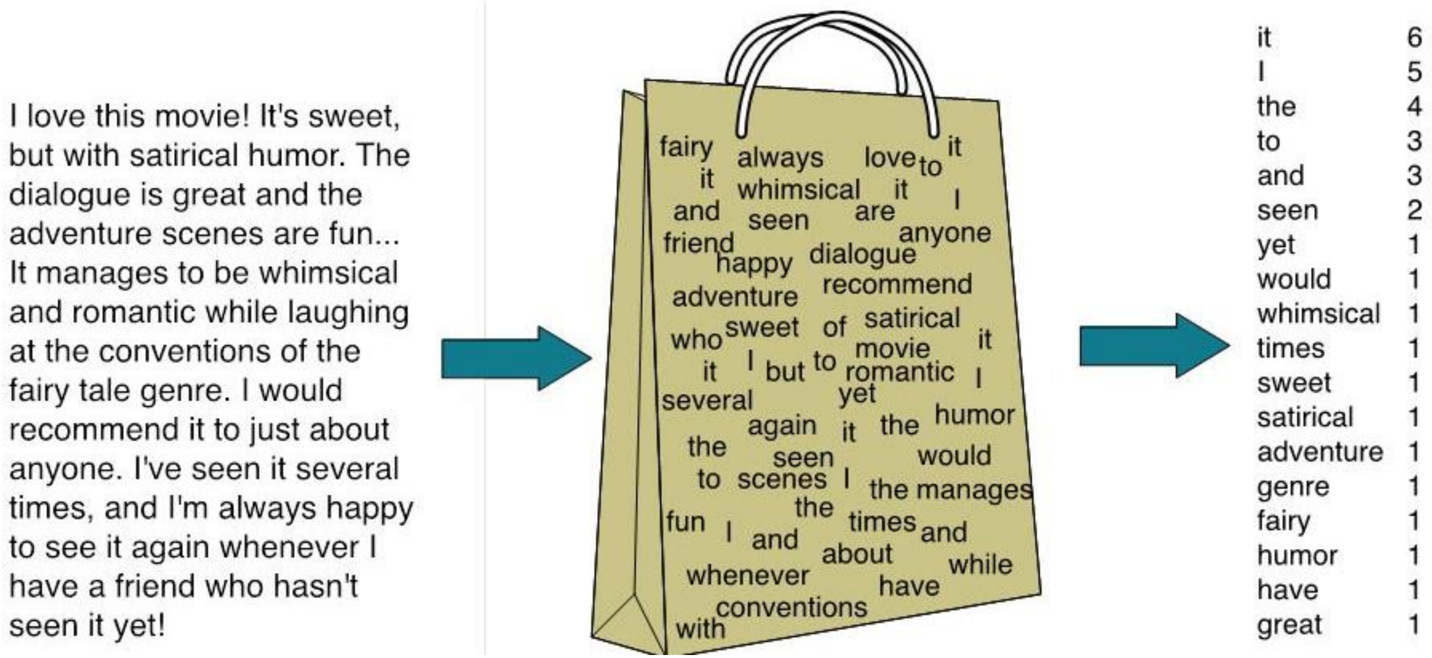


I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1

Lecture 4: Document Representation I Count-based Representations

Pilsung Kang

School of Industrial Management Engineering

Korea University

AGENDA

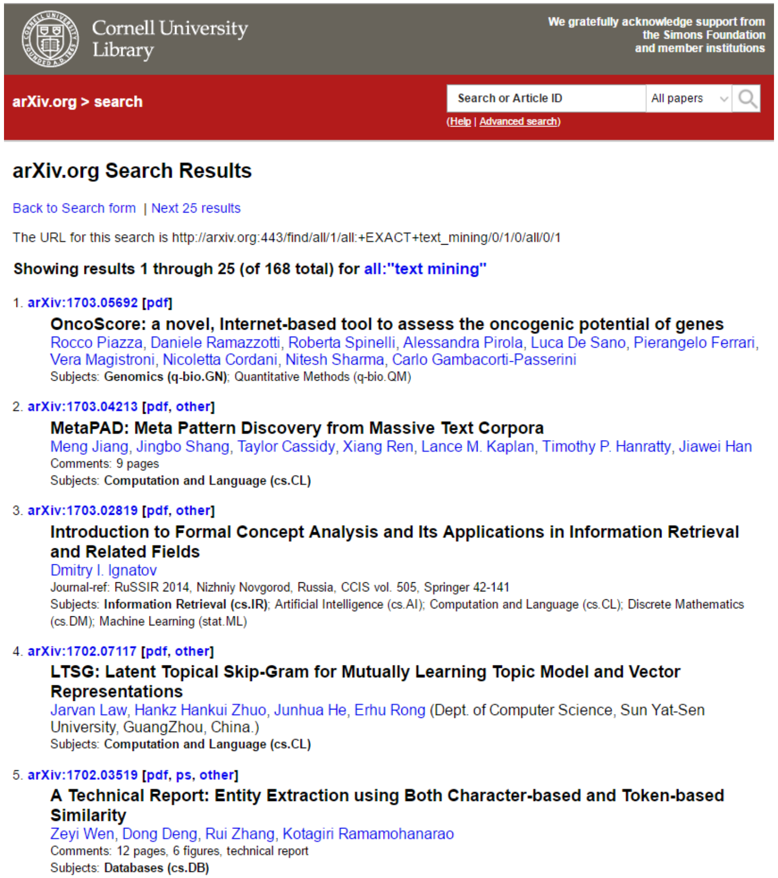
01 Bag of Words

02 Word Weighting

03 N-Grams

What We Have Done So Far…

Collecting Text Data


Cornell University Library
We gratefully acknowledge support from the Simons Foundation and member institutions
arXiv.org > search
Search or Article ID
All papers
(Help | Advanced search)

arXiv.org Search Results

Back to Search form | Next 25 results

The URL for this search is http://arxiv.org:443/find/all/1/all:+EXACT+text_mining/0/1/0/all/0/1

Showing results 1 through 25 (of 168 total) for all:"text mining"

1. arXiv:1703.05692 [pdf]
   OncoScore: a novel, Internet-based tool to assess the oncogenic potential of genes
   Rocco Piazza, Daniele Ramazzotti, Roberta Spinelli, Alessandra Pirola, Luca De Sano, Pierangelo Ferrari, Vera Magistroni, Nicoletta Cordani, Nitesh Sharma, Carlo Gambacorti-Passerini
   Subjects: **Genomics (q-bio.GN)**; Quantitative Methods (q-bio.QM)

2. arXiv:1703.04213 [pdf, other]
   MetaPAD: Meta Pattern Discovery from Massive Text Corpora
   Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M. Kaplan, Timothy P. Hanratty, Jiawei Han
   Comments: 9 pages
   Subjects: **Computation and Language (cs.CL)**

3. arXiv:1703.02819 [pdf, other]
   Introduction to Formal Concept Analysis and Its Applications in Information Retrieval and Related Fields
   Dmitry I. Ignatov
   Journal-ref: RuSSIR 2014, Nizhniy Novgorod, Russia, CCIS vol. 505, Springer 42-141
   Subjects: **Information Retrieval (cs.IR)**; Artificial Intelligence (cs.AI); Computation and Language (cs.CL); Discrete Mathematics (cs.DM); Machine Learning (stat.ML)

4. arXiv:1702.07117 [pdf, other]
   LTSG: Latent Topical Skip-Gram for Mutually Learning Topic Model and Vector Representations
   Jarvan Law, Hankz Hankui Zhuo, Junhua He, Erhu Rong (Dept. of Computer Science, Sun Yat-Sen University, GuangZhou, China.)
   Subjects: **Computation and Language (cs.CL)**

5. arXiv:1702.03519 [pdf, ps, other]
   A Technical Report: Entity Extraction using Both Character-based and Token-based Similarity
   Zeyi Wen, Dong Deng, Rui Zhang, Kotagiri Ramamohanarao
   Comments: 12 pages, 6 figures, technical report
   Subjects: **Databases (cs.DB)**

