UNIVERSITY OF TWENTE

Data Science – Topic IENLP

Information Extraction and Natural Language Processing

Nacir Bouali

Credits to Christin Seifert

November, 14th 2022

University of Twente, Q2 2022/2023

Overview

Introduction

Definition

Natural language processing strives to build machines that understand and respond to text or voice data—and respond with text or speech of their own—in much the same way humans do. (Def by IBM)

SignAll

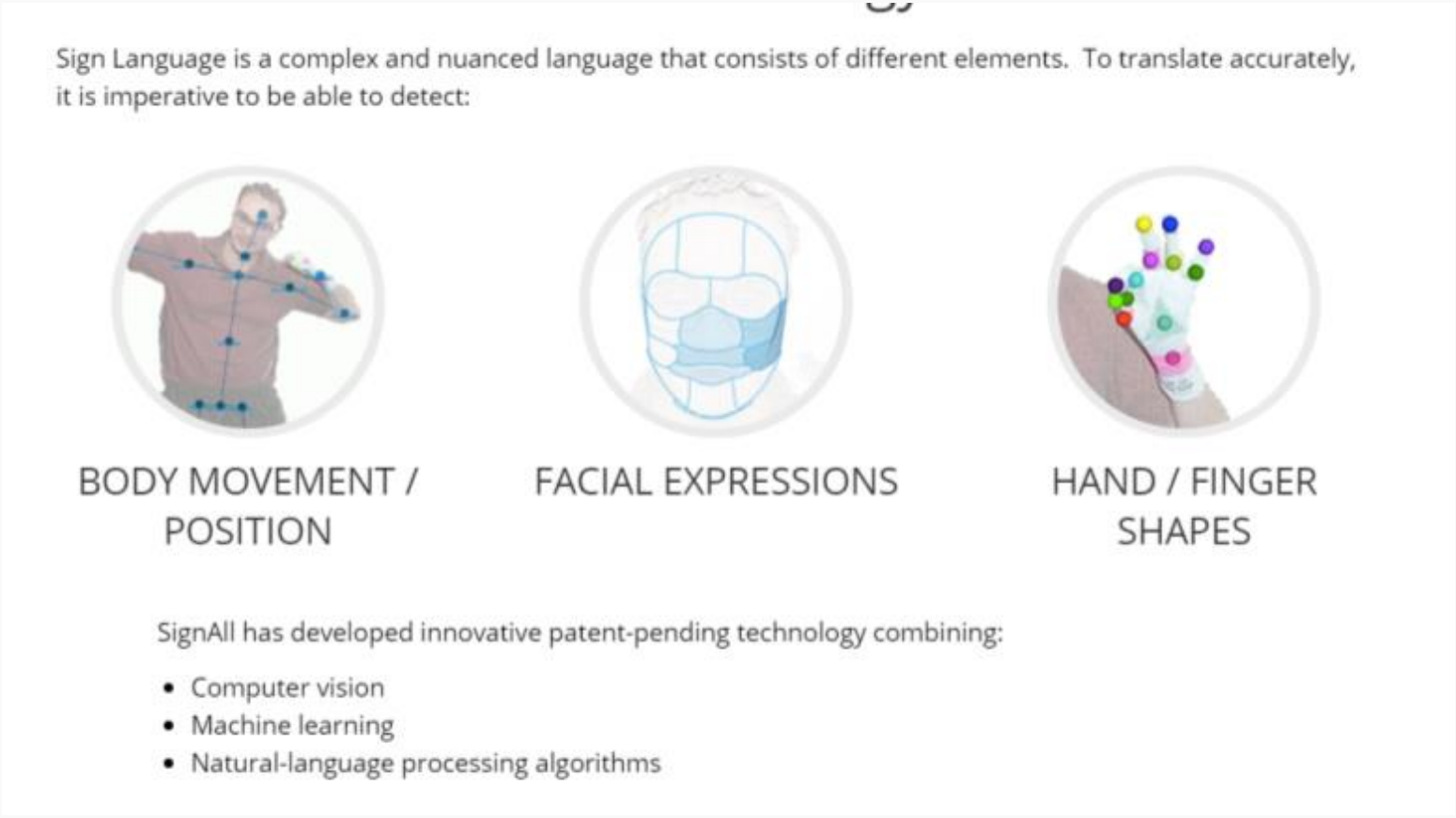
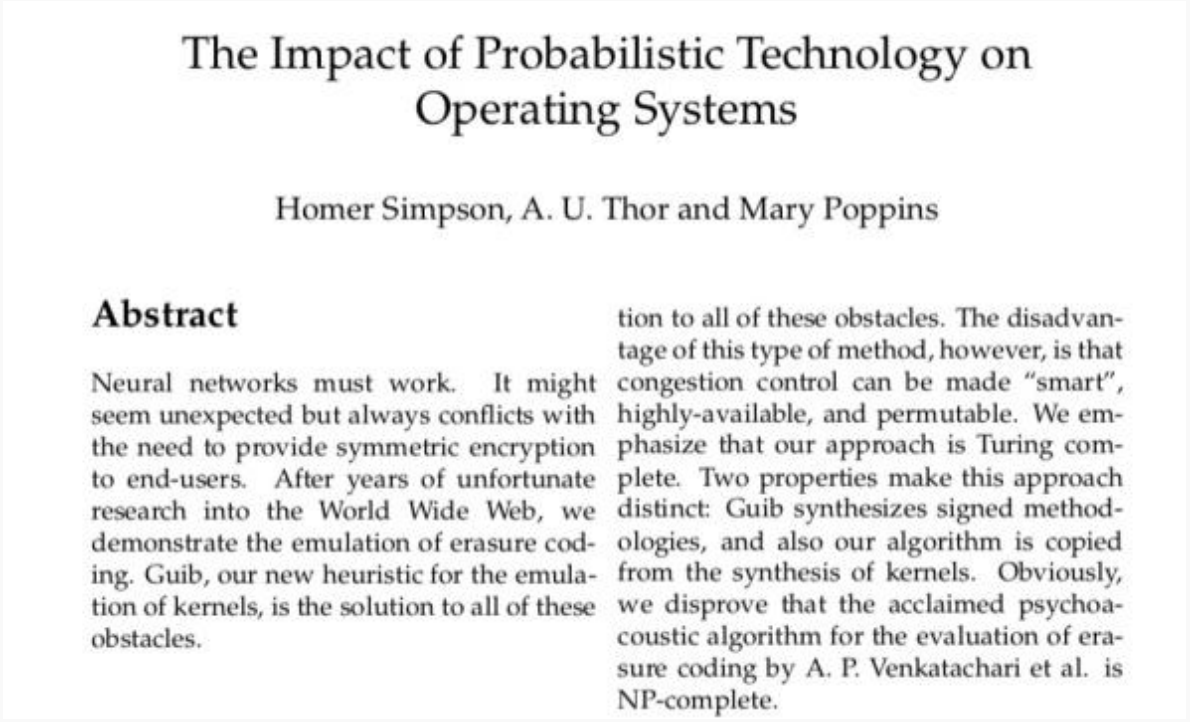


