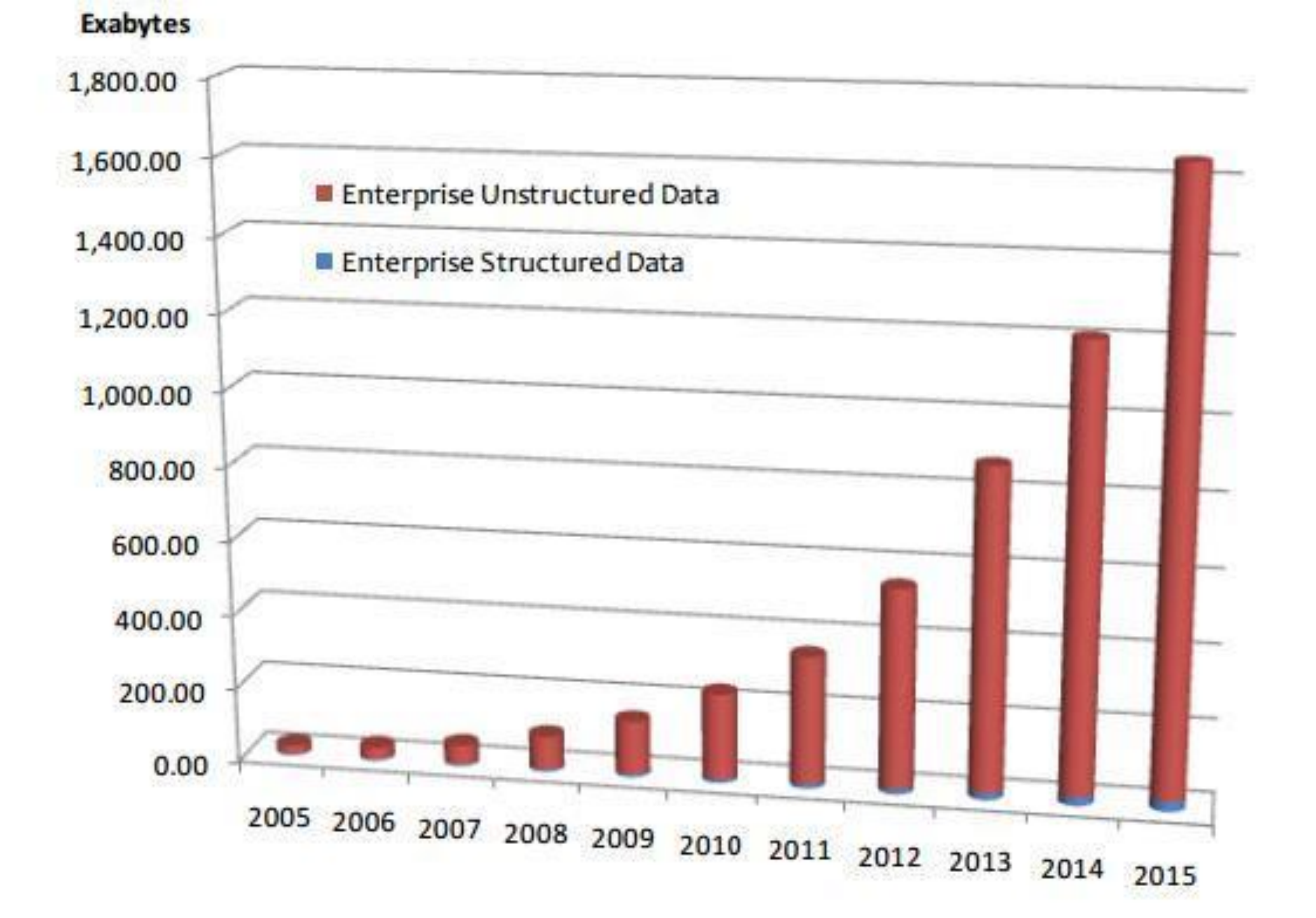


Twitter Data Example



```
{"entities":{"user_mentions":[],"urls":[],"hashtags":[]},"in_reply_to_screen_name":null,"text":"Fking hot weather i swear im migrating to
{"entities":{"user_mentions":[{"indices":[3,15],"id_str":"178253493","screen_name":"mikalabrags","name":"Mika Labrague","id":178253493}],"
{"entities":{"user_mentions":[{"indices":[3,16],"id_str":"230522654","screen_name":"hatena_sugoi","name":"\u300c\u3053\u308c\u306f\u3059\u
{"entities":{"user_mentions":[],"urls":[],"hashtags":[]},"in_reply_to_screen_name":null,"text":"Loving the weather for tomorrow!","id_str"
{"entities":{"user_mentions":[],"urls":[],"hashtags":[]},"in_reply_to_screen_name":null,"text":"Surely June is a summer month?! So why is
