Information Retrieval

CS5154/6054

Yizong Cheng

8/23/2022

COLLEGE OF
ENGINEERING & APPLIED SCIENCE

UNIVERSITY OF Cincinnati

CS5154/6054 Information Retrieval
Course Syllabus

Fall Semester 2022

Instructor Information:

Instructor:	Yizong Cheng
Office Location:	812A Rhodes
Email Address:	chengy@ucmail.uc.edu
Phone Number:	513-556-1809
Office Hours:	3:00-4:00 MW

General Course Information:

Course Number:	CS5154/6054
Course Title:	Information Retrieval
Credit Hours:	3
Contact Hours:	3
Meeting Days/Times:	9:30-10:50 TH, Swift 500
Prerequisites:	Python programming
Course Description:	Information storage and retrieval with unstructured data. Inverted index, tf-idf and cosine similarity, and relevance-based evaluation. Probabilistic information retrieval with text classification, feature selection, and clustering.
Course Delivery Mode:	This course is an in-person course. There may be frequent in-class quizzes.
Course Location and/or Access:	This course can be found on Canvas (accessible via canopy.uc.edu). Lecture notes, reference links, and homework assignments will be posted on the course site in Canvas.

Course Resources:

Required Textbooks and Materials:	Introduction to Information Retrieval by Manning, Raghavan, and Schutze, Cambridge 2008 (IIR)
Supplemental Textbooks and Materials:	Blueprints for Text Analytics Using Python by Albrecht, Ramachandran, and Winkler, O'Reilly, 2020 (Blueprints)
Required Software or Hardware:	Python

Course Assignments:

Quizzes:	in-class
Assignments:	mostly Python programming
Exams:	in-class, midterm and final

Grading:

Final numerical grades for the course are determined based on the following weights for assignments.

Assignment:	**Weighting:**
Quizzes and assignments:	50%
Exams:	50%
Total:	100%

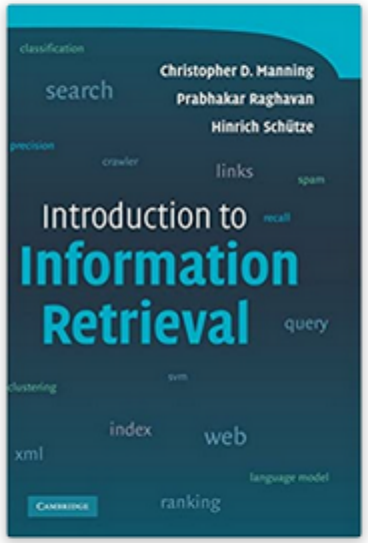
Final letter grades for the course will be determined using your final numerical grade. The range will be different for undergraduate (CS5154) and graduate (CS6054) sections.

File Edit View History Bookmarks Tools Help

Introduction to Information Ret

https://www.amazon.com/Introduction-Information-Retrieval-Christopher-Manning/

Look inside



See all 3 images

Introduction to Information Retrieval Illustrated Edition

by Christopher D. Manning (Author), & 2 more

166 ratings

See all formats and editions

eTextbook	Hardcover
$22.22 - $38.00	**$19.99 - $67.55**

Read with Our Free App

18 Used from $40.01
14 New from $63.50
2 Rentals from $19.99

Class-tested and coherent, this groundbreaking new textbook teaches web-era information retrieval, including web search and the related areas of text classification and text clustering from basic concepts. Written from a computer science perspective by three leading experts in the field, it gives an up-to-date treatment of all aspects of the design and

Read more

Rent
$19.99

List Price: ~~$70.99~~
Save: $51.00 (72%)

Due Date: Dec 14, 2022 Rental Details

- **FREE** return shipping at the end of the semester.
- Access codes and supplements are not guaranteed with rentals.

In Stock.

Rented from Apex_media
Fulfilled by Amazon

FREE delivery **Wednesday, August 24**

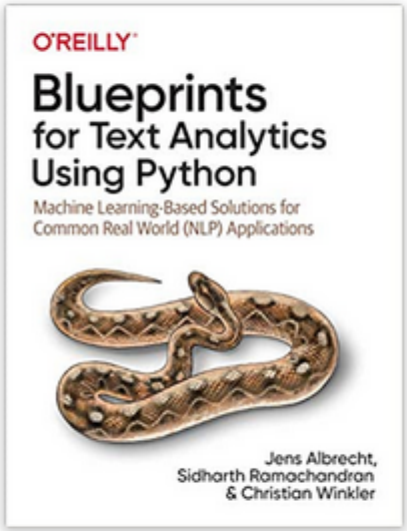
Or fastest delivery **Saturday, August 20**. Order within 9 hrs

File Edit View History Bookmarks Tools Help

Blueprints for Text Analytics Usi

https://www.amazon.com/Blueprints-Text-Analytics-Using-Python/dp/149207408X/

Look inside



See all 2 images

Blueprints for Text Analytics Using Python: Machine Learning-Based Solutions for Common Real World (NLP) Applications 1st Edition

by Jens Albrecht (Author), & 2 more

★★★★★ 24 ratings

See all formats and editions

Kindle	Paperback
$44.99	**$51.99**
Read with Our **Free App**	6 Used from $32.00 22 New from $39.38

Turning text into valuable information is essential for businesses looking to gain a competitive advantage. With recent improvements in natural language processing (NLP)

Buy new: $51.99

List Price: ~~$69.99~~

Save: $18.00 (26%)

& FREE Returns

FREE delivery **Wednesday, August 24**

Or fastest delivery **Saturday, August 20**. Order within 7 hrs 52 mins

Select delivery location

Only 13 left in stock (more on the way).

As an alternative, the Kindle eBook is available now and can be read on any device with the free Kindle app.