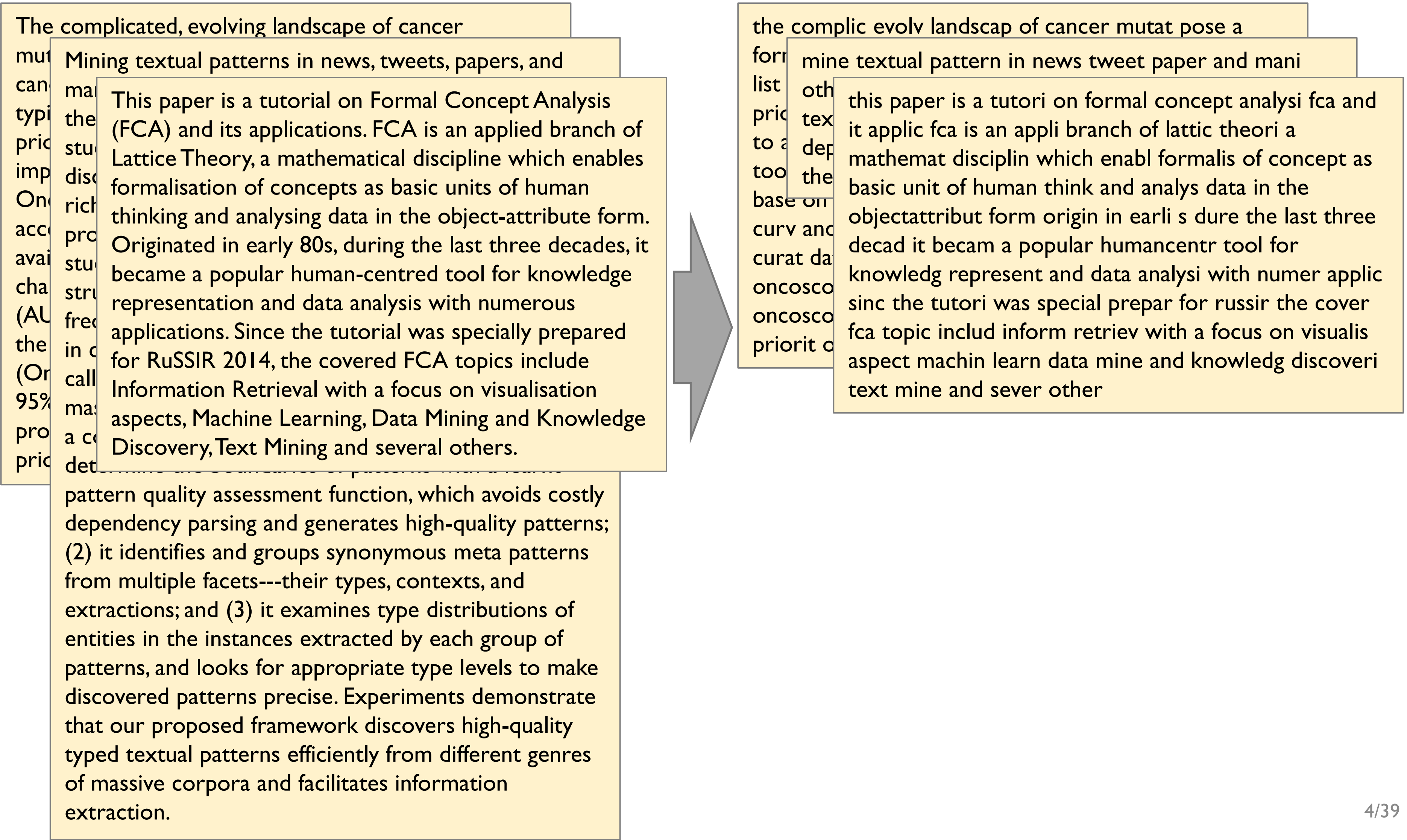
The complicated, evolving landscape of cancer
mut
can
typi
pric
imp
On
acc
avai
cha
(AU
the
(Or
95%
pro
pric

Mining textual patterns in news, tweets, papers, and
mar
the
stu
disc
rich
pro
stu
stru
freq
in c
call
mas
a co
det
pattern quality assessment function, which avoids costly dependency parsing and generates high-quality patterns; (2) it identifies and groups synonymous meta patterns from multiple facets---their types, contexts, and extractions; and (3) it examines type distributions of entities in the instances extracted by each group of patterns, and looks for appropriate type levels to make discovered patterns precise. Experiments demonstrate that our proposed framework discovers high-quality typed textual patterns efficiently from different genres of massive corpora and facilitates information extraction.

This paper is a tutorial on Formal Concept Analysis (FCA) and its applications. FCA is an applied branch of Lattice Theory, a mathematical discipline which enables formalisation of concepts as basic units of human thinking and analysing data in the object-attribute form. Originated in early 80s, during the last three decades, it became a popular human-centred tool for knowledge representation and data analysis with numerous applications. Since the tutorial was specially prepared for RuSSIR 2014, the covered FCA topics include Information Retrieval with a focus on visualisation aspects, Machine Learning, Data Mining and Knowledge Discovery, Text Mining and several others.

What We Have Done So Far…

Preprocessing with some NLP techniques



The complicated, evolving landscape of cancer
mut
can
typi
pric
imp
On
acc
avai
cha
(AU
the
(Or
95%
pro
pric

Mining textual patterns in news, tweets, papers, and
ma
the
stu
dis
rich
pro
stu
stru
freq
in c
call
mas
a co
det

This paper is a tutorial on Formal Concept Analysis (FCA) and its applications. FCA is an applied branch of Lattice Theory, a mathematical discipline which enables formalisation of concepts as basic units of human thinking and analysing data in the object-attribute form. Originated in early 80s, during the last three decades, it became a popular human-centred tool for knowledge representation and data analysis with numerous applications. Since the tutorial was specially prepared for RuSSIR 2014, the covered FCA topics include Information Retrieval with a focus on visualisation aspects, Machine Learning, Data Mining and Knowledge Discovery, Text Mining and several others.

pattern quality assessment function, which avoids costly dependency parsing and generates high-quality patterns; (2) it identifies and groups synonymous meta patterns from multiple facets---their types, contexts, and extractions; and (3) it examines type distributions of entities in the instances extracted by each group of patterns, and looks for appropriate type levels to make discovered patterns precise. Experiments demonstrate that our proposed framework discovers high-quality typed textual patterns efficiently from different genres of massive corpora and facilitates information extraction.

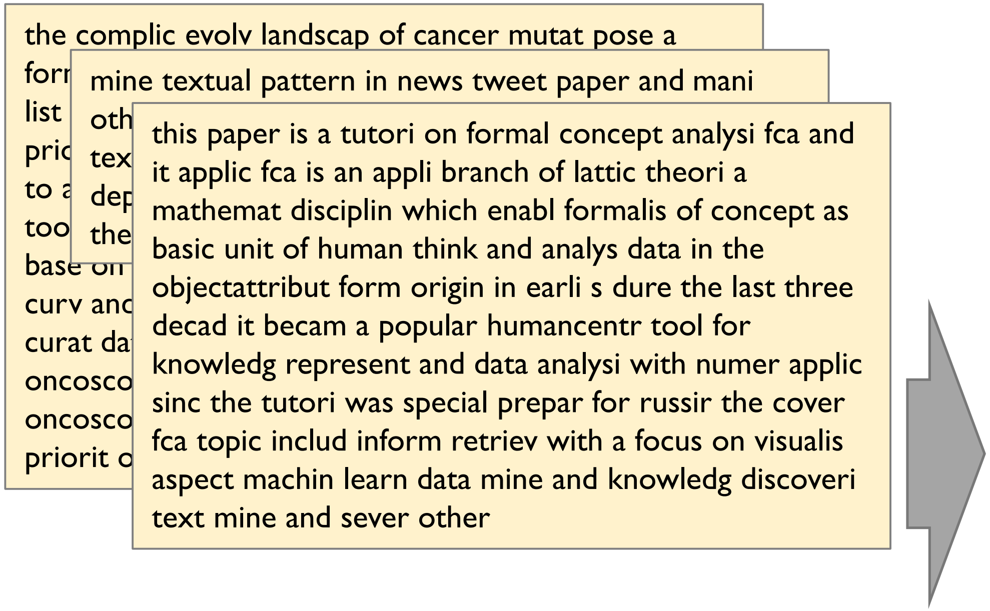
the complic evolv landscap of cancer mutat pose a
forr
list
pric
to a
too
base on
curv and
curat da
oncosco
oncosco
priorit o

mine textual pattern in news tweet paper and mani
oth
tex
dep
the

this paper is a tutori on formal concept analysi fca and it applic fca is an appli branch of lattic theori a mathemat disciplin which enabl formalis of concept as basic unit of human think and analys data in the objectattribut form origin in earli s dure the last three decad it becam a popular humancentr tool for knowledg represent and data analysi with numer applic sinc the tutori was special prepar for russir the cover fca topic includ inform retriev with a focus on visualis aspect machin learn data mine and knowledg discoveri text mine and sever other

What We Will Do…

Transform unstructured data into structured data



the complic evolv landscap of cancer mutat pose a
forr
list
pric
to a
too
base on
curv and
curat da
oncosco
oncosco
priorit o