{"entities":{"user_mentions":[],"media":[{"type":"photo","display_url":"pic.twitter.com\/ONuNC8nP","indices":[109,129],"id_str":"210621133
{"entities":{"user_mentions":[{"indices":[0,10],"id_str":"83831112","screen_name":"KSatayBoy","name":"Kenny Kwek","id":83831112}],"urls":[
{"entities":{"user_mentions":[],"urls":[],"hashtags":[]},"in_reply_to_screen_name":null,"text":"Noooooo,Cape Town weather pisses me off nx
{"entities":{"user_mentions":[],"urls":[],"hashtags":[]},"in_reply_to_screen_name":null,"text":"Competing in this weather will be horrendo
{"entities":{"user_mentions":[],"urls":[],"hashtags":[]},"in_reply_to_screen_name":null,"text":"But seriously tho, why did this arctic wea
```

How much unstructured data?



Taken from Data Science Central (IDC)

Semi-structured data

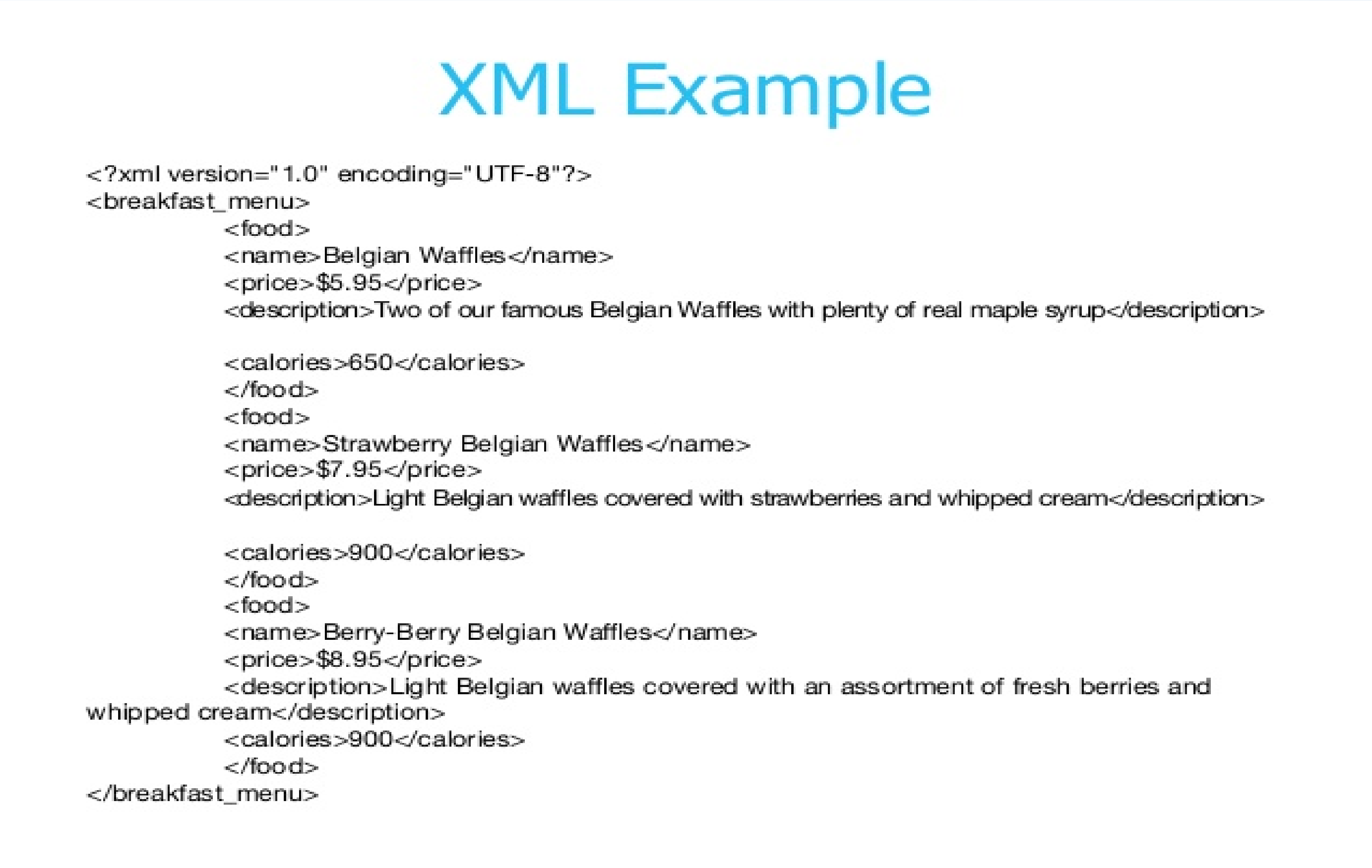
1. No fixed schema
2. Structured is irregular
3. Examples