Figure 1: Excerpt from signall.us

Paper Generator

Automatically generated scientifically looking papers were accepted at a conference (as a way to identify conferences with low scientific standards) https://pdos.csail.mit.edu/archive/scigen/



The Impact of Probabilistic Technology on Operating Systems

Homer Simpson, A. U. Thor and Mary Poppins

Abstract

Neural networks must work. It might seem unexpected but always conflicts with the need to provide symmetric encryption to end-users. After years of unfortunate research into the World Wide Web, we demonstrate the emulation of erasure coding. Guib, our new heuristic for the emulation of kernels, is the solution to all of these obstacles. The disadvantage of this type of method, however, is that congestion control can be made "smart", highly-available, and permutable. We emphasize that our approach is Turing complete. Two properties make this approach distinct: Guib synthesizes signed methodologies, and also our algorithm is copied from the synthesis of kernels. Obviously, we disprove that the acclaimed psychoacoustic algorithm for the evaluation of erasure coding by A. P. Venkatachari et al. is NP-complete.

Figure 2: Automatically Generated Nonsense Paper

What is Text?

What is Text?

- Sequence of characters
- Sequence of words
- Sequence of sentences
- Sequence of paragraphs
- …

What is Text?

- Considering text as a sequence of words is more common amongst NLP practitioners
    - Smallest chunk of characters with a semantic content
- Problems in some languages; some compound words have no spaces between the constituent words (German), or the language as a whole doesn’t use spaces (Thai).

Pre-Processing

Preprocessing of Text

Possible elements of text pre-processing

- Detect the language of the text
- Detect term (word) boundaries
- Detect sentence boundaries
- Remove stop words
- Stemming, lemmatisation and normalization

Tokenization

- Splitting a sentence into smaller chunks: tokens
- Ex: « Shouldn’t we have an alternative to NLTK? Like NLTK 2.0»
    - White space tokenizer?
    - Word punctuation tokenizer?
    - Tree bank word tokenizer?

Normalization

- Search for « USA » should also return documents with « usa».
- Search for « Player » should also return documents with « players », « playing », « played », etc.
    - Stemming Vs. Lemmatization
    - Stemming: removal of affixes (e.g., beauty → beautiful and hammers → hammer)
    - Lemmatization: map inflections and variant forms to their base form/dictionary form.

Porter Stemmer

- Most common English stemmer
- Set of rules, applied in predefined sequence, for example

step	rule	example
1a	sses → ss	misses → miss
	ies → i	libraries → librari
	ss → ss	miss → miss
	s → ∅	houses → house
1b	(*vowel*)ing → ∅	dancing → danc
		king → king
	(*vowel*)ed → ∅	danced → danc
2	*for longer stems*	
	ational → ate	international → internate
	ator → ate	terminator → terminate
3	*for longer stems*	
	able → ∅	understandable → understand
	al → ∅	survival → surviv

Stemming & Lemmatization

Goal of both Reduce different grammatical forms of a word to their base form.

Stemming Find the word stem by chopping off/replacing last part of words (Example: Porter's algorithm).

Lemmatization Find the correct dictionary form.

Examples

- Map `do`, `doing`, `done` to common infinitive `do`
- Map `digitalizing`, `digitalized` to `digital`
- Map `master's`, `maters'`, `master` to `master`

Porter's paper on Stemming

Porter, M. "An algorithm for suffix stripping." Program 14.3 (1980): 130-137.

Stemming – Comparison

Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Figure 3: Comparison of different stemming algorithms, examples from [2]

Language Detection

- Later processing might depend on the language of the text. For instance stop-word lists are language-dependent.
- Simplest method is based on **Letter Frequencies**

	Percentage of occurrence	
Letter	English	German
A	8.17	6.51
E	12.70	17.40
I	6.97	7.55
O	7.51	2.51
U	2.76	4.35

- Better—more elaborate—methods use statistics over more than one letter, e.g., statistics over two, three or even more consecutive letters (**N-Gram Frequencies**).