Qty: 1

Course Schedule:

Week	Class Period	Topic	Reading
1	Aug 23	term-document matrix	1 IIR
	Aug 25	tokenization with regular expressions	2 IIR and 4 Blueprints
2	Aug 30	set similarity and ranking	3 IIR
	Sep 1	spelling correction	3 IIR
3	Sep 6	tf-idf and cosine similarity	6 IIR
	Sep 8	sklearn vectorizer and ngram	5 Blueprints
4	Sep 13	precision and recall	8 IIR
	Sep 15	evaluating IR	8 IIR

What to Learn Today

- Information retrieval
  - Documents and terms
  - retrieve all documents containing a term
  - Boolean query : AND, OR, NOT between terms
- The ad hoc retrieval: fixed documents and changing query
  - Filtering or classification: fixed query and changing documents
- Documents as sets of terms
  - document-term matrix
  - Inverted index
  - Python implementation with list, set, and dictionary (dict)

Information Retrieval

- Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).
- The term information retrieval was coined by Calvin Mooers (1919-1994) in 1948/1950.

ICM 1950

University of Cincinnati,
Cincinnati, Ohio, U. S. A.

INFORMATION RETRIEVAL VIEWED AS TEMPORAL SIGNALLING

Calvin N. Mooers

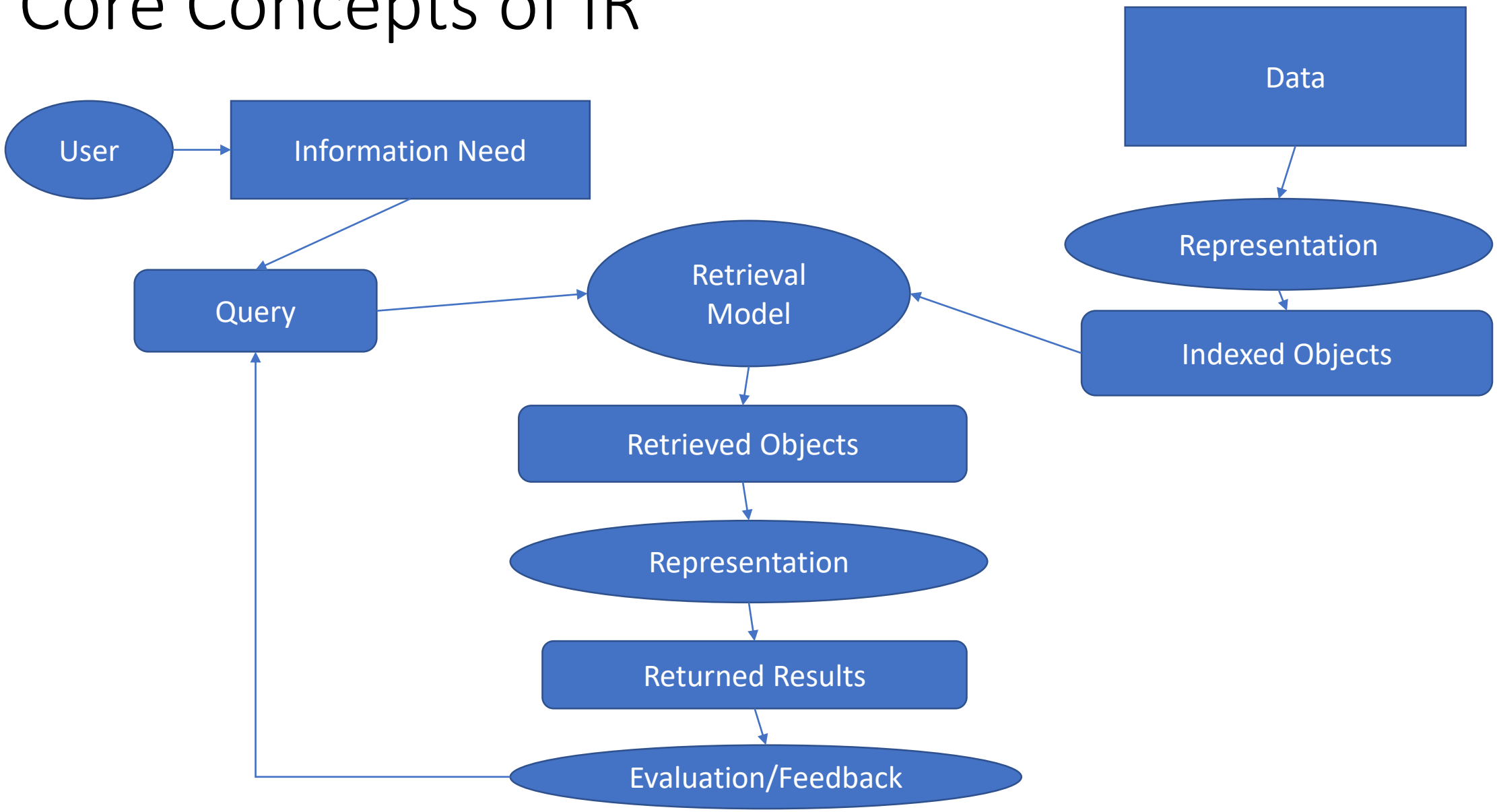
The problem of directing a user to stored information, some of which may be unknown to him, is the problem of "information retrieval". Signalling theories can be applied, though this is a form of temporal signalling, which distinguishes it from the point-to-point signalling currently under study by others.

In information retrieval, the addressee or receiver rather than the sender is the active party. Other differences are that communication is temporal from one epoch to a later epoch in time, though possibly at the same point in space; communication is in all cases unidirectional; the sender cannot know the particular message that will be of later use to the receiver and must send all possible

The *ad hoc* retrieval problem

- Given a user information need and a collection of documents, the IR system determines how well the documents satisfy the query and returns a subset of relevant documents to the user.
  - Collection is relatively stable.
  - Queries are created and used dynamically; change fast
  - “Ad-hoc”: formed or used for specific or immediate problems or needs – Merriam-Webster’s collegiate Dictionary
- Filtering: Queries are stable, while the collection changes.

Core Concepts of IR



Quoting ACM Survey 2019 Article 15

- In their broader sense, information retrieval systems (IRSs) aim to fulfill users' information needs expressed in a keyword-based query.
- More precisely, information retrieval (IR) refers to the process of selection, from a document repository, of the documents *likely* to be relevant to a particular information need, formulated by a query.
- Based on this definition, an IRS has to deal with a collection of documents, with users' information needs, and with the notion of relevance.