   Web Pages

   Information integration

   XML

Semi-structured data example



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
                <food>
                <name>Belgian Waffles</name>
                <price>$5.95</price>
                <description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>

                <calories>650</calories>
                </food>
                <food>
                <name>Strawberry Belgian Waffles</name>
                <price>$7.95</price>
                <description>Light Belgian waffles covered with strawberries and whipped cream</description>

                <calories>900</calories>
                </food>
                <food>
                <name>Berry-Berry Belgian Waffles</name>
                <price>$8.95</price>
                <description>Light Belgian waffles covered with an assortment of fresh berries and
whipped cream</description>
                <calories>900</calories>
                </food>
</breakfast_menu>
```

What we need!

1. To process large document collections quickly.

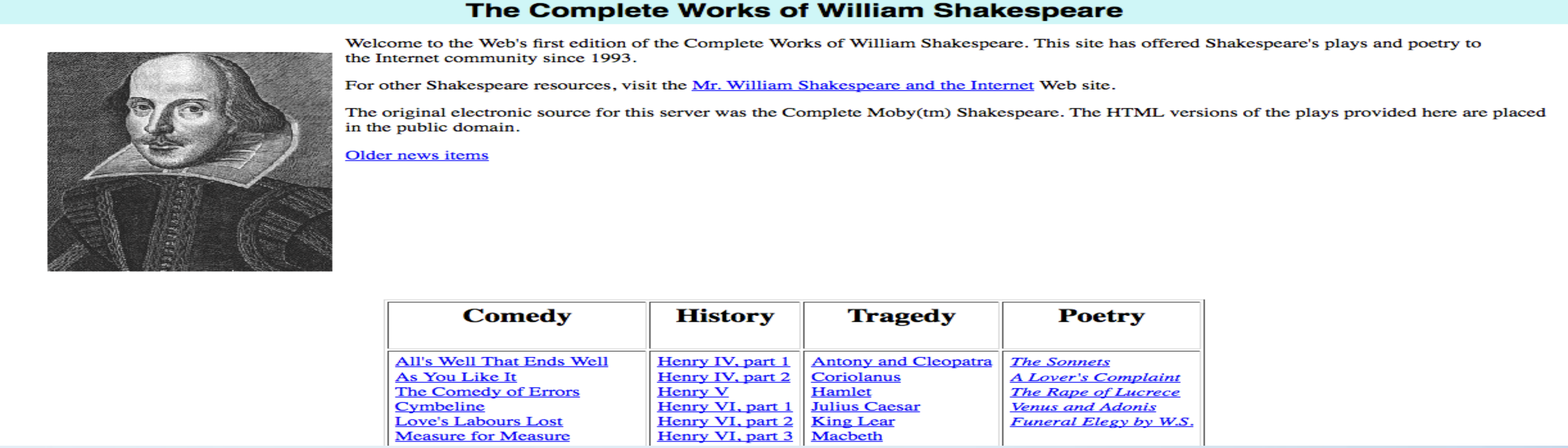
Billions/Trillions of words

2. To allow more flexible matching operations.

“Romans **NEAR** countrymen”

1. To allow **ranked** retrieval.

An example IR problem



The Complete Works of William Shakespeare