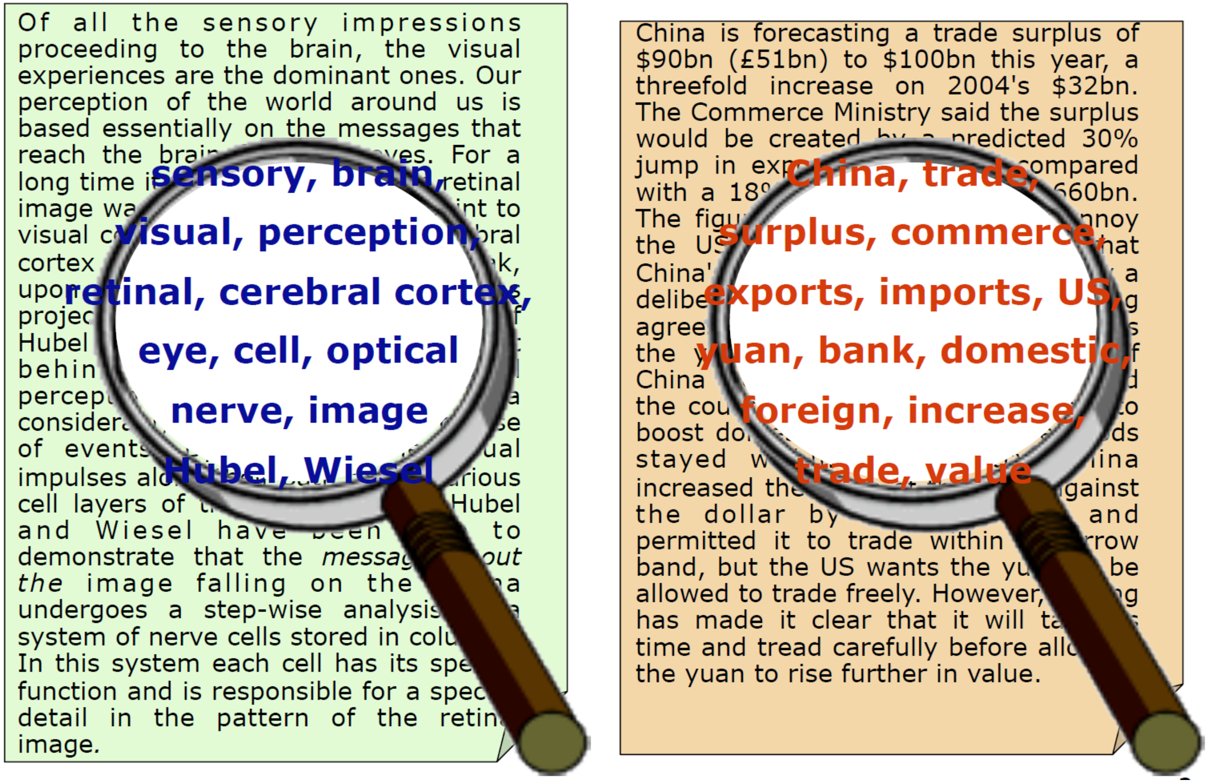
mine textual pattern in news tweet paper and mani
oth
tex
dep
the

this paper is a tutori on formal concept analysi fca and it applic fca is an appli branch of lattic theori a mathemat disciplin which enabl formalis of concept as basic unit of human think and analys data in the objectattribut form origin in earli s dure the last three decad it becam a popular humancentr tool for knowledg represent and data analysi with numer applic sinc the tutori was special prepar for russir the cover fca topic includ inform retriev with a focus on visualis aspect machin learn data mine and knowledg discoveri text mine and sever other

	Var 1	Var 2	…	…	Var P
Doc 1					
Doc 2					
Doc 3					
…					
…					
…					
Doc D					

Bag of Words: Motivation

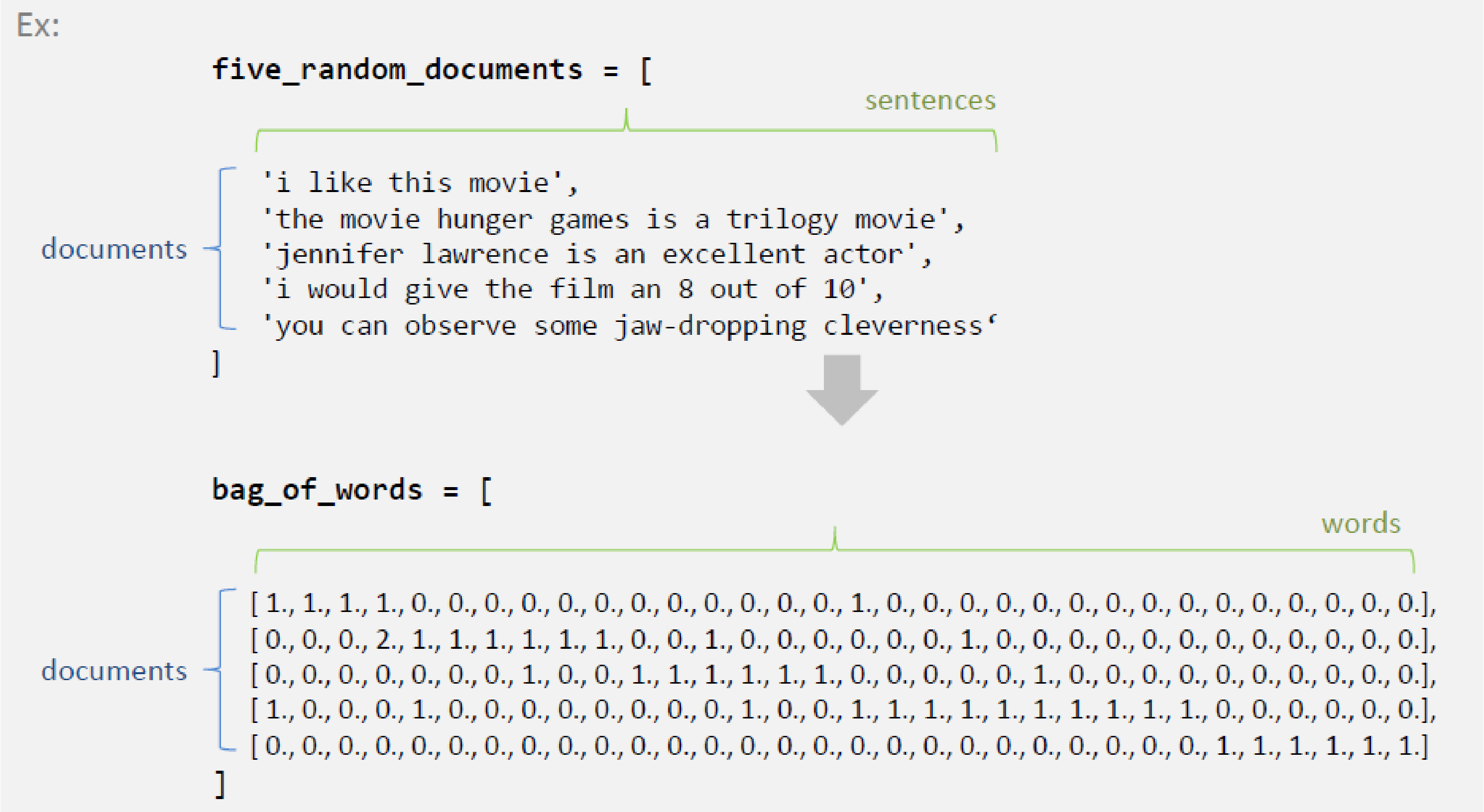
- Document Representation
  - ✓ How to represent a document in a structured way?
  - ✓ How to convert a unstructured text into a vector/matrix form to apply machine learning algorithms based on a vector space?



Bag of Words: Idea

박은정 (2016)

- Bag-of-words
  - ✓ A simplified representation method for documents where a text is represented in a vector of an unordered collection of words
  - ✓ Consider words as atomic symbols, represented in the discrete space



Ex:

```
five_random_documents = [
    'i like this movie',
    'the movie hunger games is a trilogy movie',
    'jennifer lawrence is an excellent actor',
    'i would give the film an 8 out of 10',
    'you can observe some jaw-dropping cleverness‘
]

bag_of_words = [
    [ 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
    [ 0., 0., 0., 2., 1., 1., 1., 1., 1., 1., 0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
    [ 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 1., 1., 1., 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., 0., 0., 1., 0., 0., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0.],
    [ 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1.]
]
```

Bag of Words: Idea

- Bag-of-words: Term-Document Matrix
  - ✓ Simplifying representation method for documents where a text is represented in a vector of an unordered collection of words

S1: John likes to watch movies. Mary likes too.

S2: John also likes to watch football game.

Binary representation

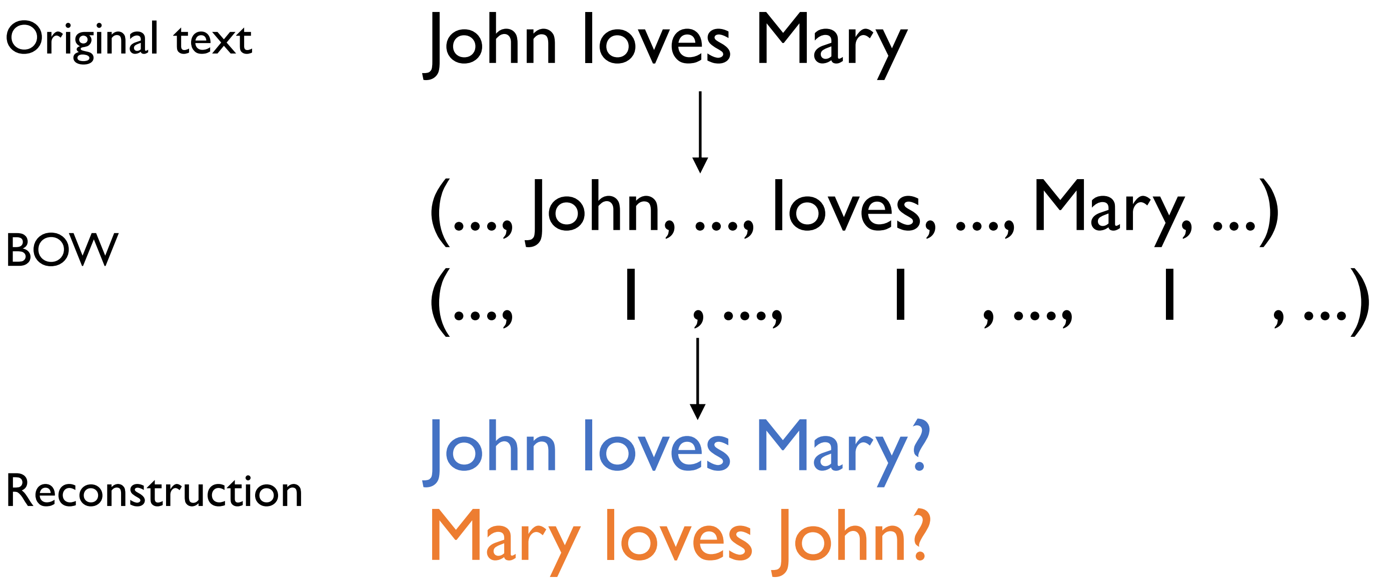
Word	S1	S2
John	1	1
Likes	1	1
To	1	1
Watch	1	1
Movies	1	0
Also	0	1
Football	0	1
Games	0	1
Mary	1	0
too	1	0