ACM Computing Surveys, Vol. 52, No. 1, Article 15. Publication date: February 2019

Documents and the Query

- In IR, documents are carriers of information;
  - in their original forms, they are human-understandable objects (e.g., Web pages, articles, books, and images),
  - which an IRS must transform into machine-understandable objects.
- This process is called indexing,
  - and its outcome is the association of a set of features (terms in textual documents) with documents.
  - These features constitute the basic elements employed to formally represent a document.
- A user's information needs are motivated by a user's information gap;
  - a query is a representation of these needs.
  - Once formal representations have been provided for both documents and queries, the system compares them to assess the relevance of each document to the considered query.

The IR Model

- Relevance is a complex notion composed of several dimensions, such as topicality, popularity, and novelty.
- An IRS can only estimate relevance, and generally topicality is the core relevance dimension.
- The assessment of topical relevance relies on the definition of a model, the IR model, which provides a formal means to represent and compare both documents and queries.
- Different mathematical theories have been employed to define IR models, which include set theory, linear algebra, probability theory, and formal logics.

ACM Computing Surveys, Vol. 52, No. 1, Article 15. Publication date: February 2019

Boolean Retrieval

- The Boolean model is arguably the simplest model to base an information retrieval system on.
- Queries are Boolean expressions, e.g., Caesar and Brutus
- The search engine returns all documents that satisfy the Boolean expression.
- Windows OS has files indexed, too.
- Document representation for Boolean retrieval: sets of words
  - Each document is a set of words. (words in general are substrings of the text)
  - The collection is a collection of sets.
  - A collection of sets is called a hypergraph in mathematics. Each word is a vertex and each document (a set of words) is a hyperedge.

Brutus AND Caesar AND NOT Calpurnia

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.

File Edit View History Bookmarks Tools Help

The Python Tutorial — Python 3

https://docs.python.org/3/tutorial/

3.10.6 Go

The Python Tutorial

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see The Python Standard Library. The Python Language Reference gives a more formal definition of the language. To write extensions in C or C++, read Extending and Embedding the Python Interpreter and Python/C API Reference Manual. There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in The Python Standard Library.

File Edit View History Bookmarks Tools Help

5. Data Structures — Python 3.1

https://docs.python.org/3/tutorial/datastructures.html

3.10.6 Go

5. Data Structures

This chapter describes some things you've learned about already in more detail, and adds some new things as well.

5.1. More on Lists

The list data type has some more methods. Here are all of the methods of list objects:

`list.`**`append`**`(x)`

Add an item to the end of the list. Equivalent to `a[len(a):] = [x]`.

`list.`**`extend`**`(iterable)`

Extend the list by appending all the items from the iterable. Equivalent to `a[len(a):] = iterable`.

`list.`**`insert`**`(i, x)`

Insert an item at a given position. The first argument is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list, and `a.insert(len(a), x)` is equivalent to `a.append(x)`.

`list.`**`remove`**`(x)`

Remove the first item from the list whose value is equal to *x*. It raises a `ValueError` if there is no such item.

`list.`**`pop`**`([i])`

Preferred Way to Run Python Programs

Command Prompt

```
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

Y:\>c:

C:\>cd classes

C:\classes>cd 6054

C:\classes\6054>python IR1A.py
21647 Thus saith the LORD; For three transgressions of Damascus, and for four, I will not turn away the punishment there
of; because they have threshed Gilead with threshing instruments of iron:

21650 Thus saith the LORD; For three transgressions of Gaza, and for four, I will not turn away the punishment thereof;
because they carried away captive the whole captivity, to deliver them up to Edom:

21653 Thus saith the LORD; For three transgressions of Tyrus, and for four, I will not turn away the punishment thereof;
 because they delivered up the whole captivity to Edom, and remembered not the brotherly covenant:

21655 Thus saith the LORD; For three transgressions of Edom, and for four, I will not turn away the punishment thereof;
because he did pursue his brother with the sword, and did cast off all pity, and his anger did tear perpetually, and he
kept his wrath for ever:

21657 Thus saith the LORD; For three transgressions of the children of Ammon, and for four, I will not turn away the pun
ishment thereof; because they have ripped up the women with child of Gilead, that they might enlarge their border:

21660 Thus saith the LORD; For three transgressions of Moab, and for four, I will not turn away the punishment thereof;
because he burned the bones of the king of Edom into lime:

21663 Thus saith the LORD; For three transgressions of Judah, and for four, I will not turn away the punishment thereof;
```

Editing Python Program Using Notepad

*IR1A - Notepad

File Edit Format View Help

```
# IR1A.py CS5154/6054 cheng 2022
# read lines from a text file
# retrieve documents for boolean queries
# Usage: python IR1A.py

f = open("bible.txt", "r")
docs = f.readlines()
f.close()
word1 = 'punishment'
word2 = 'transgressions'
for i in range(len(docs)):
    if word1 in docs[i] and word2 in docs[i]:
        print(i, docs[i])
```

Using Command Line Arguments

IR1A2 - Notepad

File Edit Format View Help

```
# IR1A2.py CS5154/6054 cheng 2022
# read lines from a text file
# retrieve documents for boolean queries
# Usage: python IR1A2.py bible.txt punishment transgressions

import sys
f = open(sys.argv[1], "r")
docs = f.readlines()
f.close()

word1 = sys.argv[2]
word2 = sys.argv[3]
for i in range(len(docs)):
    if word1 in docs[i] and word2 in docs[i]:
        print(i, docs[i])
```