Sentence Detection

Naive approach

Every period and every “?” and “!” mark the end of a sentence

Problem

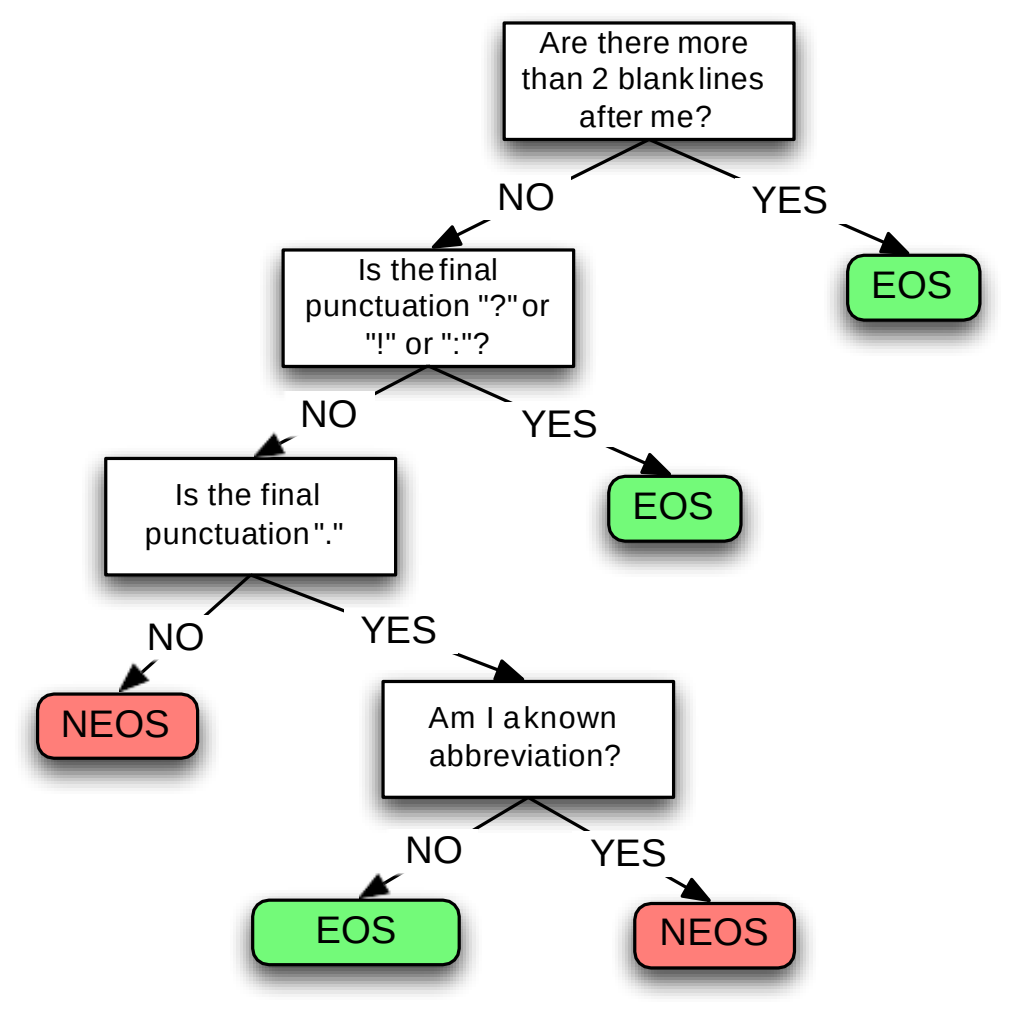
Periods also mark abbreviations, decimal points, email-addresses, e.g.,

- Dr. House is calling.
- The height is 0.14 inch.
- Mendeley Ldt. now is part of the Elsevier Corporation.

Solution

Build a binary classifier, deciding for each period whether it is the end of the sentence (EOS) or not (NEOS).

Sentence Detection



More Features:

- Is the next letter capitalized?
- Is the period surrounded by digits?
- Has the next word a high probability of occuring at the beginning of a sentence (e.g, Die, The)?

Stop Words

- Extremely common words that appear in nearly every text
- As stop words are so common, their occurrence does not characterize a text
- Just drop them

Stop word list Ignore word that are given on a list (black list), e.g., articles (*a*, *an*, *the*), conjunction (*and*, *or*, *but*, ...), pre- and postposition (*in*, *for*, *from*)

Problem Special names and phrases (*The Who*, *Let It Be*, ...)

Solution Make another list... (white list)

Regular Expressions

Regular Expressions

- Specific Letter: `[Aa]` matches either `A` or `a`.
- Range: `[A-Za-z0-9]` matches all alphabet in both cases and all numerals.
- Negation: `[^A-Z]` match all characters except capital letters.
- Disjunction: `red|green` matches either `red` or `green`
- Question mark (?): `colou?r` matches either `colour` or `color`.
- Asterisk(*): `wo*w`, zero or more occurrences of the previous character, matches `ww`, `wow`, `woow`, `wooooooooow`, etc.
- Plus(+): `wo+w`, one or more of the previous character, matches `wow`, `woow`, `wooow`, `wooooooooow`, etc.

Regular Expressions

- Dot(.): `end.`, matches exactly one character; matches `end.` or `end!` or `end?`, etc.
- Escape (\): escapes special characters. `end\` matches `end.`
- Carat (^): represents the start of a string ⇒ ^[A-Z] matches all strings that start with capital letter
- Word boundary (\b): `\b ha\b` matches `ha` but not `hahaha`.