Frequency representation

Word	S1	S2
John	1	1
Likes	2	1
To	1	1
Watch	1	1
Movies	1	0
Also	0	1
Football	0	1
Games	0	1
Mary	1	0
too	1	0

Bag of Words: Idea

- Bag of words Representation in a Vector Space
  - ✓ The contents can be inferred from the frequency of words
  - ✓ Vector representation does not consider the ordering of words in a document
    - ▪ Visual words = independent features
    - ▪ John is quicker than Mary = Mary is quicker than John in BOW representation
  - ✓ We cannot reconstruct the original text based on the term-document matrix



Original text

John loves Mary

BOW

(..., John, ..., loves, ..., Mary, ...)

(..., 1 , ..., 1 , ..., 1 , ...)

Reconstruction

John loves Mary?

Mary loves John?

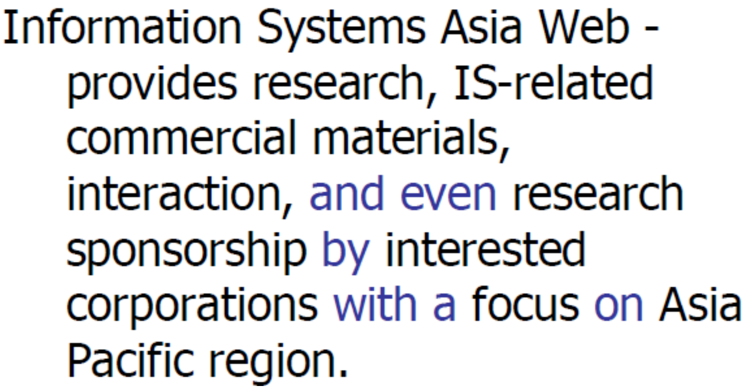
Text Preprocessing

- Remove unnecessary information
  - ✓ They vs. they: different words in many systems
    - ▪ lower case is commonly used
  - ✓ Punctuation
    - ▪ Punctuations do not contain significant information → Remove them!
  - ✓ Numbers
    - ▪ Numbers are not critical in some domains but critical in other domains
    - ▪ Removing numbers should be carefully determined based on the domain for which a collection of text is about to be analyzed

Stop Words

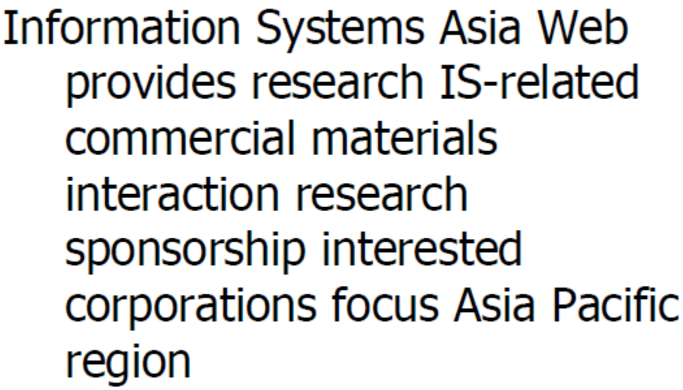
- What are stop words?
  - ✓ Words that do not carry any information
    - Mainly functional role
    - Usually remove them to help the machine learning algorithms to perform better
  - ✓ Natural language dependent
    - English: a, about, above, across, after, again, against, all, also, etc.
    - 한국어: …습니다, …로서(써), …를 등

[Original text]



Information Systems Asia Web - provides research, IS-related commercial materials, interaction, and even research sponsorship by interested corporations with a focus on Asia Pacific region.

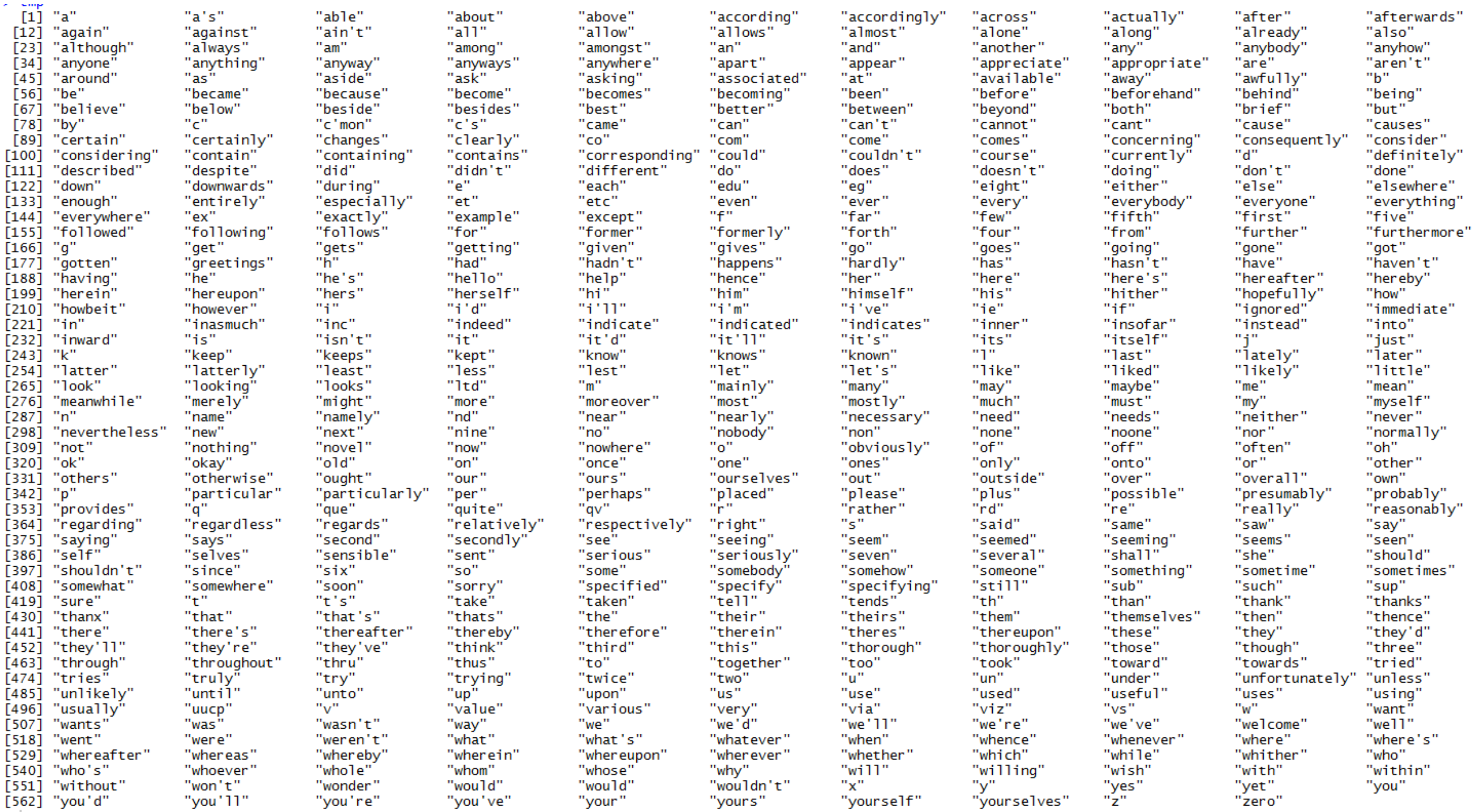
[After removing stop words]



Information Systems Asia Web provides research IS-related commercial materials interaction research sponsorship interested corporations focus Asia Pacific region

