Data Mining and Machine Learning

Lecture 2

Statistical [illegible] exts

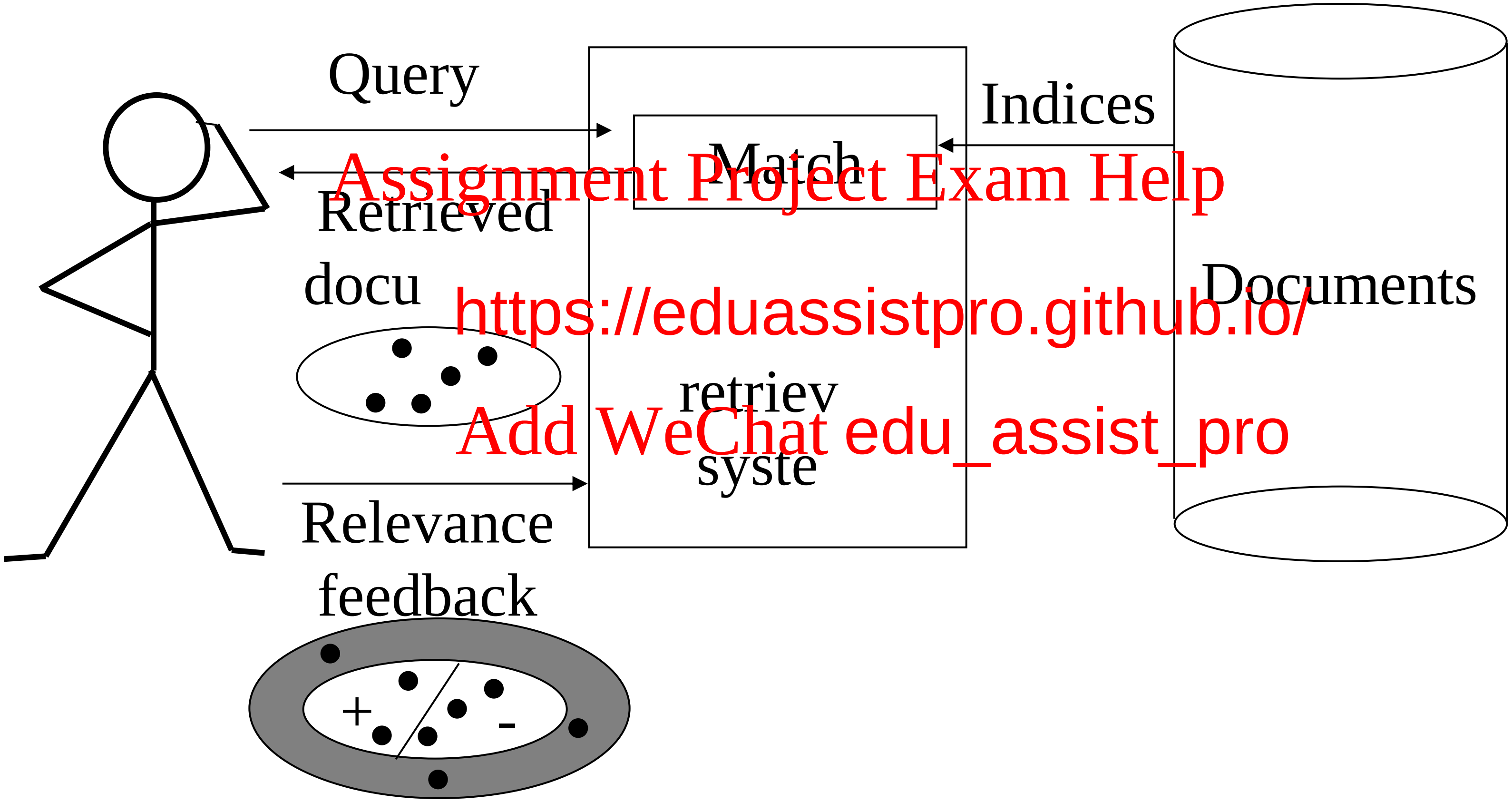
Peter Jančovič

UNIVERSITY OF
BIRMINGHAM

Objectives

- Understand different approaches to text-based IR
  - Rationalism vs Empiricism
- “Bundles of
- Introduction
- Statistical analysis of wor e in text
- Zipf’s Law
- Examples

A Basic Search Engine [Belew]



Information Retrieval Components

- The **Documents**
  - Identify words which are ‘important’ for discriminating between documents, and how important they are
- The **Index**
  - Specifies th [illegible] ese ‘keywords’ and the docume [illegible]
- The **query**
- **Matching**
  - Measuring the **similarity** between the query and each document
- Retrieved documents
- **Assessment** and **Relevance Feedback**

Example Text

“There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but hen there was no company, dined had brought with it clouds so sombre, and a rain so , that further out-door exercise was now out of .”