Command Prompt

```
C:\classes\6054>python IR1A2.py bible.txt punishment transgressions
21647 Thus saith the LORD; For three transgressions of Damascus, and for four, I will not turn away the punishment there
of; because they have threshed Gilead with threshing instruments of iron:

21650 Thus saith the LORD; For three transgressions of Gaza, and for four, I will not turn away the punishment thereof;
because they carried away captive the whole captivity, to deliver them up to Edom:

21653 Thus saith the LORD; For three transgressions of Tyrus, and for four, I will not turn away the punishment thereof;
 because they delivered up the whole captivity to Edom, and remembered not the brotherly covenant:

21655 Thus saith the LORD; For three transgressions of Edom, and for four, I will not turn away the punishment thereof;
because he did pursue his brother with the sword, and did cast off all pity, and his anger did tear perpetually, and he
kept his wrath for ever:

21657 Thus saith the LORD; For three transgressions of the children of Ammon, and for four, I will not turn away the pun
ishment thereof; because they have ripped up the women with child of Gilead, that they might enlarge their border:

21660 Thus saith the LORD; For three transgressions of Moab, and for four, I will not turn away the punishment thereof;
because he burned the bones of the king of Edom into lime:

21663 Thus saith the LORD; For three transgressions of Judah, and for four, I will not turn away the punishment thereof;
 because they have despised the law of the LORD, and have not kept his commandments, and their lies caused them to err,
after the which their fathers have walked:

21665 Thus saith the LORD; For three transgressions of Israel, and for four, I will not turn away the punishment thereof
; because they sold the righteous for silver, and the poor for a pair of shoes;


C:\classes\6054>
```

Using User Input

IR1A3 - Notepad

File Edit Format View Help

```
# IR1A3.py CS5154/6054 cheng 2022
# read lines from a text file
# retrieve documents for boolean queries
# Usage: python IR1A3.py bible.txt

import sys
f = open(sys.argv[1], "r")
docs = f.readlines()
f.close()

word1 = input("Enter word 1: ")
word2 = input("Enter word 2: ")
for i in range(len(docs)):
    if word1 in docs[i] and word2 in docs[i]:
        print(i, docs[i])
```

Command Prompt

```
C:\classes\6054>python IR1A3.py bible.txt
Enter word 1: punishment
Enter word 2: transgressions
21647 Thus saith the LORD; For three transgressions of Damascus, and for four, I will not turn away the punishment there
of; because they have threshed Gilead with threshing instruments of iron:

21650 Thus saith the LORD; For three transgressions of Gaza, and for four, I will not turn away the punishment thereof;
because they carried away captive the whole captivity, to deliver them up to Edom:

21653 Thus saith the LORD; For three transgressions of Tyrus, and for four, I will not turn away the punishment thereof;
 because they delivered up the whole captivity to Edom, and remembered not the brotherly covenant:

21655 Thus saith the LORD; For three transgressions of Edom, and for four, I will not turn away the punishment thereof;
because he did pursue his brother with the sword, and did cast off all pity, and his anger did tear perpetually, and he
kept his wrath for ever:

21657 Thus saith the LORD; For three transgressions of the children of Ammon, and for four, I will not turn away the pun
ishment thereof; because they have ripped up the women with child of Gilead, that they might enlarge their border:

21660 Thus saith the LORD; For three transgressions of Moab, and for four, I will not turn away the punishment thereof;
because he burned the bones of the king of Edom into lime:

21663 Thus saith the LORD; For three transgressions of Judah, and for four, I will not turn away the punishment thereof;
 because they have despised the law of the LORD, and have not kept his commandments, and their lies caused them to err,
after the which their fathers have walked:

21665 Thus saith the LORD; For three transgressions of Israel, and for four, I will not turn away the punishment thereof
; because they sold the righteous for silver, and the poor for a pair of shoes;
```

A COMPLETE

CONCORDANCE

TO THE

BIBLE OF THE

LAST TRANSLATION.

By helpe whereof any passage of holy Scripture may bee readily turned unto.

The whole reviewed, corrected, and much enlarged by CLEMENT COTTON.

And againe reuieued and corrected by H.T.

The further use and benefit of this Worke is more fully declared in the Prefaces to the Reader.

DEUTERENOMIE 11:18.

Ye shall lay up these my words in your heart.

COLOSS. 3:16.

Let the word of Christ dwell in you richly.

E Libris Guilhelmi Ambrosij

Printed Anno Dom. 1635.

1[8].10.Pharis & the other a *Publicane*
11.or even as this *Publicane*
13.the *Publicane* standing afar

Publicanes.

*Mat.*5.46. doe not eu n the *Publicanes* So Verse 47.
9.10. many *Publicanes* and sinners So *Mark.*2 15.
11. with *Publicanes* and sinners? So *Mar.*2,16
11.19.a friend of *Publicanes* and
21.31 that the *Pulicanes* & harlots
32.the *Publicanes* and harlots
*Luk.*3 12.then came also *Publicanes*
7.29.the *Publicanes* iustified God
15.1. drew neere to him all the *Publ.*
19.2. the chiefe among the *Publica.*

Publike/ly.

*Mar.*1.10 here *publike* example
*Act.*8.28. and that *publikely*, shewing
20.20.& hauе taught you *publikely*

Publish.

*Deut.*32.3.*publish* the name of the Lord
1 *Sa.*31,9.to *publish* it in the house of
2 *Sam.*1.20. *publish* it not in the streets
*Neh.*8.15. that they should *publish*, and
*Psal.*26.7.that I may *publish* with the
*Ier.*4.5.*publish* in Ierusalem, and say
16. *publish* against Ierusalem, that
5.20.and *publish*, it in Iudah, saying
31.7. *publish* ye. Praise ye, and say
46.14 *publish* in Migdol.
publish in Noph
50.2. *publish* and set vp a standard,
publish & conceale not
*Amos.*3.9.*publish* in the places of Ashd.
4.5.& *publish* the free offrings, for
*Mar.*1.45.began to *publish* it much

Published.

*Est.*1.22. and that it should be *published*
3.14. was *published* to all people that
*Psal.*68.11. of those that *published* it
Ionah 3.7.*published* through Ninneuh
*Mar.*7.36. great deale they *published* it
13.10. must first be *published* among
*Luk.*8.39. he went his way & *published*
*Act.*13 49. word of the Lord was *published*.

Publisheth.

Isay 52 7.that *publisheth* peace, that
of God that *publisheth* salvation
*Ier.*4.15. and *publisheth* affliction from

Publius.

*Act.*28.7.man whose name was *Publ.*
8 the father of *Publius* lay sicke

Puffed vp.