Stop Words

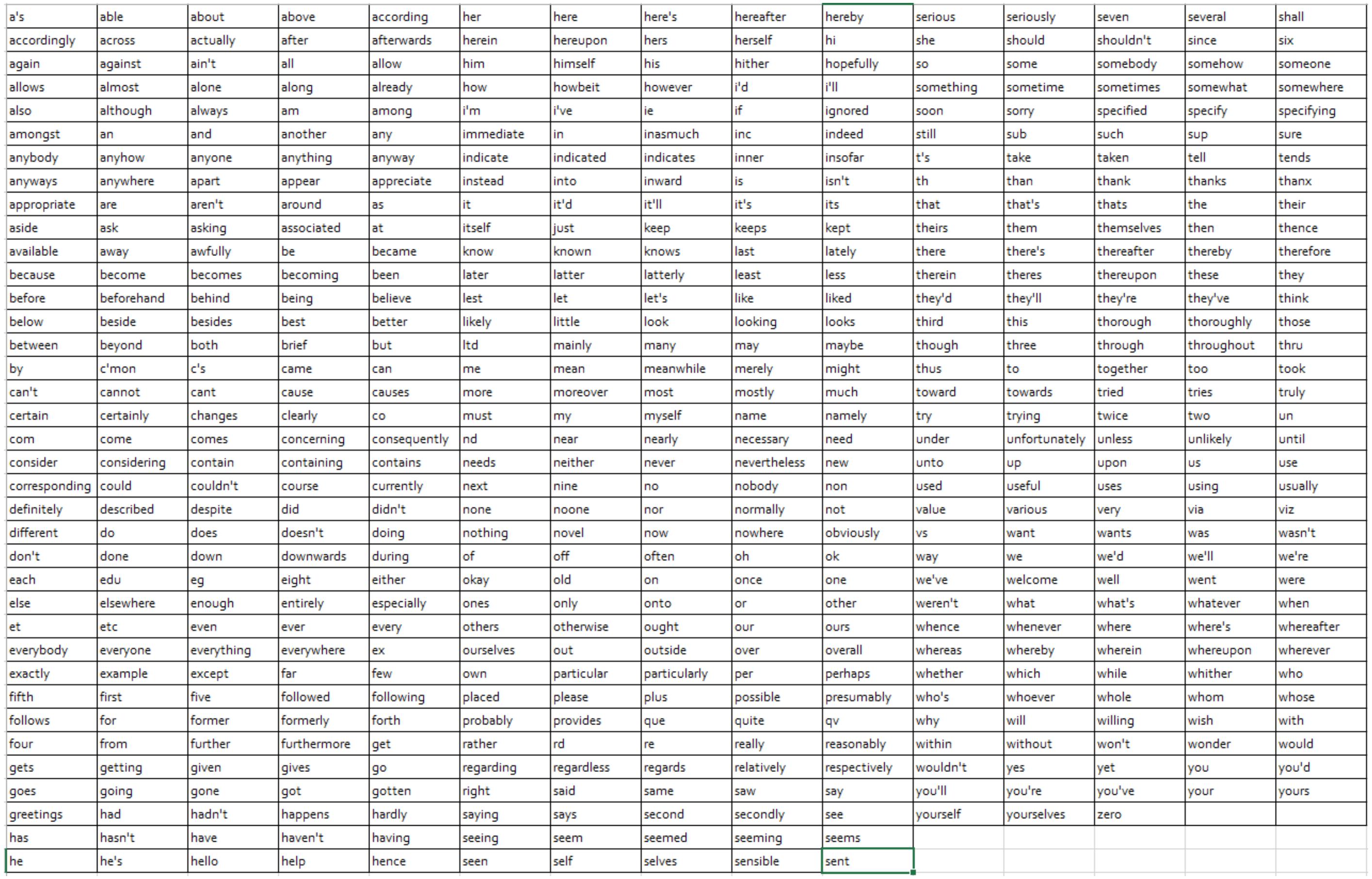
- Example 1: SMART stop words list
  - ✓ SMART: **S**ystem for the **M**echanical **A**nalysis and **R**etrieval of **T**ext
    - ▪ A total of 571 stop words



```
  [1] "a"             "a's"           "able"          "about"         "above"         "according"     "accordingly"   "across"        "actually"      "after"         "afterwards"
 [12] "again"         "against"       "ain't"         "all"           "allow"         "allows"        "almost"        "alone"         "along"         "already"       "also"
 [23] "although"      "always"        "am"            "among"         "amongst"       "an"            "and"           "another"       "any"           "anybody"       "anyhow"
 [34] "anyone"        "anything"      "anyway"        "anyways"       "anywhere"      "apart"         "appear"        "appreciate"    "appropriate"   "are"           "aren't"
 [45] "around"        "as"            "aside"         "ask"           "asking"        "associated"    "at"            "available"     "away"          "awfully"       "b"
 [56] "be"            "became"        "because"       "become"        "becomes"       "becoming"      "been"          "before"        "beforehand"    "behind"        "being"
 [67] "believe"       "below"         "beside"        "besides"       "best"          "better"        "between"       "beyond"        "both"          "brief"         "but"
 [78] "by"            "c"             "c'mon"         "c's"           "came"          "can"           "can't"         "cannot"        "cant"          "cause"         "causes"
 [89] "certain"       "certainly"     "changes"       "clearly"       "co"            "com"           "come"          "comes"         "concerning"    "consequently"  "consider"
[100] "considering"   "contain"       "containing"    "contains"      "corresponding" "could"         "couldn't"      "course"        "currently"     "d"             "definitely"
[111] "described"     "despite"       "did"           "didn't"        "different"     "do"            "does"          "doesn't"       "doing"         "don't"         "done"
[122] "down"          "downwards"     "during"        "e"             "each"          "edu"           "eg"            "eight"         "either"        "else"          "elsewhere"
[133] "enough"        "entirely"      "especially"    "et"            "etc"           "even"          "ever"          "every"         "everybody"     "everyone"      "everything"
[144] "everywhere"    "ex"            "exactly"       "example"       "except"        "f"             "far"           "few"           "fifth"         "first"         "five"
[155] "followed"      "following"     "follows"       "for"           "former"        "formerly"      "forth"         "four"          "from"          "further"       "furthermore"
[166] "g"             "get"           "gets"          "getting"       "given"         "gives"         "go"            "goes"          "going"         "gone"          "got"
[177] "gotten"        "greetings"     "h"             "had"           "hadn't"        "happens"       "hardly"        "has"           "hasn't"        "have"          "haven't"
[188] "having"        "he"            "he's"          "hello"         "help"          "hence"         "her"           "here"          "here's"        "hereafter"     "hereby"
[199] "herein"        "hereupon"      "hers"          "herself"       "hi"            "him"           "himself"       "his"           "hither"        "hopefully"     "how"
[210] "howbeit"       "however"       "i"             "i'd"           "i'll"          "i'm"           "i've"          "ie"            "if"            "ignored"       "immediate"
[221] "in"            "inasmuch"      "inc"           "indeed"        "indicate"      "indicated"     "indicates"     "inner"         "insofar"       "instead"       "into"
[232] "inward"        "is"            "isn't"         "it"            "it'd"          "it'll"         "it's"          "its"           "itself"        "j"             "just"
[243] "k"             "keep"          "keeps"         "kept"          "know"          "knows"         "known"         "l"             "last"          "lately"        "later"
[254] "latter"        "latterly"      "least"         "less"          "lest"          "let"           "let's"         "like"          "liked"         "likely"        "little"
[265] "look"          "looking"       "looks"         "ltd"           "m"             "mainly"        "many"          "may"           "maybe"         "me"            "mean"
[276] "meanwhile"     "merely"        "might"         "more"          "moreover"      "most"          "mostly"        "much"          "must"          "my"            "myself"
[287] "n"             "name"          "namely"        "nd"            "near"          "nearly"        "necessary"     "need"          "needs"         "neither"       "never"
[298] "nevertheless"  "new"           "next"          "nine"          "no"            "nobody"        "non"           "none"          "noone"         "nor"           "normally"
[309] "not"           "nothing"       "novel"         "now"           "nowhere"       "o"             "obviously"     "of"            "off"           "often"         "oh"
[320] "ok"            "okay"          "old"           "on"            "once"          "one"           "ones"          "only"          "onto"          "or"            "other"
[331] "others"        "otherwise"     "ought"         "our"           "ours"          "ourselves"     "out"           "outside"       "over"          "overall"       "own"
[342] "p"             "particular"    "particularly"  "per"           "perhaps"       "placed"        "please"        "plus"          "possible"      "presumably"    "probably"
[353] "provides"      "q"             "que"           "quite"         "qv"            "r"             "rather"        "rd"            "re"            "really"        "reasonably"
[364] "regarding"     "regardless"    "regards"       "relatively"    "respectively"  "right"         "s"             "said"          "same"          "saw"           "say"
[375] "saying"        "says"          "second"        "secondly"      "see"           "seeing"        "seem"          "seemed"        "seeming"       "seems"         "seen"
[386] "self"          "selves"        "sensible"      "sent"          "serious"       "seriously"     "seven"         "several"       "shall"         "she"           "should"
[397] "shouldn't"     "since"         "six"           "so"            "some"          "somebody"      "somehow"       "someone"       "something"     "sometime"      "sometimes"
[408] "somewhat"      "somewhere"     "soon"          "sorry"         "specified"     "specify"       "specifying"    "still"         "sub"           "such"          "sup"
[419] "sure"          "t"             "t's"           "take"          "taken"         "tell"          "tends"         "th"            "than"          "thank"         "thanks"
[430] "thanx"         "that"          "that's"        "thats"         "the"           "their"         "theirs"        "them"          "themselves"    "then"          "thence"
[441] "there"         "there's"       "thereafter"    "thereby"       "therefore"     "therein"       "theres"        "thereupon"     "these"         "they"          "they'd"
[452] "they'll"       "they're"       "they've"       "think"         "third"         "this"          "thorough"      "thoroughly"    "those"         "though"        "three"
[463] "through"       "throughout"    "thru"          "thus"          "to"            "together"      "too"           "took"          "toward"        "towards"       "tried"
[474] "tries"         "truly"         "try"           "trying"        "twice"         "two"           "u"             "un"            "under"         "unfortunately" "unless"
[485] "unlikely"      "until"         "unto"          "up"            "upon"          "us"            "use"           "used"          "useful"        "uses"          "using"
[496] "usually"       "uucp"          "v"             "value"         "various"       "very"          "via"           "viz"           "vs"            "w"             "want"
[507] "wants"         "was"           "wasn't"        "way"           "we"            "we'd"          "we'll"         "we're"         "we've"         "welcome"       "well"
[518] "went"          "were"          "weren't"       "what"          "what's"        "whatever"      "when"          "whence"        "whenever"      "where"         "where's"
[529] "whereafter"    "whereas"       "whereby"       "wherein"       "whereupon"     "wherever"      "whether"       "which"         "while"         "whither"       "who"
[540] "who's"         "whoever"       "whole"         "whom"          "whose"         "why"           "will"          "willing"       "wish"          "with"          "within"
[551] "without"       "won't"         "wonder"        "would"         "would"         "wouldn't"      "x"             "y"             "yes"           "yet"           "you"
[562] "you'd"         "you'll"        "you're"        "you've"        "your"          "yours"         "yourself"      "yourselves"    "z"             "zero"
```