Text Classification

Document-Term-Matrices

- Example (from [1]): 9 documents (consider title only) in two categories “human-computer-interaction” (h) and “graphs” (g)

h1 `Human machine interface for Lab ABC computer applications`

h2 `A survey of user opinion of computer system response time`

h3 `The EPS user interface management system`

h4 `System and human system engineering testing of EPS`

h5 `Relation of user-perceived response time to error measurement`

g1 `The generation of random, binary, unordered trees`

g2 `The intersection graph of paths in trees`

g3 `Graph minors IV: Widths of trees and well-quasi-ordering`

g4 `Graph minors: A survey`

Document-Term-Matrices

- Example (from [1]): 9 documents (consider title only) in two categories “human-computer-interaction” (h) and “graphs” (g)

	document	label
0	Human machine interface for Lab ABC computer applications	h
1	A survey of user opinion of computer system response time	h
2	The EPS user interface management system	h
3	System and human system engineering testing of EPS	h
4	Relation of user-perceived response time to error measurement	h
5	The generation of random, binary, unordered trees	g
6	The intersection graph of paths in trees	g
7	Graph minors IV: Widths of trees and well-quasi-ordering	g
8	Graph minors: A survey	g

Document-Term-Matrices

- Assume pre-processing (e.g., sentence splitting, stopword removal) has been done already

	document	label
0	human machin interfac lab abc comput applic	h
1	survey user opinion comput system respons time	h
2	ep user interfac manag system	h
3	system human system engin test ep	h
4	relat userperceiv respons time error measur	h
5	gener random binari unord tree	g
6	intersect graph path tree	g
7	graph minor iv width tree wellquasiord	g
8	graph minor survey	g

Document-Term-Matrices

	abc	applic	binari	comput	engin	ep	error	gener	graph	human	...	survey	system	test	time	tree	unord	user	userperceiv	wellquasiord	width
0	1	1	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	...	1	1	0	1	0	0	1	0	0	0
2	0	0	0	0	0	1	0	0	0	0	...	0	1	0	0	0	0	1	0	0	0
3	0	0	0	0	1	1	0	0	0	1	...	0	2	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	...	0	0	0	1	0	0	0	1	0	0
5	0	0	1	0	0	0	0	1	0	0	...	0	0	0	0	1	1	0	0	0	0
6	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	1	0	0	0	1	1
8	0	0	0	0	0	0	0	0	1	0	...	1	0	0	0	0	0	0	0	0	0

9 rows × 33 columns

Document-Term-Matrices

h1 Human machine interface for Lab ABC computer applications

Count occurrences of a particular token in the input

	h1	h2	h3	h4	h5	g1	g2	g3	g4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Drawbacks?

Each row (token) is a feature=> text vectorization

Document-Term-Matrices

- High-frequency n-grams (stop words?)
- Low-frequency n-grams (a typo?)
- Medium-frequency n-grams.

Term Weighting

- Assume that we have a document d_1 from the « h » class: « human system interaction: favoring human experience or system performance »
- If we represent this document as a vector, the words « human » and « system » will have both a count of 2.
- Is that fair? How do we fix it?
- Think about how does your belief about the class of a document change when you see the word « human » and how it changes when you see the word « system »!

Term Weighting

- TF (term frequency), i.e. word count, captures how important a word is for a document
- But some words are more frequent in general, TF-IDF weighting scheme aims to capture this (by downweighting those words)
  - tf_t^d – term frequency, number of times term *t* occurs in document *d*
  - *N* – total number of documents
  - df_t – document frequency, number of documents term *d* occurs in
  - idf_t – inverse document frequency

The TF-IDF measure for term *t* is defined as

$$\text{tf-idf}_t^d = \text{tf}_t^d \cdot \text{idf}_t \qquad \text{with } \text{idf}_t = log \frac{N}{df_t}$$

Bayes Classification

Bayes' Rule

Bayes's rule defines how conditional probabilities can be calculated from each other.

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (1)$$

Using notation from classification (C - class, D - data) it translates to

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)}$$

Bayes Classification

Bayes' Rule for Classification

$$P(C|D) = \frac{P(D|C) \cdot P(C)}{P(D)}$$

- Given a new document d1 « human computer interaction in OS», we want to predict its class, h or g?
- Every document is represented as a set of features, so

$$P(h|d1) = P(h|human, computer, interaction)$$

$$= P(human, computer, interaction|h) * \frac{P(h)}{P(human, computer, interaction)}$$

$$P(g|d1) = P(g|human, computer, interaction)$$

$$= P(human, computer, interaction|g) * \frac{P(g)}{P(human, computer, interaction)}$$

Bayes Classification

Bayes' Rule for Classification

- For d1, what we need to calculate is the following;

$$P(h|d1) = P(human, computer, interaction|h) * P(h)$$
$$P(g|d1) = P(human, computer, interaction|g) * P(g)$$

h1 Human machine interface for Lab ABC computer applications

h2 A survey of user opinion of computer system response time

h3 The EPS user interface management system

h4 System and human system engineering testing of EPS

h5 Relation of user-perceived response time to error measurement

g1 The generation of random, binary, unordered trees

g2 The intersection graph of paths in trees

g3 Graph minors IV: Widths of trees and well-quasi-ordering

g4 Graph minors: A survey

Bayes Classification

Bayes' Rule for Classification

- For d1, what we need to calculate is the following;

$$P(h|d1) = P(human, computer, interaction|h) * P(h)$$
$$P(g|d1) = P(human, computer, interaction|g) * P(g)$$

Two simplifying assumptions:

- Bag of Words: word order doesn't matter
- Conditional Independence assumption : features are independent given the class.

$$P(h|d1) = P(human\,|h) * P(computer|h) * P(interaction|h) * P(h)$$
$$P(g|d1) = P(human\,|g) * P(computer|g) * P(interaction|g) * P(g)$$