1.*Cor.*4.6.none of you be *puffed vp*
18.now some are *puffed vp* as
19.of them which are *puffed vp*
5.2.and ye are *puffed vp*, and
13.4.it selfe, is not *puffed vp*
*Col.*2.18.vainly *puffed vp* with his

Puffeth, puffe:

*Psal.*10.5.enemies, he *puffeth* at them
12.5.from him that *puffeth* at him
1.*Cor.*8.7.knowledge *puffeth vp*: but

Puhites

1.*Chro.*2.53. the Ithrites & the *Puhites*

Pul.

2.*Kin.*15.19. & *Pul* the King of Assyria
and Menahem gaue *Pul*
1 *Chro.*5.26. stirred vp the spirit of *Pul*
Isay 66.19.*Pul* and Lud that draw the

Pull.

1 *Kin.*13.4.he could not *pull* it in again
*Psal.*31.4.*pull* me out of the net that
*Isay*22.19 starre shall he *pull* thee down
*Ier.*1.10.and to *pull* down & to destroy
12.3 *pull* them out like sheep, for the
18.7.and to *pull* down and to destroy
26.6. and not *pull* them down, and I
*Ezek.*17 9.shall he not *pull* vp the roots
*Mich.*7.8.ye *pulled* off the robe with the
Luke 6.42.to *pull* out the mote that is

Pulled/ing:

*Gen.*8 9.*pull*, her in vnto him into the
19.10. *pulled* Lot into the house to
*Ezra.*6.11 let timber be *pulled* down
*Lam.*3.11,and *pulled* me in pieces, he
*Amos.*9.15. shall no more be *pulled* vp
*Zech.*7.11. & *pulled* away the shoulder
2.*Cor.*10.4 to the *pulling* down of strong
*Iude.*23.*pulling* them out of the fire, hau.

Pulpit:

*Neh.*8.4.stood vpon a *pulpit* of wood

Pulse:

2.*Sam.*17.28.lentiles & parched *pulse*
*Dan.*1.12.let them giue *pulse* to eate
16.should drink, & giue them *pulse*

Punish:

*Levit.*26.18.I will *punish* you. So Verse 24.
*Pro.*17.26.to *punish* the iust is not good
*Isay.*10.12. I will *punish* the fruit of the
13.11.I will *punish* the world
24.21. the Lord shall *punish* the host
26.21 comes out of his place to *pun.*
27.1.shall *punish* Leuiathan the
*Ier.*9.25.Ile *punish* all them that are
11,22.the young men, I will *punish*
13.21. wherefore shall *punish* thee, for
23.34.I will even *punish* that man
25.12.Ile *punish* the King of Babylon
27.1.that nation will I *punish* saith
29.32.behold, Ile *punish* Shemaiah
30.10.I will *punish* all that oppresse
36.31.I will *punish* him and his seed
44.13.I will *punish* them that dwell
29.I will *punish* you in this place
46.25.I will *punish* the multitude of
50.18.I will *punish* the K.of Babylon
51.44. I will *punish* Bel in Babylon
*Hos.*4.9.I *punish* them for their waies
14.I will not *punish* your daughters
12.2. I will *punish* Iacob according
Amos 3.2.therefore will I *punish* you
Zeph 1.8.I will *punish* the Princes and
9 I will *punish* all them that leape
12. & *punish* them that are settled
*Act.*4.21. how they might *punish* them

Punished:

*Exod.*21.20.he shall be surely *punished*.
21. he shall not be *punished*, for he
22. he shall be surely be *punished* as
*Ezra.*9.12.hast *punished* vs less than
Iob 31.11.to be *punished* by the Iudges
28. to be *punished* by the Iudge, for
*Prov.*21.11. when the scorner is *punished*,
22.3. simple passe on and are *punished*
*Ier.*44.12. as I have *punished* Ierusalem
50.9.as I have *punished* the King of
*Zeph.*2.7. howsoever I *punished* them
*Zech.*10.3.I *punished* the goats for the
Act 22.5 bound to be *punished*
26 11.I *punished* them oft in
2 *Thes* 1.9. who shall be *punished* with
2 *Pet.*2.9.the vnrighteous to be *punished*

Punishment.

*Gen.*4.13.my *punishm* is greater than
19.15. See the *Margin*
*Leu.*26.41. then accept of the *punism.*
43. they shall accept of the *punis.*
1 *Sam.*28.10. there shall no *punishment*
Iob 31.3.and a strange *punishm.* to the
*Pro.*19.19 shall suffer *punishm* for if
*Lam.*4.39.for the *punishment* of his sins
4.6. for the *punishm.* of the iniquity
22.the *punishment* of thine iniquity
Ezek 14.10.shall bearе the *punishment*
the *punishment* of the Prophet
shall be even as the *punishment*
*Zech.*14.19. this shall be the *punishment*
and the *punishment* of all nations
*Mat.*25.46.into euerlasting *punishment*
2.*Cor.*2.6.sufficient is this *punishment*
Heb 10.29.of how much sorer *punism.*
1 *Pet.*2.14.or the *punishment* of euill

Punishments.

Iob 19.29.wrath brings the *punishments*
*Psal.*149.7. & *punism.* on the people.

Punon.

*Num.*33.42.& pitched in *punon*, & they
43.departed from *punon*, & pitcht

Pur.

*Est.*3.7.they cast *pur* that is the lot
9.24. and had cast *pur* that is the lot

Purchase.