Charlotte Brontë, “Jane Eyre”, first paragraph

“Jane Eyre” extract

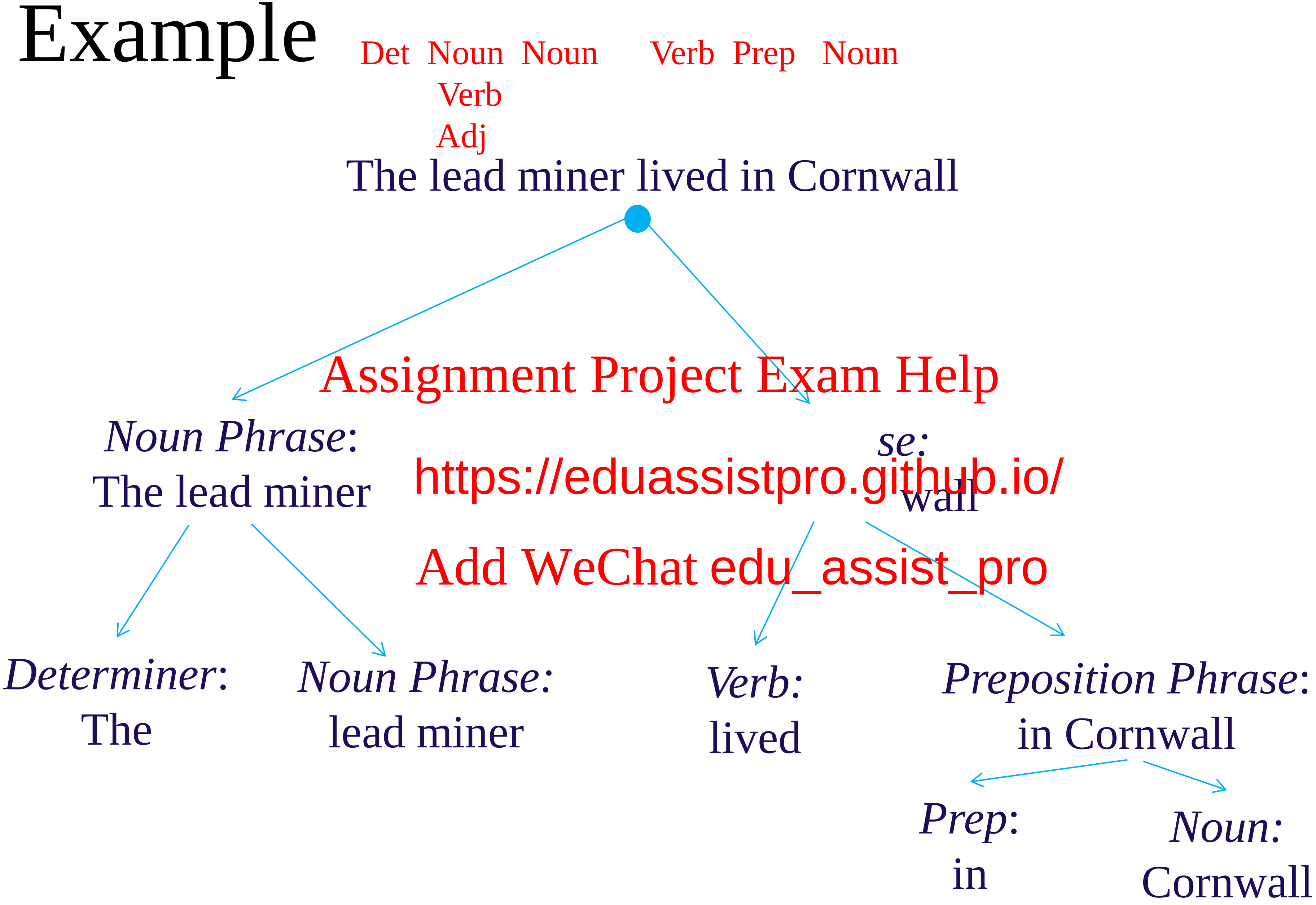
- What is it **about**?
- How do you know?
- What is your anding what a text is **about**
- What are the component t
  - Exercise (walk, wandering,
  - Gardens (shrubbery)
  - Weather (cold, winter, wind, clouds, rain)

Structure in text

- Words
  - **Keywords** (some words are more important than others)
  - *Cold*, *Walk* and *Shrubbery* are important
  - *There*, *and* and *that* are not

- Sentences (Gra
  - Word seque nderstand and to remove ambiguity
  - ‘Parts of speech’
    - *The lead miner lived in Cornwall*
    - *Keep that dog on a lead!*
    - *He won the lead role in the new film*

Example


Det
Noun
Verb
Adj
Noun
Verb
Prep
Noun
The lead miner lived in Cornwall
Noun Phrase:
The lead miner
se:
wall
Determiner:
The
Noun Phrase:
lead miner
Verb:
lived
Preposition Phrase:
in Cornwall
Prep:
in
Noun:
Cornwall

Rationalism vs. Empiricism 1

- Rationalism:
  - Try to copy human language processing

- Two questions:
  - Do we und w we do it?
  - Is our knowledge ‘computa ul’? I.e. is our knowledge sufficiently ‘sol rt algorithms and computer programs?

- These are topics in Natural Language Processing (NLP) and Computational Linguistics

Available knowledge

- Word inventories
  - Electronic dictionaries
- Word forms (noun, verb etc)
  - Available in electronic dictionaries

- Word meani

  - Expressed in terms of predi roperties)

- Grammar / syntax
  - Grammatical rules
- Parsers
  - Apply grammatical rules to a word sequence to determine if it is grammatical and, if so, its grammatical structure

Natural Language Processing

- Use word sense and meaning plus grammatical structure to infer ‘meaning’
- Several problems
  - Grammar [illegible] – accept non-grammatica[illegible]
  - Grammar may be too restri[illegible] valid sentences
  - The number of interpretations of a simple sentence may be huge (“I saw the man on the hill with the telescope”)
- Language is dynamic and changing

Rationalism vs. Empiricism 2

- Empiricism ("Big Data")
  - Use large corpora of text instead of human knowledge
  - Use machine-learning to identify important structure and relationships
  - Quantify the
  - Rely on qua red from these large corpor on