Bayes Classification

Bayes' Rule for Classification

$$P(h|d1) = P(human\,|h) * P(computer|h) * P(interaction|h) * P(h)$$
$$P(g|d1) = P(human\,|g) * P(computer|g) * P(interaction|g) * P(g)$$

$$P(h|d1) = P(human\,|h) * P(computer|h) * P(interaction|h) * P(h)$$

$$= \frac{2}{40} * \frac{2}{40} * \frac{0}{40} * \frac{5}{9} = 0$$

h1 Human machine interface for Lab ABC computer applications

h2 A survey of user opinion of computer system response time

h3 The EPS user interface management system

h4 System and human system engineering testing of EPS

h5 Relation of user-perceived response time to error measurement

40

g1 The generation of random, binary, unordered trees

g2 The intersection graph of paths in trees

g3 Graph minors IV: Widths of trees and well-quasi-ordering

g4 Graph minors: A survey

26

Bayes Classification

Bayes' Rule for Classification

$$P(h|d1) = P(human\,|h) * P(computer|h) * P(interaction|h) * P(h)$$

$$P(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in c} count(w, c) + vocab}$$

h1 Human machine interface for Lab ABC computer applications

h2 A survey of user opinion of computer system response time

h3 The EPS user interface management system

h4 System and human system engineering testing of EPS

h5 Relation of user-perceived response time to error measurement

40

g1 The generation of random, binary, unordered trees

g2 The intersection graph of paths in trees

g3 Graph minors IV: Widths of trees and well-quasi-ordering

g4 Graph minors: A survey

26

Bayes Classification

Bayes' Rule for Classification

$$P(h|d1) = P(human\,|h) * P(computer|h) * P(interaction|h) * P(h)$$

$$= \frac{2+1}{40+66} * \frac{2+1}{40+66} * \frac{0+1}{40+66} * \frac{5}{9} = 4.19*10^{-6}$$

$$P(g|d1) = P(human\,|g) * P(computer|g) * P(interaction|g) * P(g)$$

$$= \frac{0+1}{26+66} * \frac{0+1}{26+66} * \frac{0+1}{26+66} * \frac{4}{9} = 5.7 * 10^{-7}$$

h1 Human machine interface for Lab ABC computer applications

h2 A survey of user opinion of computer system response time

h3 The EPS user interface management system

h4 System and human system engineering testing of EPS

h5 Relation of user-perceived response time to error measurement

40

g1 The generation of random, binary, unordered trees

g2 The intersection graph of paths in trees

g3 Graph minors IV: Widths of trees and well-quasi-ordering

g4 Graph minors: A survey

26

Evaluation

ACCURACY

- Consider the following scenario:
  - P(cancer)=0.1 , and P(¬cancer)=0.9
  - You train a model on a dataset of 10.000 examples. 90% of which are labelled as “not cancer” and 10% as “cancer”.
  - Your model doesn’t learn a thing (sometimes they don’t)
  - You test your model on a test set of a 1000 examples, and it classifies everything as “not cancer”.

$$\boldsymbol{Accuracy = \frac{900}{900 + 100 + 0 + 0} = 90\%}$$

	ACTUAL: Cancer	ACTUAL: Not Cancer
PREDICTION: Cancer	0	0
PREDICTION: Not Cancer	100	900

PRECISION AND RECALL

- Consider another model that learned from the same data and was tested on the same test set. It delivered the results shown below;

$$\boldsymbol{Accuracy = \frac{50 + 800}{50 + 50 + 100 + 800} = 85\%}$$

		ACTUAL	
		Cancer	Not Cancer
PREDICTION	Cancer	50	50
	Not Cancer	100	800

- We can calculate the precision for class cancer as $\frac{TP}{TP+FP} = \frac{50}{50+50} = 50\%$
- We can calculate the recall for class cancer as $\frac{TP}{TP+FN} = \frac{50}{50+100} = 33\%$

F-MEASURE

F_1**-measure**: harmonic mean of precision and recall

$$F_1 = 2 \frac{\pi\rho}{\pi+\rho}$$

F_β -measure: generalisation allowing to emphasize either precision or recall (recall is β times more important)

$$F_\beta = (1 + \beta^2) \frac{\pi\rho}{\beta^2\pi + \rho}$$

Set beta to a value smaller than 1 to give precision more weight, set it to a value higher than 1 otherwise.

π	ρ	F_1	$F_{0.5}$	F_2
1	0.1	0.18	0.12	0.36
0.1	1	0.18	0.36	0.12
0	1	0.00	0.00	0.00
1	0	0.00	0.00	0.00
0.7	0.3	0.42	0.34	0.55
0.3	0.7	0.42	0.55	0.34
0.94	0.96	0.95	0.96	0.94