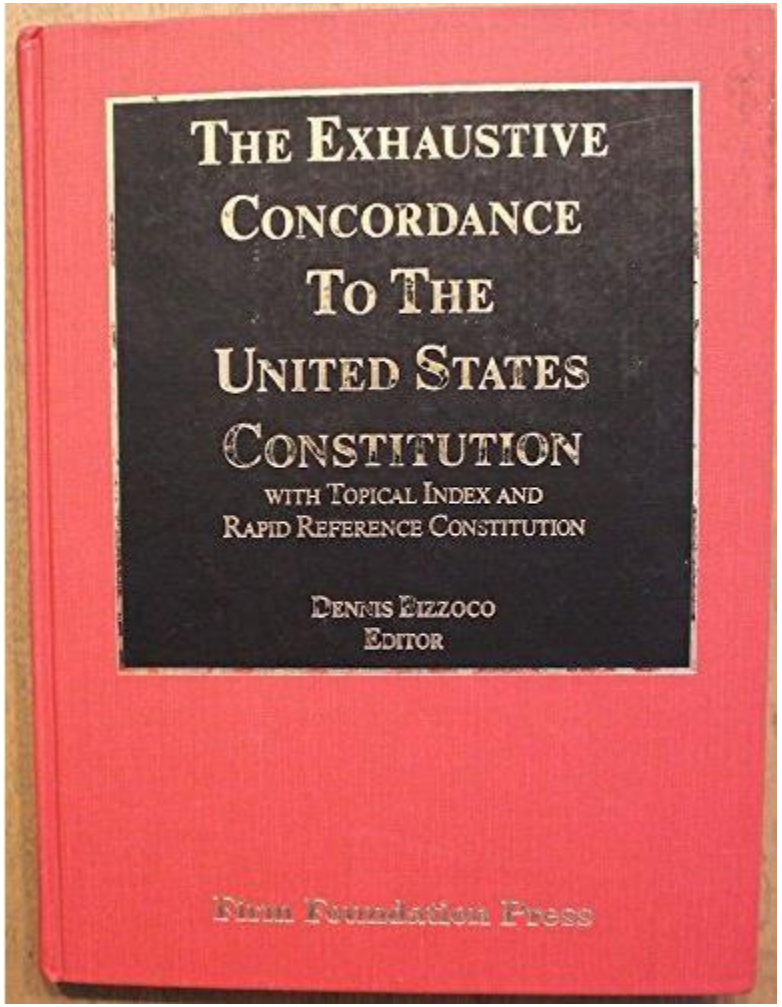
*Gen.*49.32.the *purchase* of the field &
*Levit.*25.33. but if a man *purch.* of the
*Ier.*32,11. the euidence of the *purchase*
12,the booke of the *purcha.* before
1 *Tim.*3.13.*purchase* to themselues

Purchased.

*Gen.*15.10.which Abraham *purchased*
*Exod.*15.16. which thou hast *purchased*
Ruth 4.10.haue I *purchased* to be my
*Psal.*78.54. his right hand had *purchased*
Act 1.18.this man *purchased* a field
8.20.to be *purchased* with money
20.28. *purchased* with his own blood
*Ephes.*1, 14. of the *purchased* possession

Pure.

*Exod.*30,23,*pure* myrrhe 500.sekles
34. with *pure* frankincense of each
35. tempered together *pure* & holy
37.29. and the *pure* incense of sweete
Levit 24.2.*pure* oyle olive beaten for
6. vpon the *pure* Table before the
7. thou shalt put *pure* frankincense
2 *Sam.*22,27.with the *pure* thou wilt
shew thy selfe *pure*, and with
1 *Chro.*13.11.vpon the *pure* Table, and
Iob 4.17.shall man be more *pure* than
8.6.if thou wert *pure* and vpright
11.4.my doctrine is *pure*, and I am
16,17,my hands, also my prayer is *pure*
25.5.stars are not *pure* in his sight
*Psal.*12.6. words are *pure* words, as silver &c.
18.26.with the *pure* thou wilt shew
thy selfe *pure*, and with the
19.8, commandement of the Lord is *pure*
24.4.clean hands & *pure* heart
119.140.thy word is very *pure*, therefore
*Pro.*15


THE EXHAUSTIVE
CONCORDANCE
TO THE
UNITED STATES
CONSTITUTION
WITH TOPICAL INDEX AND
RAPID REFERENCE CONSTITUTION
DENNIS EIZZOCO
EDITOR
Firm Foundation Press

Term-Document Incidence Matrix Fig 1.1 IIR

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
ANTHONY	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

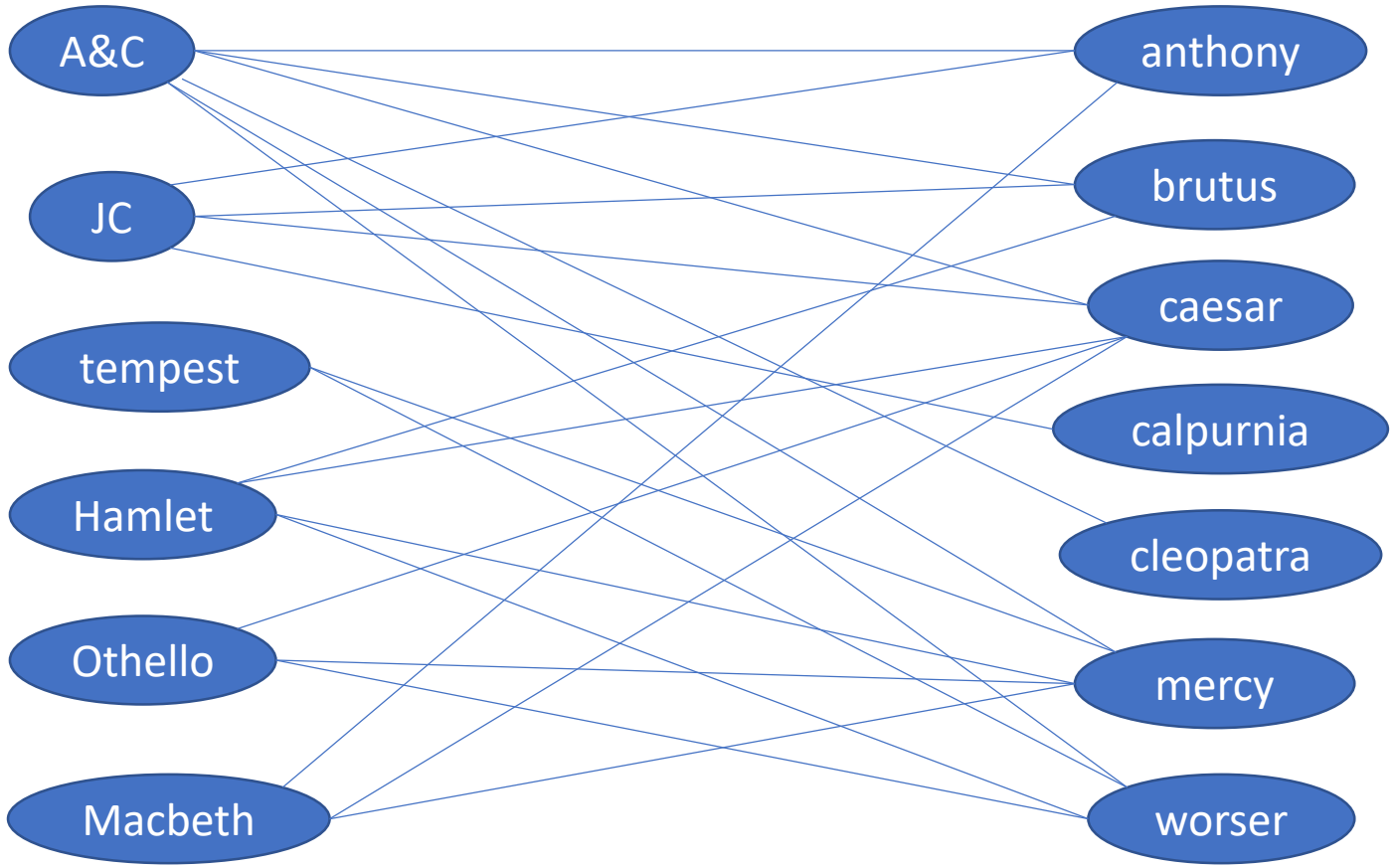
...