Welcome to the Web's first edition of the Complete Works of William Shakespeare. This site has offered Shakespeare's plays and poetry to the Internet community since 1993.

For other Shakespeare resources, visit the Mr. William Shakespeare and the Internet Web site.

The original electronic source for this server was the Complete Moby(tm) Shakespeare. The HTML versions of the plays provided here are placed in the public domain.

Older news items

Comedy	History	Tragedy	Poetry
All's Well That Ends Well As You Like It The Comedy of Errors Cymbeline Love's Labours Lost Measure for Measure	Henry IV, part 1 Henry IV, part 2 Henry V Henry VI, part 1 Henry VI, part 2 Henry VI, part 3	Antony and Cleopatra Coriolanus Hamlet Julius Caesar King Lear Macbeth	*The Sonnets* *A Lover's Complaint* *The Rape of Lucrece* *Venus and Adonis* *Funeral Elegy by W.S.*

Roughly uses 32,000 words.

"Brutus **AND** Caesar **AND NOT** Calpurnia"

Boolean Retrieval Model

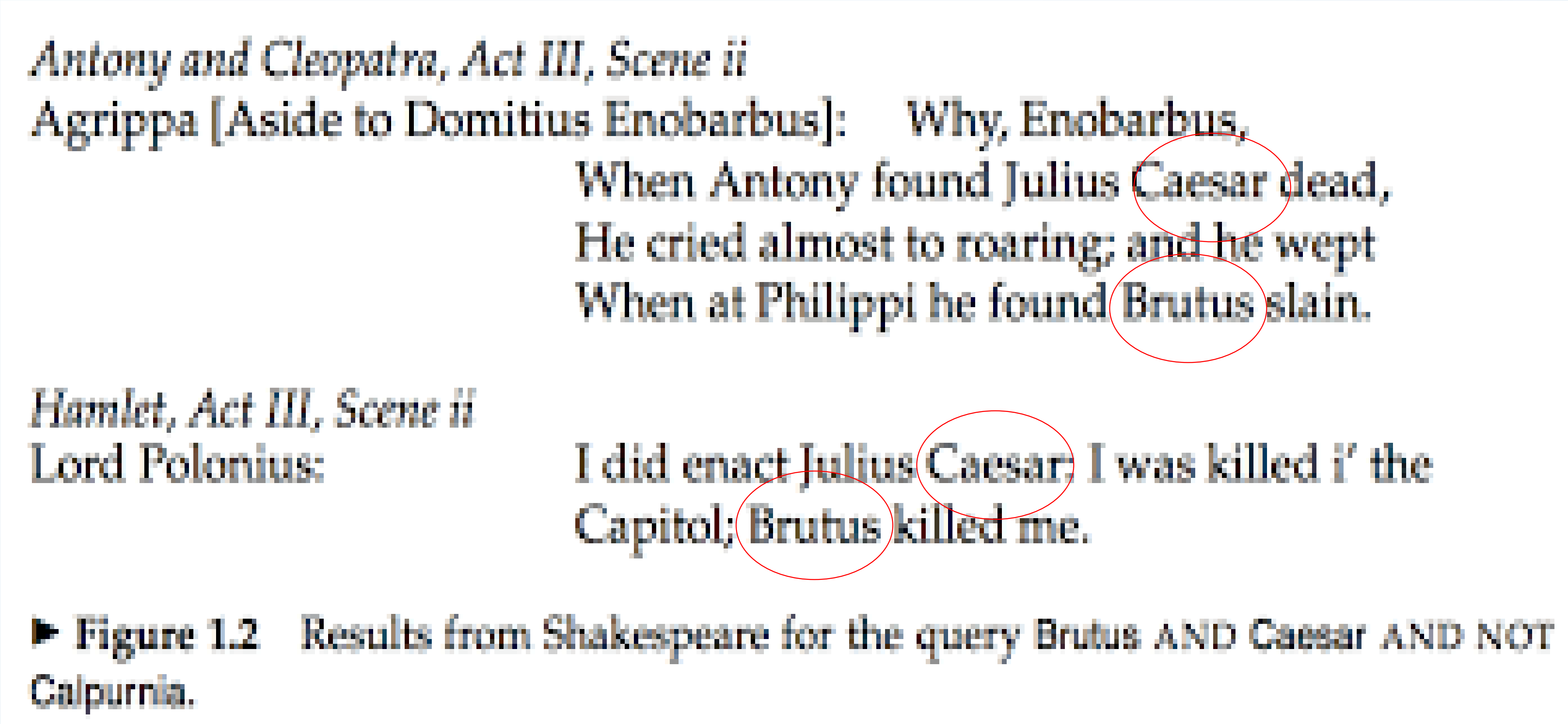
Terms	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Incidence matrix

Brutus **AND** Caesar **AND NOT** Calpurnia

110100 **AND** 110111 **AND** 100100 → *Antony* and *Cleopatra* and *Hamlet*

Information Retrieval



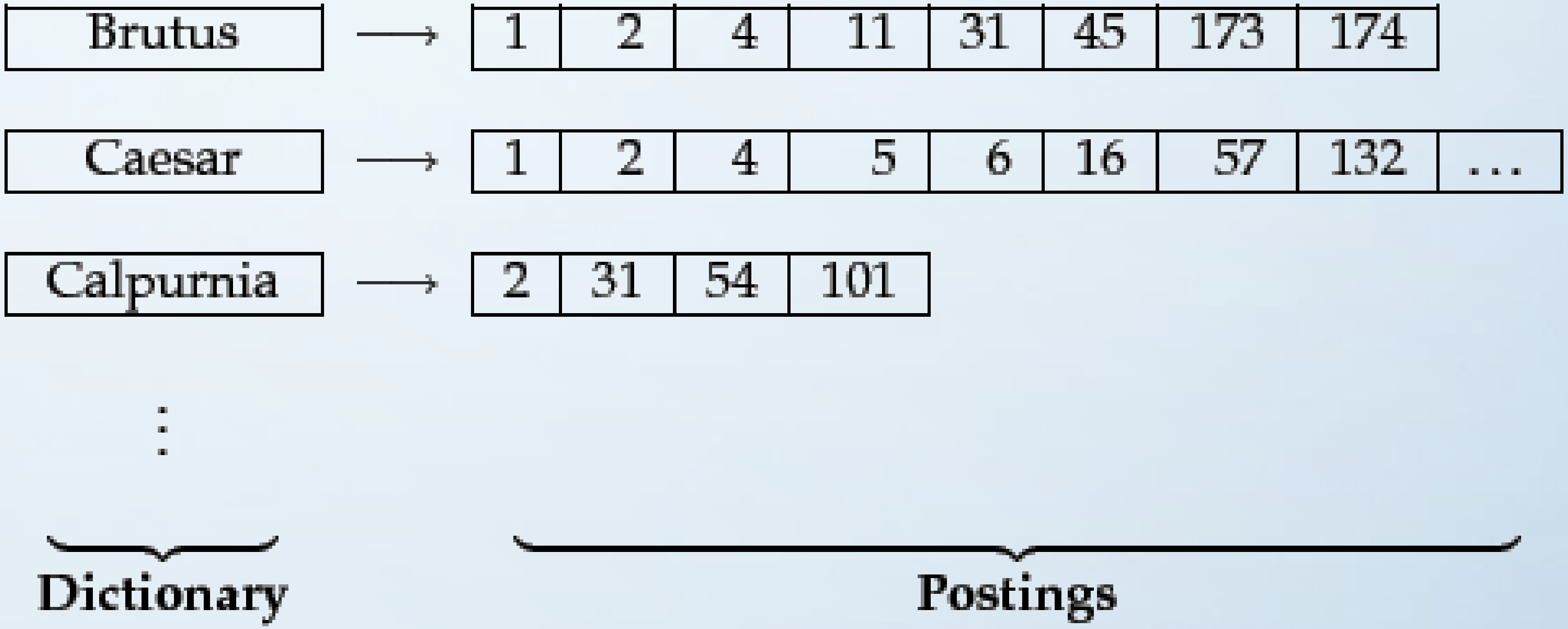
Antony and Cleopatra, Act III, Scene ii
Agrippa [Aside to Domitius Enobarbus]: Why, Enobarbus,
When Antony found Julius Caesar dead,