Further Topics

Named Entity Recognition

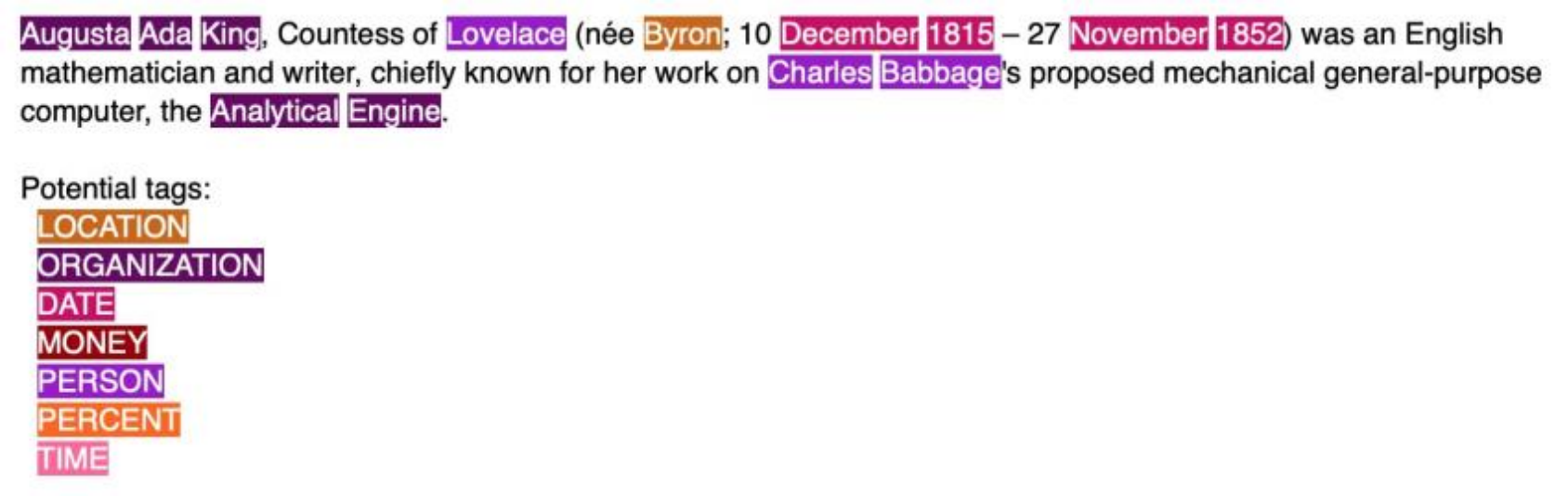


Figure 4: Named Entity Recognition with Stanford NER

`http://nlp.stanford.edu:8080/ner/process`

Useful for automatic tagging of news articles for example

Entity Disambiguation

Linking mentions to knowledge bases

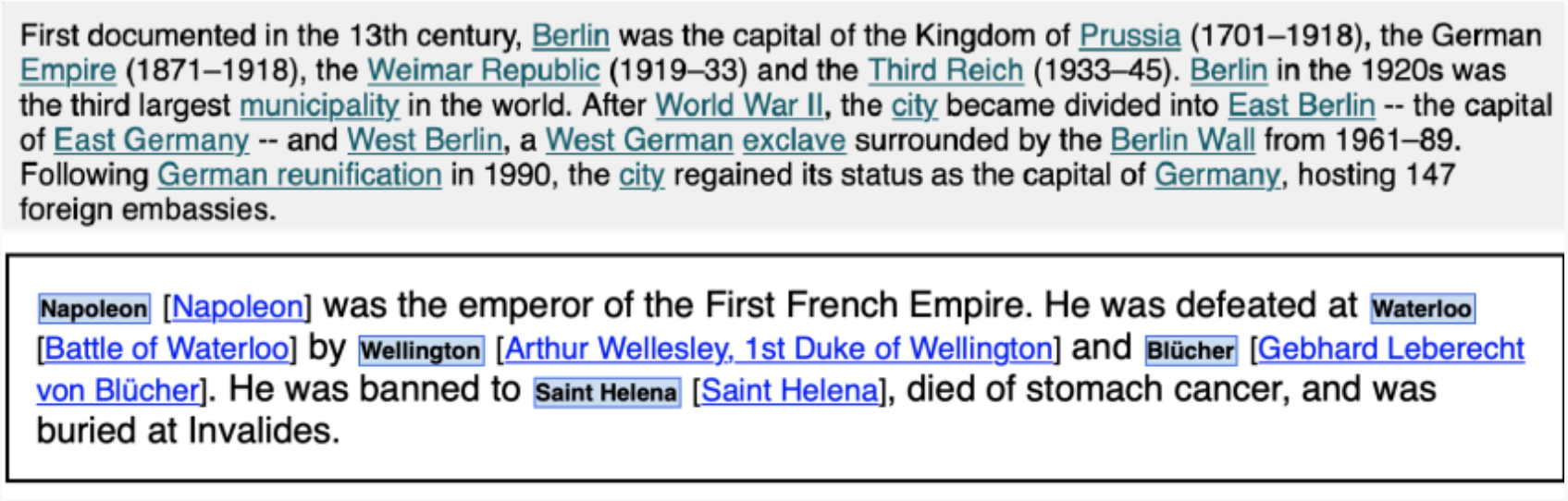


Figure 5: Disambiguation Examples by DBpedia Spotlight https://www.dbpedia-spotlight.org/demo/ and AIDA https://gate.d5.mpi-inf.mpg.de/webaida/

Relation Extraction

Look up the debate between Chomsky and Norvig on statiscal approaches: https://norvig.com/chomsky.html

Extraction of relations between objects

To give an example of Relation Extraction, if we are trying to find a birth date in:

> *"John von Neumann (December 28, 1903 – February 8, 1957) was a Hungarian and American pure and applied mathematician, physicist, inventor and polymath."*

then IEPY's task is to identify "`John von Neumann`" and "`December 28, 1903`" as the subject and object entities of the "`was born in`" relation.

Figure 6: Relation Extraction example from PyPi

https://pypi.org/project/iepy/

Useful for text summarization for example.
Rule-based approaches are useful as well. Look up Tesnière's Valency, which suggests verbs have a minimum number of "actants" to be complete. "Laugh" has only one, "give" has three. Monovalent or bivalent, etc.