- For example:
  - For each word *w* define a number *U*(*w*) which indicates how **useful** *w* is for Information Retrieval
  - Invent **algorithms** to find the **most useful** words
  - Invent **measures** of the **similarity** between queries and texts

Rationalism vs Empiricism

- Need sophisticated computationally useful models of language and semantics to infer meaning
- Rational approaches accommodate complex structure but may be fragile and hard to generalise
  - She ran, wa
- Models base g (ML) are conceptually simpler but h ained automatically
- NLP currently outperformed in most applications by methods based on ML – "Deep Learning", "Deep Neural Networks"
- Progress – Amazon Echo/Alexa

‘Bundles of Words’ approaches

There was no possibility of taking a walk that day. We had been wandering, indeed, in the leafless shrubbery an hour in the morning; but si (Mrs. Reed, when th company, dined early) the cold winter wind had brought with it clouds so sombre, and a rain so penetrating, that further out-door exercise was now out of the question

the 4	early 1	walk 1
was 3	exercise 1	wandering 1
a 2	further 1	we 1
had 2	hour 1	when 1
in 2	indeed 1	wind 1
no 2	it 1	winter 1
of 2	leafless 1	with 1
	morning 1	
	mrs 1	
	now 1	
	out 1	
	-door 1	
	etrating 1	
	sibility 1	
	stion 1	
clouds 1	rain 1	
cold 1	reed 1	
company 1	shrubbery 1	
day 1	since 1	
dined 1	sombre 1	
dinner 1	taking 1	

What is a word?

- Tokens ≡ things separated by white space
- Hyphenation
  - Database ≡ Data-base?

- Case
  - “the bath s

  - “the brown house” vs “the ”

- Morphology
  - retrieval, retrieve, retrieved, retrieving,…
- Punctuation
  - The ‘honest’ politician vs the honest politician

Some arbitrary choices…

- Tokens ≡ things separated by white space
- Ignore case:
  - London ≡ london
  - BBC ≡ bbc
- Ignore non-alphanumerics d end of token:
  - ‘honest’ ≡ honest. ≡ honest! honest

Statistical Analysis of Word Occurrence in Texts

- `zipf.c`
  - ANSII C program for simple analysis of texts
  - Finds the set of different tokens in the text
  - Counts how ccurs
  - Orders wor r of times they occur in the text (their <u>rank</u>)
  - Prints out the result, and
  - Stores results in a file `results`

zipf.c

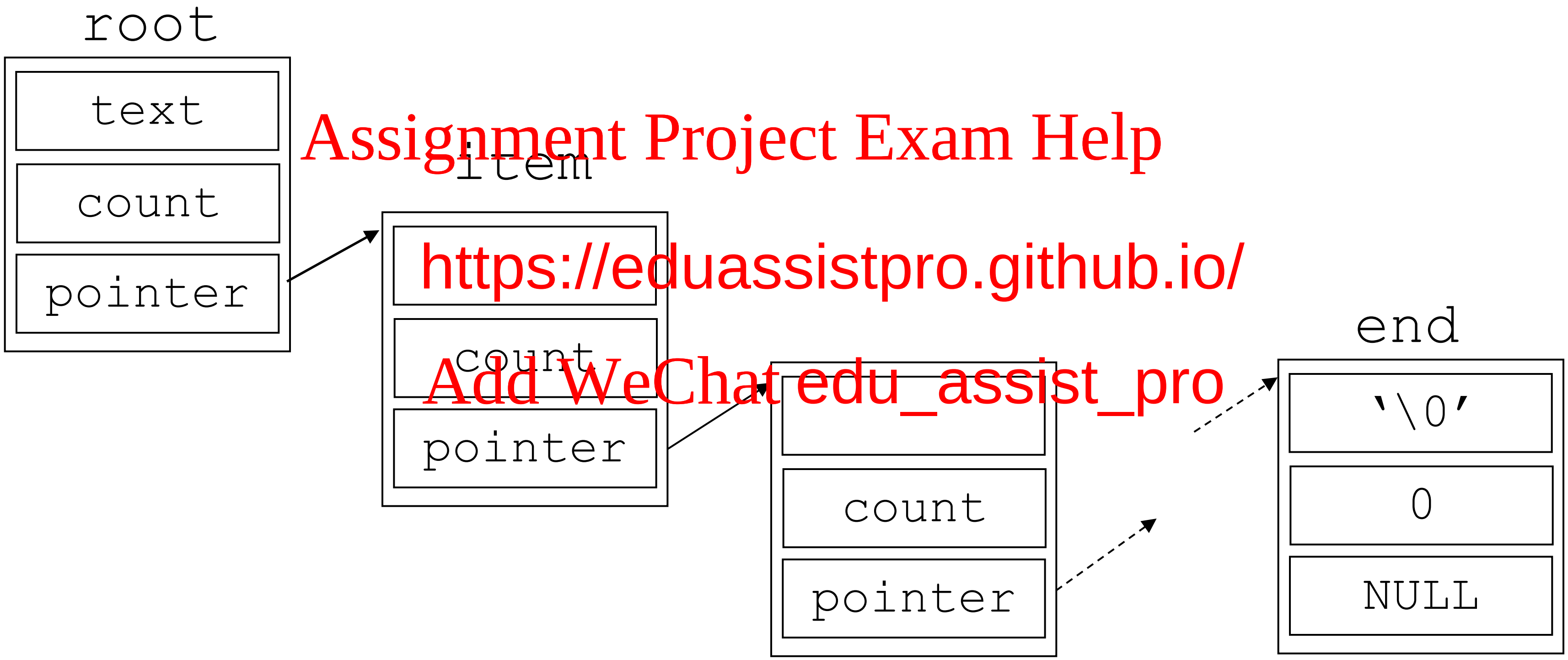
```
/* Function to read next word from text */
int nextWord(FILE *ip, char *token)
{
  int x;
  int c;
  for (c=                         ) token[c]='\0';
  x=fscan
  if (x != EOF)
    {
      upper2lower(token);
      removePunct(token);
    }
  return x;
}
```