Entry is 1 if term occurs. Example: CALPURNIA occurs in *Julius Caesar*.
Entry is 0 if term doesn't occur. Example: CALPURNIA doesn't occur in *The tempest*.

A Large but Sparse Matrix

- Size of incidence matrix: number of documents times number terms ! too large for large collections
- But the matrix is very sparse – mostly 0s, few 1s.
  - Unless, a feature is shared by the most (the word “a”).
  - In that case, we may flip the feature (“occur” becomes “does not occur)
- Mathematically, it is a bipartite graph.
- The 1s in the incidence matrix are edges of the graph.
- All we need is an edge list file for storage.

The Bipartite Graph



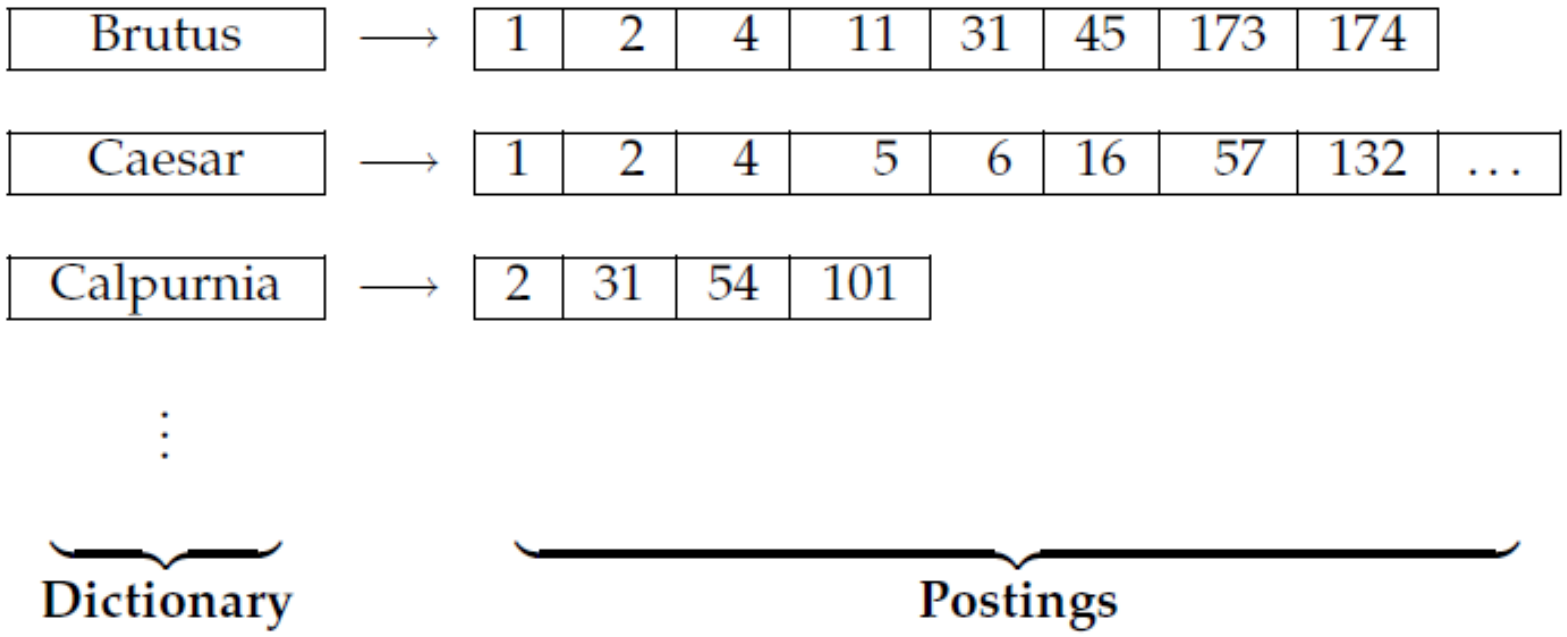
Representation of a Bipartite Graph

- The adjacency matrix (the term-document incidence matrix)
- The edges list
  - (A&C, anthony), (A&C, brutus), (JC, brutus),…
- A map from documents to sets of terms.
  - Hamlet : {brutus, caesar, mercy, worser}
  - Othello: {caesar, mercy, wroser}
- A map from terms to sets of documents. (Inverted index)
  - anthony: {A&C, JC, Macbeth}
  - brutus: {A&C, JC, Hamlet}

Inverted Index

- For each term t, we store a list of all documents that contain t.
  - A list of sets of terms
  - = For each term t, we store the 1s in its row in the incidence matrix
- It allows us to access all documents containing a term, instead of all terms in a document, which is the raw data, in an efficient way.
- In information retrieval, we may want to find or rank all documents containing a subset of terms.
- For this purpose, the inverted index is a necessary preprocessing step of the raw data, or a list of sets of terms.

Dictionary and Postings



▶ **Figure 1.3** 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.