Part-of-Speech Tagging

Annotate tokens with lexical function

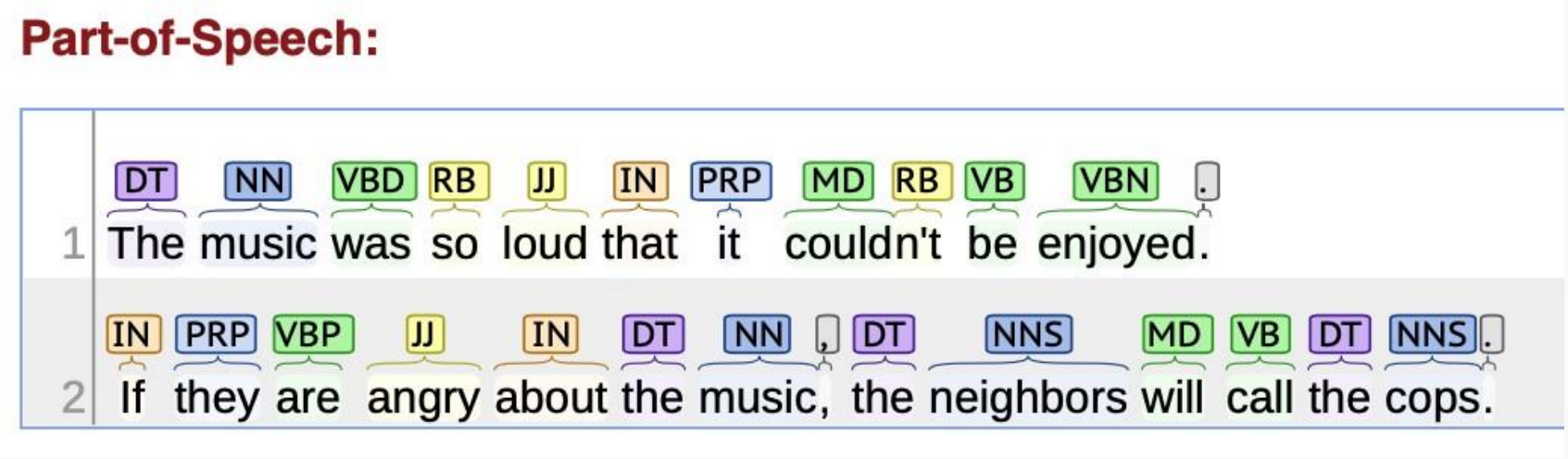


Figure 7: Part-of-Speech Tagging with Standford NLP

`http://nlp.stanford.edu:8080/corenlp/process`

Look at the tags at:
https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Or in the NLTK book, chapter 5 at:

http://www.nltk.org/book/ch05.html

Co-reference Resolution

Resolve words that reference another

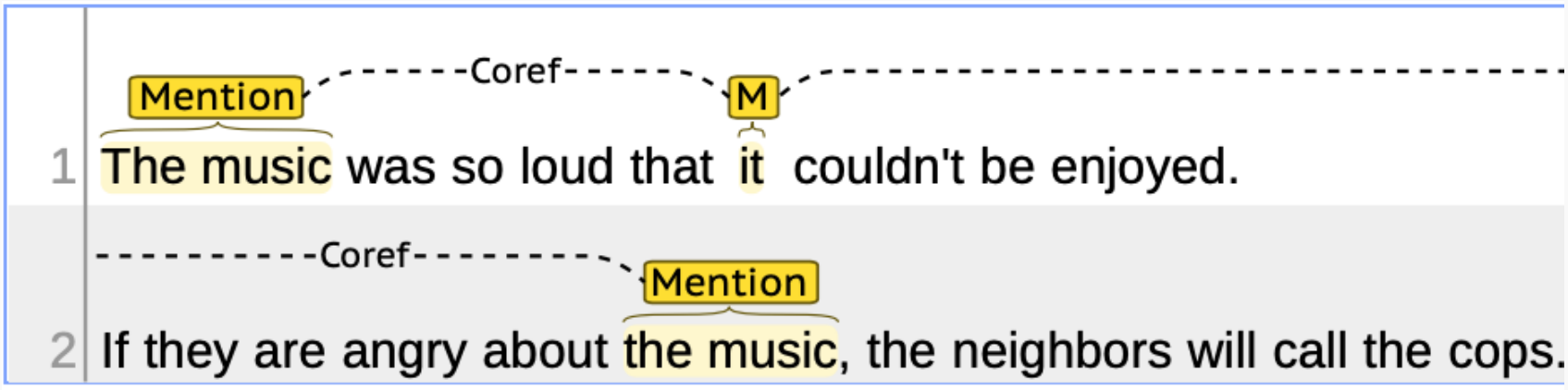
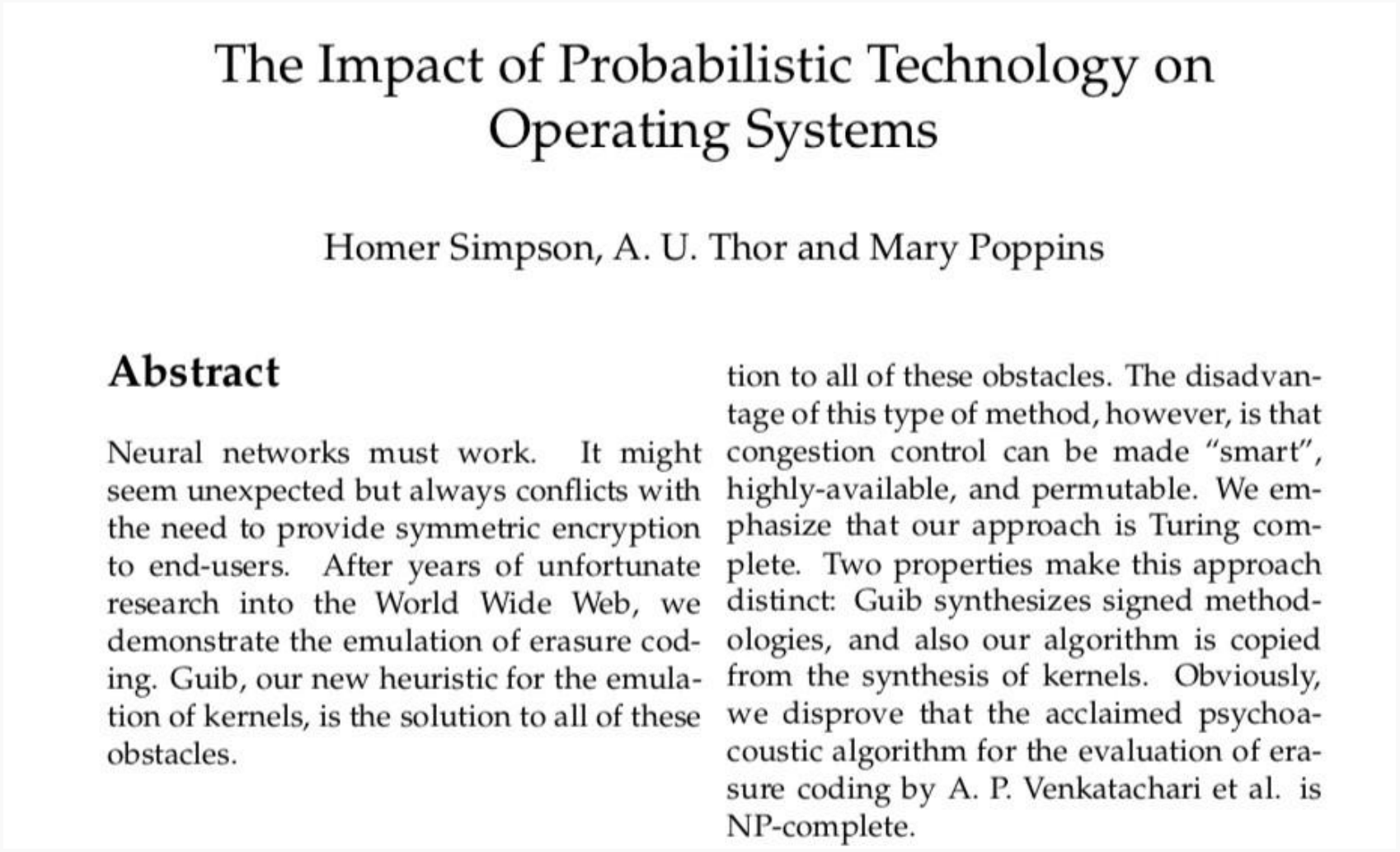