zipf.c

```
/* struture to store linked list of words */
struct item {
   char *text;
   int count;
   struct ite
};
```

zipf.c

- Linked List



Compilation of “Data Mining” C code

- Simple ANSII C
- OS independent – should work on any platform with any ANSII-compliant C compiler

- Download fr e

- Compile using MS Visual T **command line**

- `cl zipf.c`

Statistical Analysis of Word Occurrence in Texts

- Complete novels available online: http://www.literature.org

ane Eyre”,
ontë, 1847

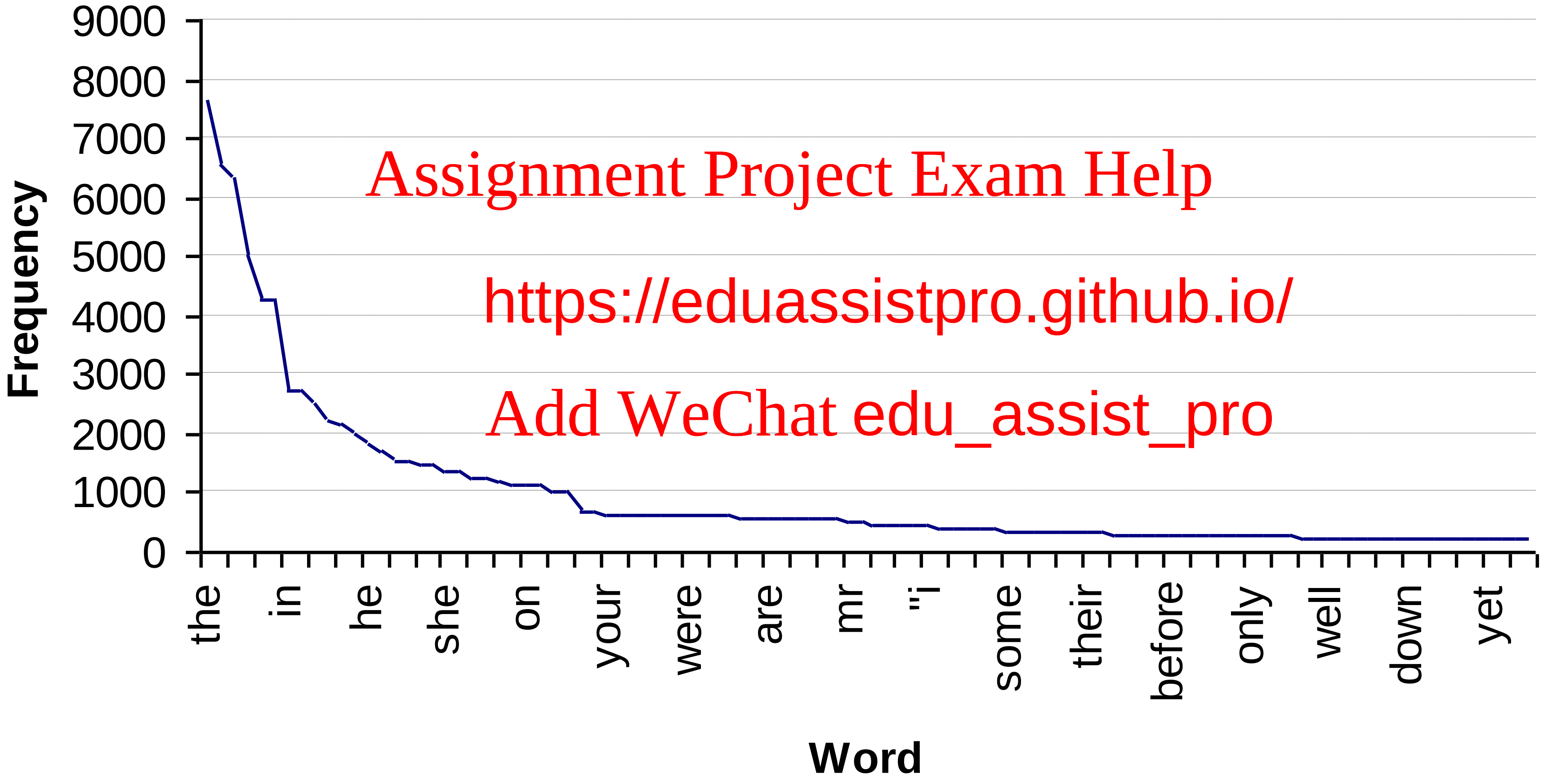
- Pen on - 489 pages
- 1,039 KBytes

“Top 10” words in “Jane Eyre”

Top 10		**101-110**		**7861-7870**	
the	7638	can	218	abate	1
i	6536	about	217	abbot’s	1
and	6335	looked	216	abigail	1
to	5028	t		ilities	1
of	4299	s		ode--whether	1
a	4294	day	2	es	1
in	2717	any	2	inable	1
you	2709	own	203	abrid	1
was	2495	much	200	abruptness	1
it	2219	come	199	absences	1