Example Documents

Doc 1
I did enact Julius Caesar: I was killed
i' the Capitol; Brutus killed me.

Doc 2
So let it be with Caesar. The noble Brutus
hath told you Caesar was ambitious:

Making the Inverted Index

1. Collect the documents to be indexed:
   Friends, Romans, countrymen. | So let it be with Caesar | ...

2. Tokenize the text, turning each document into a list of tokens:
   Friends | Romans | countrymen | So | ...

3. Do linguistic preprocessing, producing a list of normalized tokens, which are the indexing terms: friend | roman | countryman | so | ...

4. Index the documents that each term occurs in by creating an inverted index, consisting of a dictionary and postings.

Tokenization (Edges of the Bipartite Graph)

term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i′	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Sorting

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i′	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Dictionary and Postings Lists

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i′	1	→	1
it	1	→	2

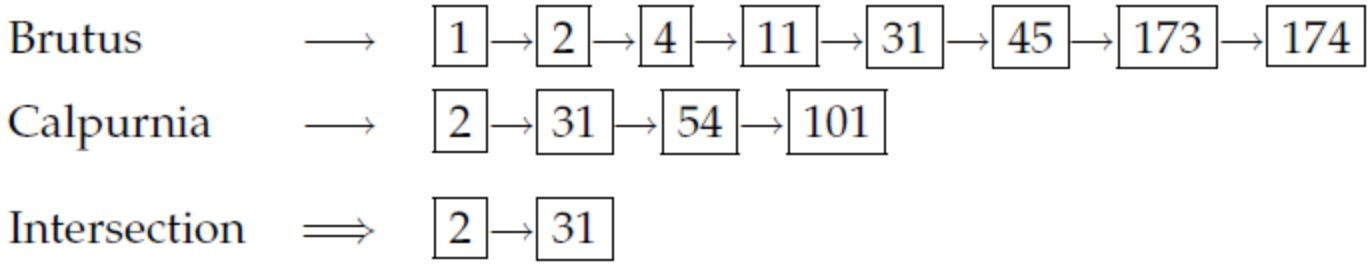
A Map from Terms to Sets of DocIDs

julius	1	→	1		
killed	1	→	1		
let	1	→	2		
me	1	→	1		
noble	1	→	2		
so	1	→	2		
the	2	→	1	→	2
told	1	→	2		
you	1	→	2		
was	2	→	1	→	2
with	1	→	2		

Processing Boolean Expressions

- Consider the query: Brutus AND Calpurnia
- To find all matching documents using inverted index:
  - 1 Locate Brutus in the dictionary
  - 2 Retrieve its postings list from the postings file
  - 3 Locate Calpurnia in the dictionary
  - 4 Retrieve its postings list from the postings file
  - 5 Intersect the two postings lists
  - 6 Return intersection to user

Intersection of Postings Lists



▶ **Figure 1.5** Intersecting the postings lists for Brutus and Calpurnia from Figure 1.3.

Inverted Index as a dict (map)

IR1B - Notepad

File Edit Format View Help

```
# IR1B.py CS5154/6054 cheng 2022
# read lines from a text file
# turn each line into a list of tokens
# make the inverted index
# retrieve documents using boolean queries
# Usage: python IR1B.py

f = open("bible.txt", "r")
docs = f.readlines()
f.close()
invertedIndex = {}
for i in range(len(docs)):
    for s in docs[i].split():
        if invertedIndex.get(s) == None:
            invertedIndex.update({s : {i}})
        else:
            invertedIndex.get(s).add(i)

word1 = 'punishment'
word2 = 'transgressions'
for j in invertedIndex.get(word1) & invertedIndex.get(word2):
    print(j, docs[j])
```

Boolean Queries

- The example was a simple conjunctive query . . .
- . . . the Boolean retrieval model can answer any query that is a Boolean expression.
  - Boolean queries are queries that use and, or and not to join query terms.
  - Views each document as a set of terms.
  - Is precise: Document matches condition or not.
- Primary commercial retrieval tool for 3 decades
- Many professional searchers (e.g., lawyers) still like Boolean queries.
  - You know exactly what you are getting.
- Many search systems you use are also Boolean: search system on your laptop, in your email reader, on the intranet etc

Pros and Cons of the Boolean Model

- Key property: Documents either match or don’t.
- Good for expert users with precise understanding of their needs and of the collection.
- Also good for applications: Applications can easily consume 1000s of results.
- Not good for the majority of users
- Most users are not capable of writing Boolean queries . . .
  - . . . or they are, but they think it’s too much work.
- Most users don’t want to wade through 1000s of results.
- This is particularly true of web search.

Problems with Boolean Search

- Boolean queries often result in either too few (=0) or too many (1000s) results.
- Query 1 (boolean conjunction): [standard user dlink 650]
  - → 200,000 hits – feast
- Query 2 (boolean conjunction): [standard user dlink 650 no card found]
  - → 0 hits – famine
- In Boolean retrieval, it takes a lot of skill to come up with a query that produces a manageable number of hits.

No Problems with Ranking

- With ranking, large result sets are not an issue.
- Just show the top 10 results and the user won’t be overwhelmed
- Premise: the ranking algorithm works: More relevant results are ranked higher than less relevant results.

Assignment 1: due 8/26/2022

- Add “import time” at the beginning of IR1A.py and IR1B.py.
- Run t1 = time.process_time_ns() (time in nanoseconds) before and after a section of the code and output the difference between two times.
  - Feel free to use variables t2, t3, t4,… so you can print(t2-t1) etc. at the end of the program.
- Find the process times for the following tasks.
  - Reading the collection of documents (both IR1A and IR1B).
  - Making the inverted index (IR1B).
  - Boolean retrieval with an expression with two terms (IR1A without and IR1B with the inverted index).
    - You may want to have a loop of 100 iterations of this block of code to see the difference.

Hints

- How to do the same thing 100 times?
  - for rep in range(100):
- Using the timer (example: number of nanoseconds reading data)

```
import time

t1 = time.process_time_ns()
f = open("bible.txt", "r")
docs = f.readlines()
f.close()
t2 = time.process_time_ns()
print(t2 – t1)
```