Figure 8: Coreference Resolution with Standford NLP

`http://nlp.stanford.edu:8080/corenlp/process`



The Impact of Probabilistic Technology on Operating Systems

Homer Simpson, A. U. Thor and Mary Poppins

Abstract

Neural networks must work. It might seem unexpected but always conflicts with the need to provide symmetric encryption to end-users. After years of unfortunate research into the World Wide Web, we demonstrate the emulation of erasure coding. Guib, our new heuristic for the emulation of kernels, is the solution to all of these obstacles.

tion to all of these obstacles. The disadvantage of this type of method, however, is that congestion control can be made "smart", highly-available, and permutable. We emphasize that our approach is Turing complete. Two properties make this approach distinct: Guib synthesizes signed methodologies, and also our algorithm is copied from the synthesis of kernels. Obviously, we disprove that the acclaimed psychoacoustic algorithm for the evaluation of erasure coding by A. P. Venkatachari et al. is NP-complete.

Figure 9: Automatically Generated Nonsense Paper

What if Netflix wants to make a new series to attract more Dutch subscribers?

NLP can’t answer the questions/subquestions alone. It needs to be integrated with other data science topics.

Tools and Software

Tools and Software

Example Notebooks (link on canvas)

- Regular Expressions, Text Preprocessing
- Named Entity Recognition, Text Classification

Tools

- Regular Expressions Online Test http://regexpal.com/
- Open Source NLP Toolkit SpaCy (python, with models for Dutch) https://spacy.io
- OpenNLP from the Apache Project (Java) https://opennlp.apache.org
- Gate language toolkit (Java) https://gate.ac.uk
- Stanford NLP http://nlp.stanford.edu:8080/corenlp/process

Acknowledgement

- Some slides (precision/recall) are adapted from the course on Machine Learning and Data Mining at the Bauhaus University Weimar (Prof. Dr. Benno Stein).
  - Overview: `webis.de/lecturenotes/overview/overview.html`
  - Slides: `webis.de/lecturenotes/slides/slides.html`
- Roman Kern, Technical University Graz (some preprocessing slides)
- Regular Expression Examples from a slide set from Mena Habib, University of Twente

Further Readings

- Introduction to Information Retrieval
  Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schu¨tze. *Introduction to Information Retrieval*.New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521865719, 9780521865715. URL: https://nlp.stanford.edu/IR-book/

References

1 Scott Deerwester et al.“Indexing by latent semantic analysis”.In: *Journal of the Society for Information Science* 41.6 (1990), pp. 391–407. URL: http://lsi.research.telcordia.com/lsi/LSIpapers.html.

2 Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*.New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521865719, 9780521865715. URL: https://nlp.stanford.edu/IR-book/.