Different words 15,827, Total words 184,640

Word frequency plot for “Jane Eyre”

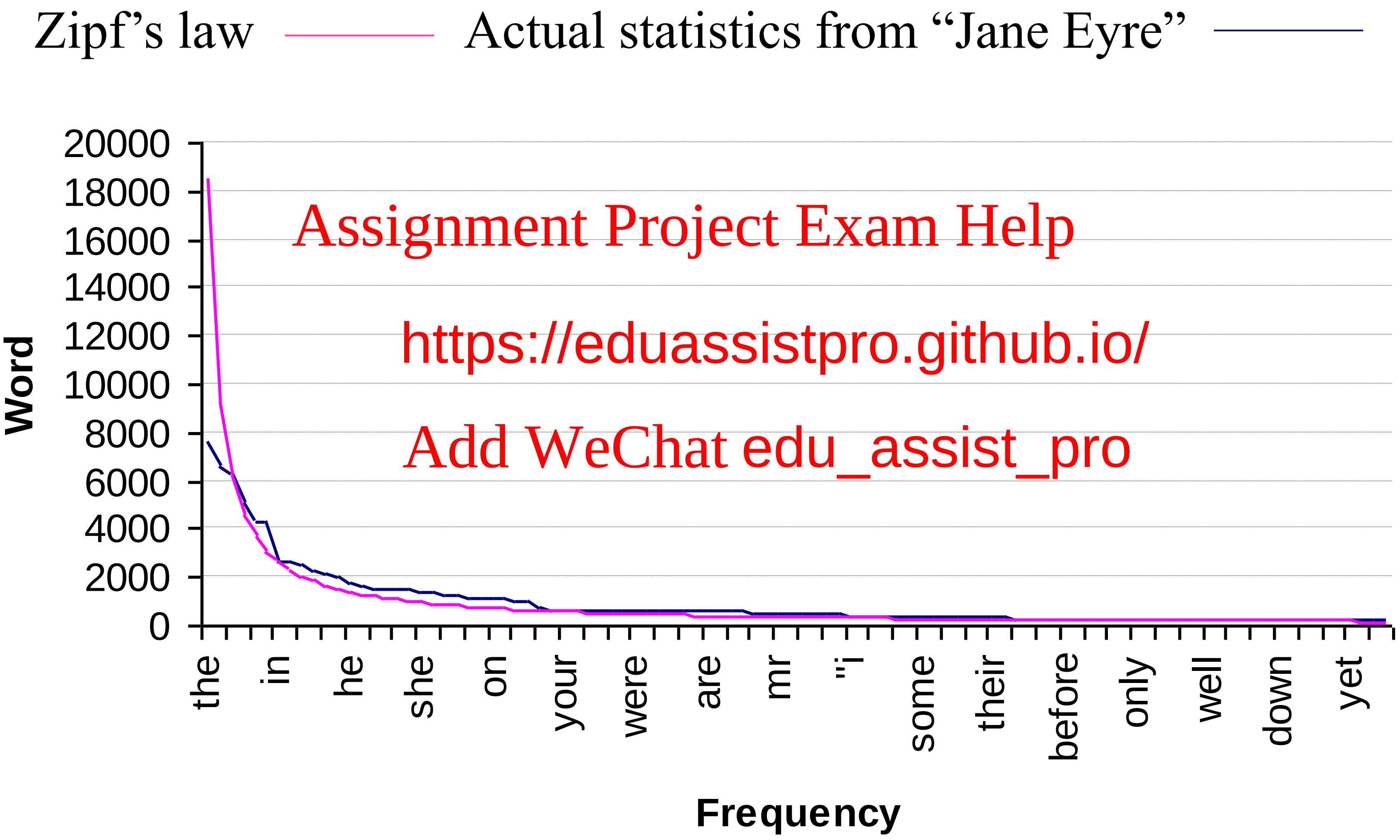


Zipf's Law

- George Kingsley Zipf (1902-1950)
  - For each word *w*, let *F*(*w*) be the number of times *w* occurs in the corpus
  - Sort the wo
  - The word's n will be fitted closely by the function:

$$F(r) = \frac{C}{r^{\alpha}},$$

Zipf's Law



UNIVERSITY OF BIRMINGHAM

Zipf's Law (logarithm form)