Stop Words

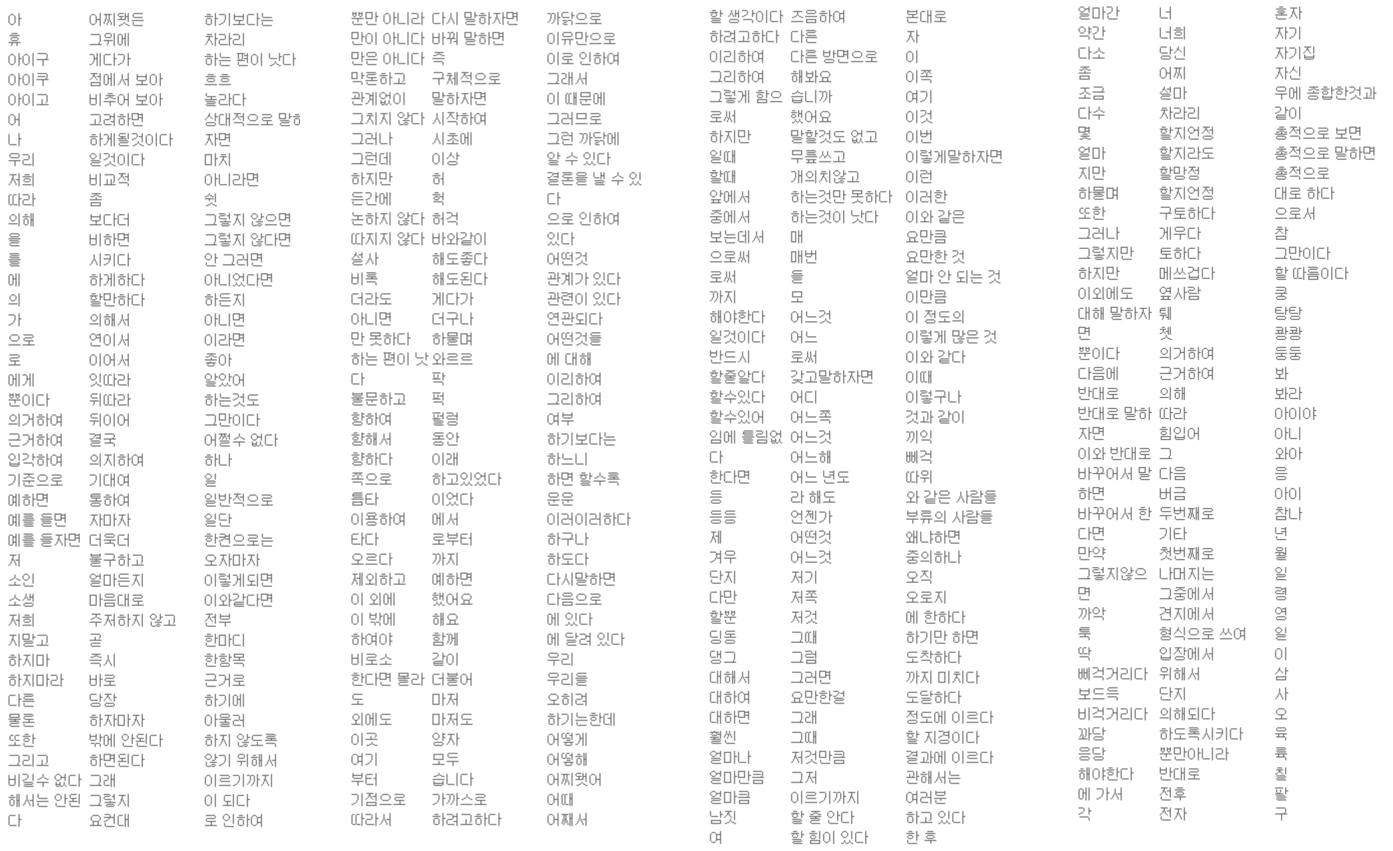
- Example 2: MySQL Stop words list
  - ✓ http://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html
    - A total of 543 stop words



a's	able	about	above	according	her	here	here's	hereafter	hereby	serious	seriously	seven	several	shall
accordingly	across	actually	after	afterwards	herein	hereupon	hers	herself	hi	she	should	shouldn't	since	six
again	against	ain't	all	allow	him	himself	his	hither	hopefully	so	some	somebody	somehow	someone
allows	almost	alone	along	already	how	howbeit	however	i'd	i'll	something	sometime	sometimes	somewhat	somewhere
also	although	always	am	among	i'm	i've	ie	if	ignored	soon	sorry	specified	specify	specifying
amongst	an	and	another	any	immediate	in	inasmuch	inc	indeed	still	sub	such	sup	sure
anybody	anyhow	anyone	anything	anyway	indicate	indicated	indicates	inner	insofar	t's	take	taken	tell	tends
anyways	anywhere	apart	appear	appreciate	instead	into	inward	is	isn't	th	than	thank	thanks	thanx
appropriate	are	aren't	around	as	it	it'd	it'll	it's	its	that	that's	thats	the	their
aside	ask	asking	associated	at	itself	just	keep	keeps	kept	theirs	them	themselves	then	thence
available	away	awfully	be	became	know	known	knows	last	lately	there	there's	thereafter	thereby	therefore
because	become	becomes	becoming	been	later	latter	latterly	least	less	therein	theres	thereupon	these	they
before	beforehand	behind	being	believe	lest	let	let's	like	liked	they'd	they'll	they're	they've	think
below	beside	besides	best	better	likely	little	look	looking	looks	third	this	thorough	thoroughly	those
between	beyond	both	brief	but	ltd	mainly	many	may	maybe	though	three	through	throughout	thru
by	c'mon	c's	came	can	me	mean	meanwhile	merely	might	thus	to	together	too	took
can't	cannot	cant	cause	causes	more	moreover	most	mostly	much	toward	towards	tried	tries	truly
certain	certainly	changes	clearly	co	must	my	myself	name	namely	try	trying	twice	two	un
com	come	comes	concerning	consequently	nd	near	nearly	necessary	need	under	unfortunately	unless	unlikely	until
consider	considering	contain	containing	contains	needs	neither	never	nevertheless	new	unto	up	upon	us	use
corresponding	could	couldn't	course	currently	next	nine	no	nobody	non	used	useful	uses	using	usually
definitely	described	despite	did	didn't	none	noone	nor	normally	not	value	various	very	via	viz
different	do	does	doesn't	doing	nothing	novel	now	nowhere	obviously	vs	want	wants	was	wasn't
don't	done	down	downwards	during	of	off	often	oh	ok	way	we	we'd	we'll	we're
each	edu	eg	eight	either	okay	old	on	once	one	we've	welcome	well	went	were
else	elsewhere	enough	entirely	especially	ones	only	onto	or	other	weren't	what	what's	whatever	when
et	etc	even	ever	every	others	otherwise	ought	our	ours	whence	whenever	where	where's	whereafter
everybody	everyone	everything	everywhere	ex	ourselves	out	outside	over	overall	whereas	whereby	wherein	whereupon	wherever
exactly	example	except	far	few	own	particular	particularly	per	perhaps	whether	which	while	whither	who
fifth	first	five	followed	following	placed	please	plus	possible	presumably	who's	whoever	whole	whom	whose
follows	for	former	formerly	forth	probably	provides	que	quite	qv	why	will	willing	wish	with
four	from	further	furthermore	get	rather	rd	re	really	reasonably	within	without	won't	wonder	would
gets	getting	given	gives	go	regarding	regardless	regards	relatively	respectively	wouldn't	yes	yet	you	you'd
goes	going	gone	got	gotten	right	said	same	saw	say	you'll	you're	you've	your	yours
greetings	had	hadn't	happens	hardly	saying	says	second	secondly	see	yourself	yourselves	zero		
has	hasn't	have	haven't	having	seeing	seem	seemed	seeming	seems					
he	he's	hello	help	hence	seen	self	selves	sensible	sent					