He cried almost to roaring; and he wept
When at Philippi he found Brutus slain.

Hamlet, Act III, Scene ii
Lord Polonius: I did enact Julius Caesar: I was killed i' the
Capitol; Brutus killed me.

▶ **Figure 1.2** Results from Shakespeare for the query Brutus AND Caesar AND NOT Calpurnia.

Why the **incidence matrix** is not convenient?

Inverted Index



Brutus → 1 | 2 | 4 | 11 | 31 | 45 | 173 | 174

Caesar → 1 | 2 | 4 | 5 | 6 | 16 | 57 | 132 | …

Calpurnia → 2 | 31 | 54 | 101

⋮

Dictionary

Postings

► **Figure 1.2** The two parts of an inverted index. The dictionary is commonly kept in memory, with pointers to each postings list, which is stored on disk.

Inverted Index Steps

1.- Collect the documents to be indexed.

2.- **Tokenize** the tex, turning each document into a list of tokens.

3.- Do linguistic preprocessing!

4.- Index the documents that each term occurs in by creating a **dictionary** and a **posting list.**

Homework2: 29 Agosto

- To implement the *Inverted index algorithm* to process a boolean query.
- Using the intersection function of the Algorithm in page 11 of the book Information to IR (Manning).
- It will be tested with my own documents.
- To write a one page(max) document, describing the problem and more important "your thoughts" and conclusions.
- Delivery time: 29/08/2018 in your folder of the course