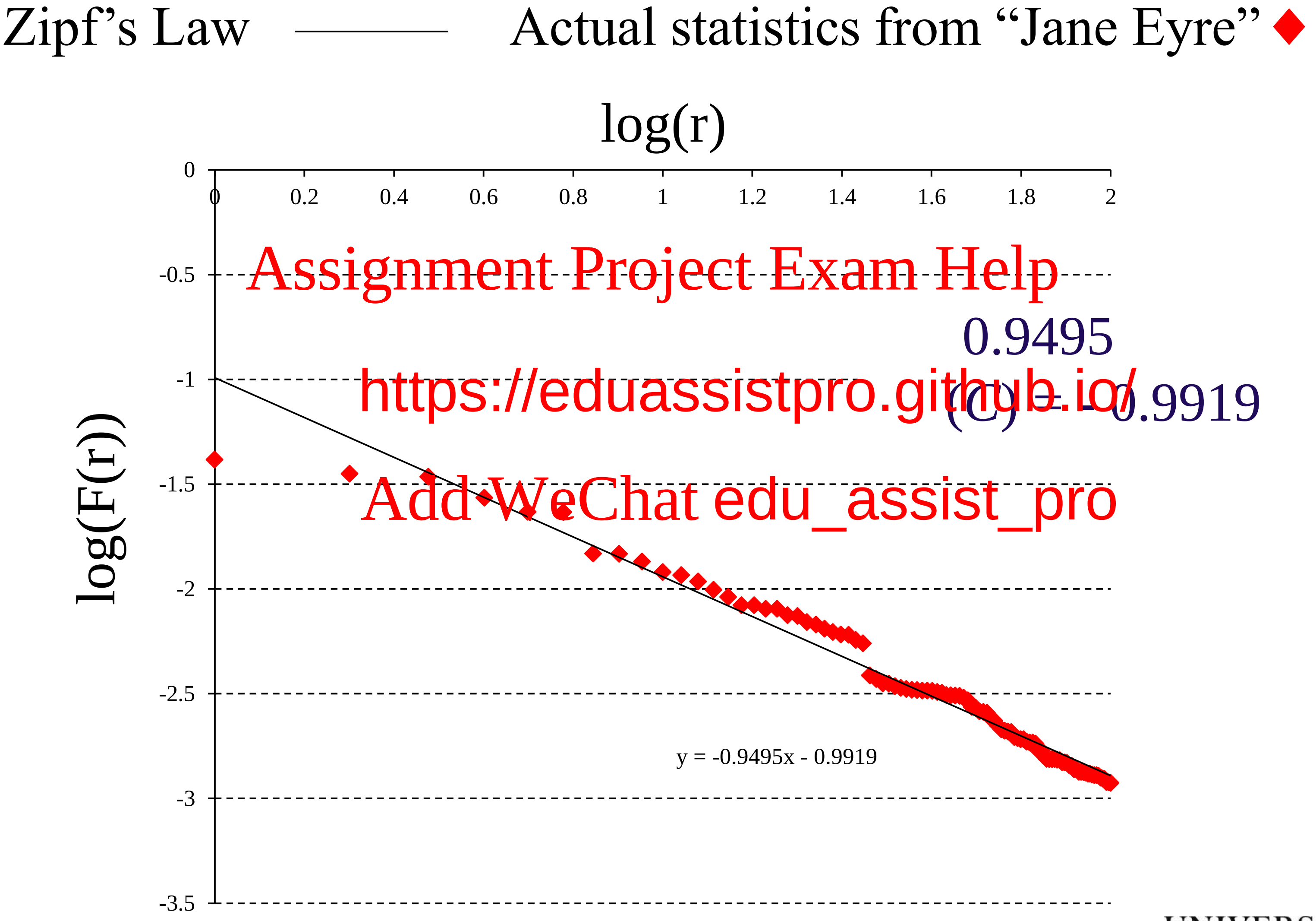
$$F(r) = \frac{C}{r^{\alpha}}, \text{where } \alpha \approx 1, C \approx 0.1$$

Therefore,

$$\log(F(r)) = \log(C) - \alpha \log(r)$$

- On a log-log scale, Zipf's [illegible]ts a straight-line relationship between log-rank and log-frequency, where α is the slope of the line and *C* is the intersection with the vertical axis
- This provides a way to estimate *C* and *α*

Zipf’s Law (logarithm form)