Stop Words

- Example 3: Stop words list in Korean
  - ✓ http://www.ranks.nl/stopwords/korean
    - A total of 677 stop words



아
휴
아이구
아이쿠
아이고
어
나
우리
저희
따라
의해
을
를
에
의
가
으로
로
에게
뿐이다
의거하여
근거하여
입각하여
기준으로
예하면
예를 들면
예를 들자면
저
소인
소생
저희
지말고
하지마
하지마라
다른
물론
또한
그리고
비길수 없다
해서는 안된
다

어찌됐든
그위에
게다가
점에서 보아
비추어 보아
고려하면
하게될것이다
일것이다
비교적
좀
보다더
비하면
시키다
하게하다
할만하다
의해서
연이서
이어서
잇따라
뒤따라
뒤이어
결국
의지하여
기대여
통하여
자마자
더욱더
불구하고
얼마든지
마음대로
주저하지 않고
곧
즉시
바로
당장
하자마자
밖에 안된다
하면된다
그래
그렇지
요컨대

하기보다는
차라리
하는 편이 낫다
흐흐
놀라다
상대적으로 말하
자면
마치
아니라면
쉿
그렇지 않으면
그렇지 않다면
안 그러면
아니었다면
하든지
아니면
이라면
좋아
알았어
하는것도
그만이다
어쩔수 없다
하나
일
일반적으로
일단
한켠으로는
오자마자
이렇게되면
이와같다면
전부
한마디
한항목
근거로
하기에
아울러
하지 않도록
않기 위해서
이르기까지
이 되다
로 인하여

뿐만 아니라
만이 아니다
만은 아니다
막론하고
관계없이
그치지 않다
그러나
그런데
하지만
든간에
논하지 않다
따지지 않다
설사
비록
더라도
아니면
만 못하다
하는 편이 낫
다
불문하고
향하여
향해서
향하다
쪽으로
틈타
이용하여
타다
오르다
제외하고
이 외에
이 밖에
하여야
비로소
한다면 몰라
도
외에도
이곳
여기
부터
기점으로
따라서

다시 말하자면
바꿔 말하면
즉
구체적으로
말하자면
시작하여
시초에
이상
허
헉
허걱
바와같이
해도좋다
해도된다
게다가
더구나
하물며
와르르
팍
퍽
펄렁
동안
이래
하고있었다
이었다
에서
로부터
까지
예하면
했어요
해요
함께
같이
더불어
마저
마저도
양자
모두
습니다
가까스로
하려고하다

까닭으로
이유만으로
이로 인하여
그래서
이 때문에
그러므로
그런 까닭에
알 수 있다
결론을 낼 수 있
다
으로 인하여
있다
어떤것
관계가 있다
관련이 있다
연관되다
어떤것들
에 대해
이리하여
그리하여
여부
하기보다는
하느니
하면 할수록
운운
이러이러하다
하구나
하도다
다시말하면
다음으로
에 있다
에 달려 있다
우리
우리들
오히려
하기는한데
어떻게
어떻해
어찌됐어
어때
어째서

할 생각이다
하려고하다
이리하여
그리하여
그렇게 함으
로써
하지만
일때
할때
앞에서
중에서
보는데서
으로써
로써
까지
해야한다
일것이다
반드시
할줄알다
할수있다
할수있어
임에 틀림없
다
한다면
등
등등
제
겨우
단지
다만
할뿐
딩동
댕그
대해서
대하여
대하면
훨씬
얼마나
얼마만큼
얼마큼
남짓
여

즈음하여
다른
다른 방면으로
해봐요
습니까
했어요
말할것도 없고
무릎쓰고
개의치않고
하는것만 못하다
하는것이 낫다
매
매번
들
모
어느것
어느
로써
갖고말하자면
어디
어느쪽
어느것
어느해
어느 년도
라 해도
언젠가
어떤것
어느것
저기
저쪽
저것
그때
그럼
그러면
요만한걸
그래
그때
저것만큼
그저
이르기까지
할 줄 안다
할 힘이 있다

본대로
자
이
이쪽
여기
이것
이번
이렇게말하자면
이런
이러한
이와 같은
요만큼
요만한 것
얼마 안 되는 것
이만큼
이 정도의
이렇게 많은 것
이와 같다
이때
이렇구나
것과 같이
끼익
삐걱
따위
와 같은 사람들
부류의 사람들
왜냐하면
중의하나
오직
오로지
에 한하다
하기만 하면
도착하다
까지 미치다
도달하다
정도에 이르다
할 지경이다
결과에 이르다
관해서는
여러분
하고 있다
한 후

얼마간
약간
다소
좀
조금
다수
몇
얼마
지만
하물며
또한
그러나
그렇지만
하지만
이외에도
대해 말하자
면
뿐이다
다음에
반대로
반대로 말하
자면
이와 반대로
바꾸어서 말
하면
바꾸어서 한
다면
만약
그렇지않으
면
까악
툭
딱
삐걱거리다
보드득
비걱거리다
꽈당
응당
해야한다
에 가서
각

너
너희
당신
어찌
설마
차라리
할지언정
할지라도
할망정
할지언정
구토하다
게우다
토하다
메쓰겁다
옆사람
퉤
쳇
의거하여
근거하여
의해
따라
힘입어
그
다음
버금
두번째로
기타
첫번째로
나머지는
그중에서
견지에서
형식으로 쓰여
입장에서
위해서
단지
의해되다
하도록시키다
뿐만아니라
반대로
전후
전자

혼자
자기
자기집
자신
우에 종합한것과
같이
총적으로 보면
총적으로 말하면
총적으로
대로 하다
으로서
참
그만이다
할 따름이다
쿵
탕탕
쾅쾅
둥둥
봐
봐라
아이야
아니
와아
응
아이
참나
년
월
일
령
영
일
이
삼
사
오
육
륙
칠
팔
구

…

AGENDA

01 Bag of Words

02 Word Weighting

03 N-Grams

Word Weighting: Term-Frequency (TF)

Nayak & Raghavan (2014)

- Term frequency $\mathrm{tf}_{t,d}$
  - ✓ The number of times that the term **t** occurs in the document **d**

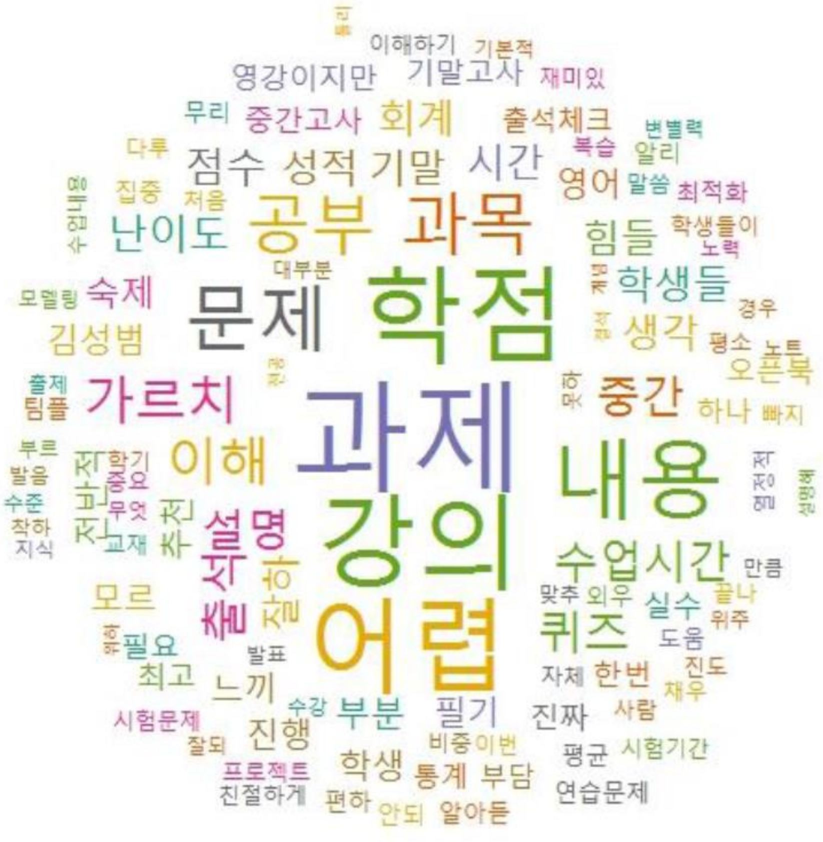
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Word Weighting: Term-Frequency (TF)