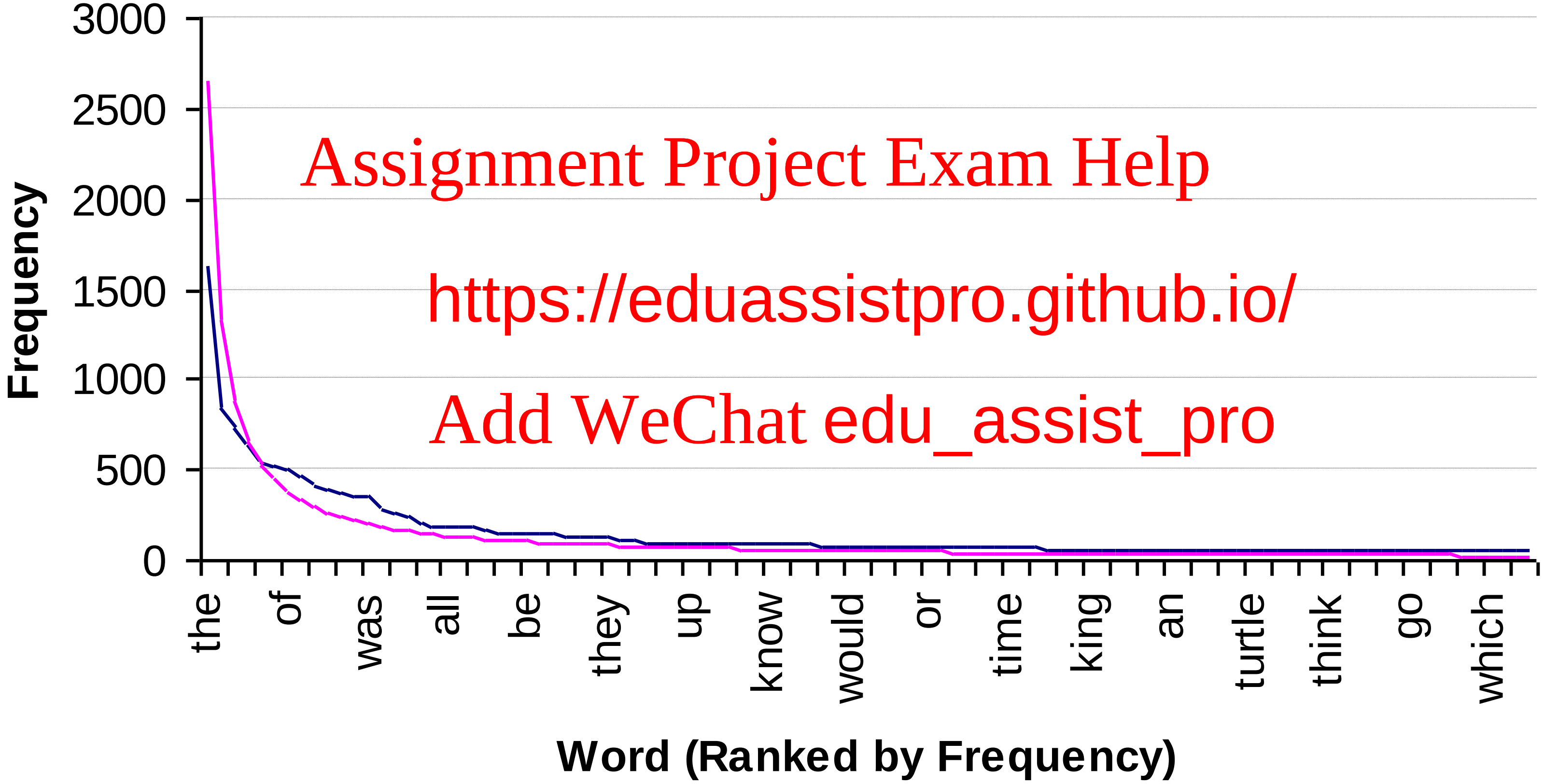


0.9495

(C) = -0.9919

Word Frequency Plot: “Alice in Wonderland”

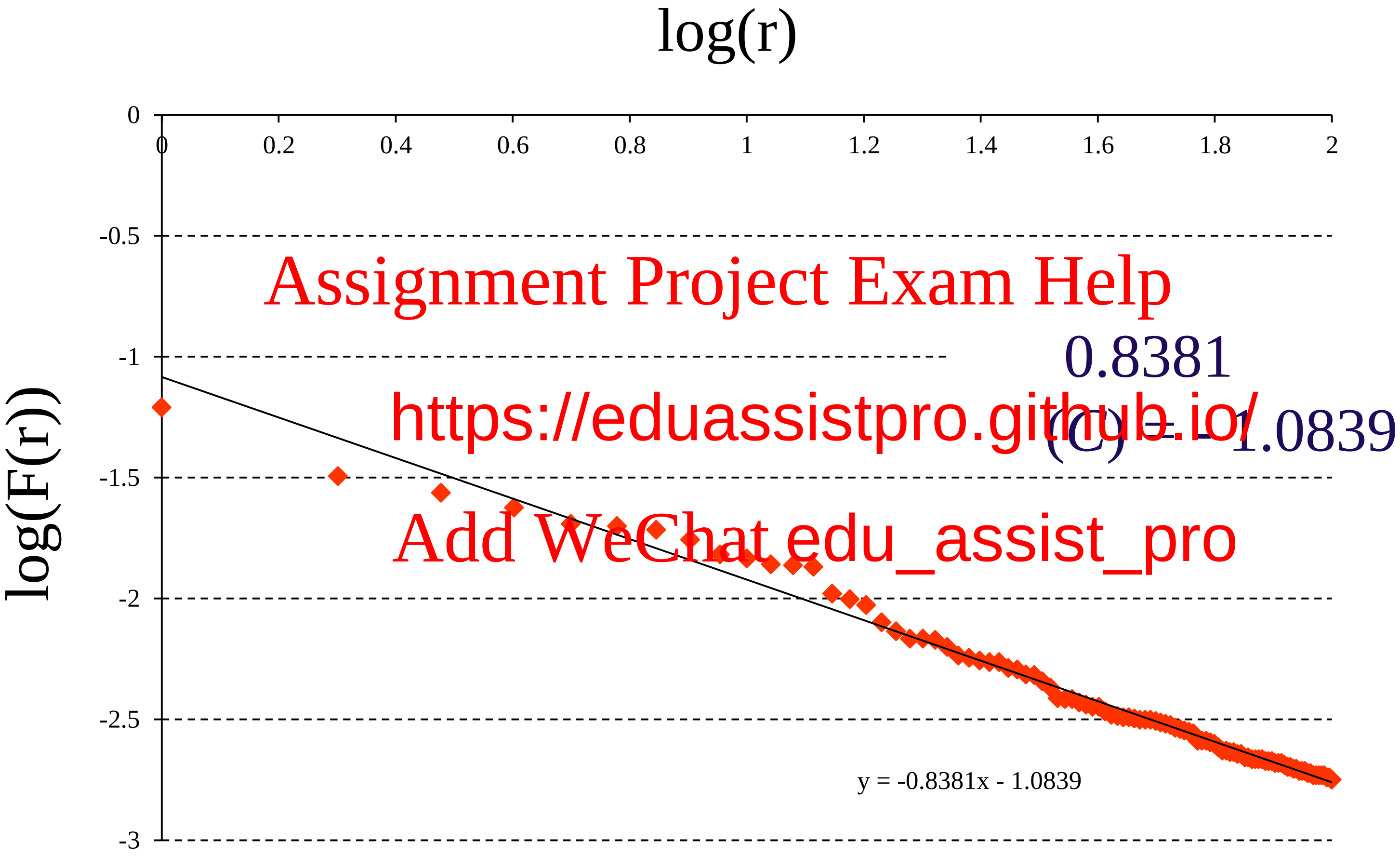
Zipf’s law ——— Actual statistics from “Alice in Wonderland” ———



Different words 2,787, Total words 26,395

UNIVERSITY OF BIRMINGHAM

Log-log plot – Alice in Wonderland



0.8381

(C) = - 1.0839

Zipf vs “Pride and Prejudice”

Zipf’s law

Zipf vs "Journey to the West"

Zipf's law

Some non-text examples

- Mathematics Today, vol. 47, no. 5, October 2011
- “Urban maths – Zipf’s Law”
  - Populations of the countries of the world
  - UK new ca
  - Counts of first digit from 1, rices quoted in The Times

Populations of countries

Taken from: "Urban Maths Zipf's Law", Mathematics Today, vol. 47, no 5, October 2011

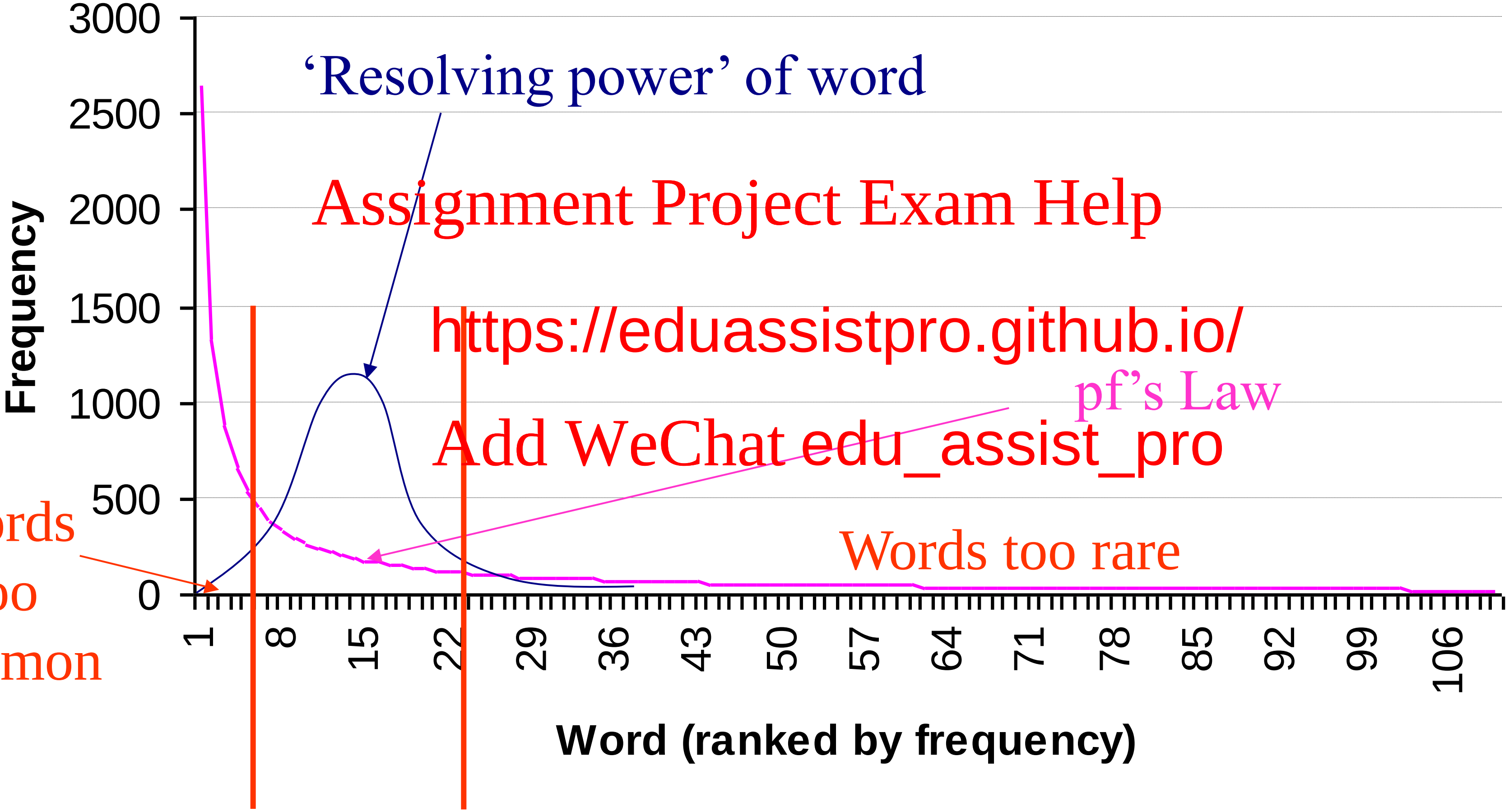
Zipf's Law

- Why does it hold?
- Is it relevant to Information Retrieval?

Why does Zipf's Law work?

- Zipf's law appears to reflect a number of factors:
  - The requirements of humans to communicate
    - Use as little effort as possible to successfully commu
  - Basic comb
  - The requirement of gramm 'glue' words
  - Author and topic vocabularies

‘Resolving Power’ of words



‘Resolving power’ of word

pf’s Law

Words too common

Words too rare

Upper cutoff

Lower cutoff

Summary

- Different approaches to text-based IR
- “Bundles of words” approaches
- Statistical an nce in text
- Zipf’s Law
- Examples