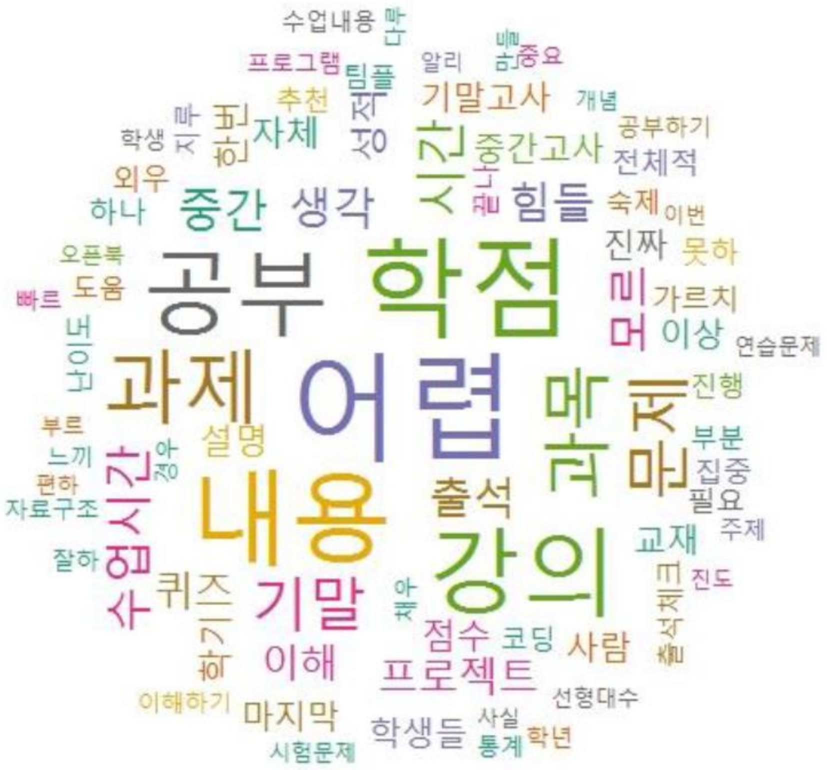
이재천, 김수경, 홍성연 (2015)

- Term frequency $tf_{t,d}$
  - ✓ The more frequently occurs, the more important it is

<산공 강의 상위 25%>



<하위 25%>



Word Weighting: Document Frequency (DF)

- Document frequency df_t
  - ✓ The number of documents in which the term **t** appears.
- Issues on DF
  - ✓ Rare terms are more informative than frequent terms across the document collection
    - is, can, the, of, …
  - ✓ Consider a term in the query that is rare in the collection (e.g., Pneumonoultramicroscopicsilicovolcanoconiosis (longest word in English, 🔊))
  - ✓ A document containing this term is very likely to be relevant to the query
  - ✓ We should give a high weight for rare terms than common terms

Word Weighting: Inverse Document Frequency (IDF)

- Inverse document frequency idf_t
  - ✓ $\text{idf}_t = \log_{10}(N/\text{df}_t)$
  - ✓ We use $\log(N/\text{df}_t)$ instead of N/df_t to "dampen" the effect of idf
- IDF example with N = 1 million

term	$\textbf{df}_t$	$\textbf{idf}_t$
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

Word Weighting: TF-IDF

- TF-IDF
  - ✓ TF-IDF weight of a term is the product of its tf weight and its idf weight

$$TF-IDF(w) = \boxed{tf(w)} \times \log\left(\boxed{\frac{N}{df(w)}}\right)$$

The word is more important if it appears several times in a target document

The word is more important if it appears in less documents

  - ✓ Best known weighting scheme in information retrieval
  - ✓ Increases with the number of occurrences within a document
  - ✓ Increases with the rarity of the term in the collection

Word Weighting: TF-IDF

Nayak & Raghavan (2014)

- Example revisited
  - ✓ Each document is now represented by a real-valued vector of tf-idf weights in $R^{|V|}$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0.35
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

- ✓ So, we have a |V|-dimensional vector space
  - Terms are axes of the space
  - Documents are points or vectors in this space
  - Very high dimensional: need to reduce the number of features!
  - Sparseness: most entries are zero

Word Weighting: TF-IDF

- TF-IDF Example
  - ✓ Q1: Which term is the most important for the document 1?
  - ✓ Q2: Which term is the least important for the document 1?

	Doc1	Doc2	Doc3
Term1	5	0	0
Term2	1	0	0
Term3	5	5	5
Term4	3	3	3
Term5	3	0	1

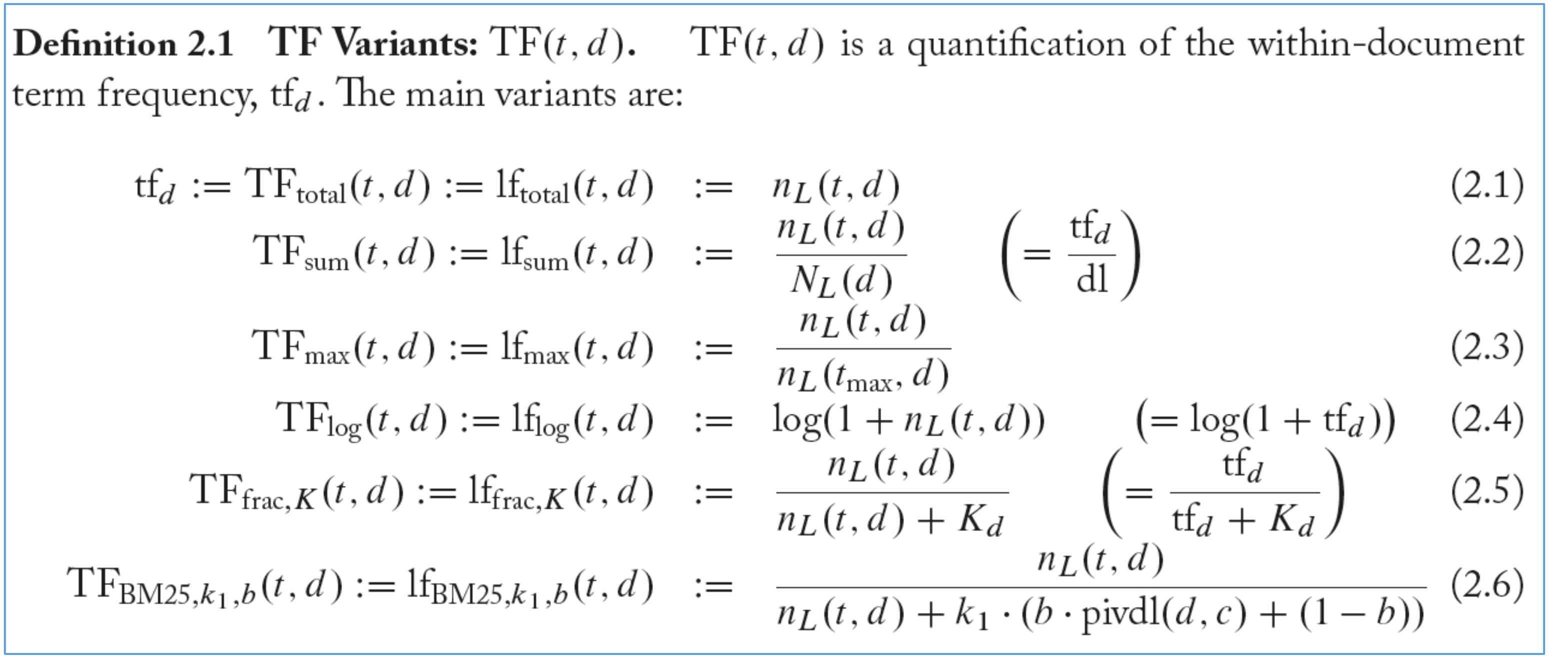
Doc1	TF	DF	IDF	TF-IDF
Term1	5	1	Log3	5log3
Term2	1	1	Log3	1log3
Term3	5	3	Log1	0
Term4	3	3	Log1	0
Term5	3	2	Log(3/2)	3log(3/2)

Word weighting: Term 1 > Term 5 > Term 2 > Term 3 = Term 4

TF Variants

Roelleke (2013)

- TF Variants



Definition 2.1 TF Variants: $\mathrm{TF}(t,d)$. $\mathrm{TF}(t,d)$ is a quantification of the within-document term frequency, tf_d. The main variants are:

$$\mathrm{tf}_d := \mathrm{TF}_{\mathrm{total}}(t,d) := \mathrm{lf}_{\mathrm{total}}(t,d) := n_L(t,d) \tag{2.1}$$

$$\mathrm{TF}_{\mathrm{sum}}(t,d) := \mathrm{lf}_{\mathrm{sum}}(t,d) := \frac{n_L(t,d)}{N_L(d)} \quad \left(= \frac{\mathrm{tf}_d}{\mathrm{dl}}\right) \tag{2.2}$$

$$\mathrm{TF}_{\mathrm{max}}(t,d) := \mathrm{lf}_{\mathrm{max}}(t,d) := \frac{n_L(t,d)}{n_L(t_{\mathrm{max}},d)} \tag{2.3}$$

$$\mathrm{TF}_{\mathrm{log}}(t,d) := \mathrm{lf}_{\mathrm{log}}(t,d) := \log(1+n_L(t,d)) \quad (= \log(1+\mathrm{tf}_d)) \tag{2.4}$$

$$\mathrm{TF}_{\mathrm{frac},K}(t,d) := \mathrm{lf}_{\mathrm{frac},K}(t,d) := \frac{n_L(t,d)}{n_L(t,d)+K_d} \quad \left(= \frac{\mathrm{tf}_d}{\mathrm{tf}_d+K_d}\right) \tag{2.5}$$

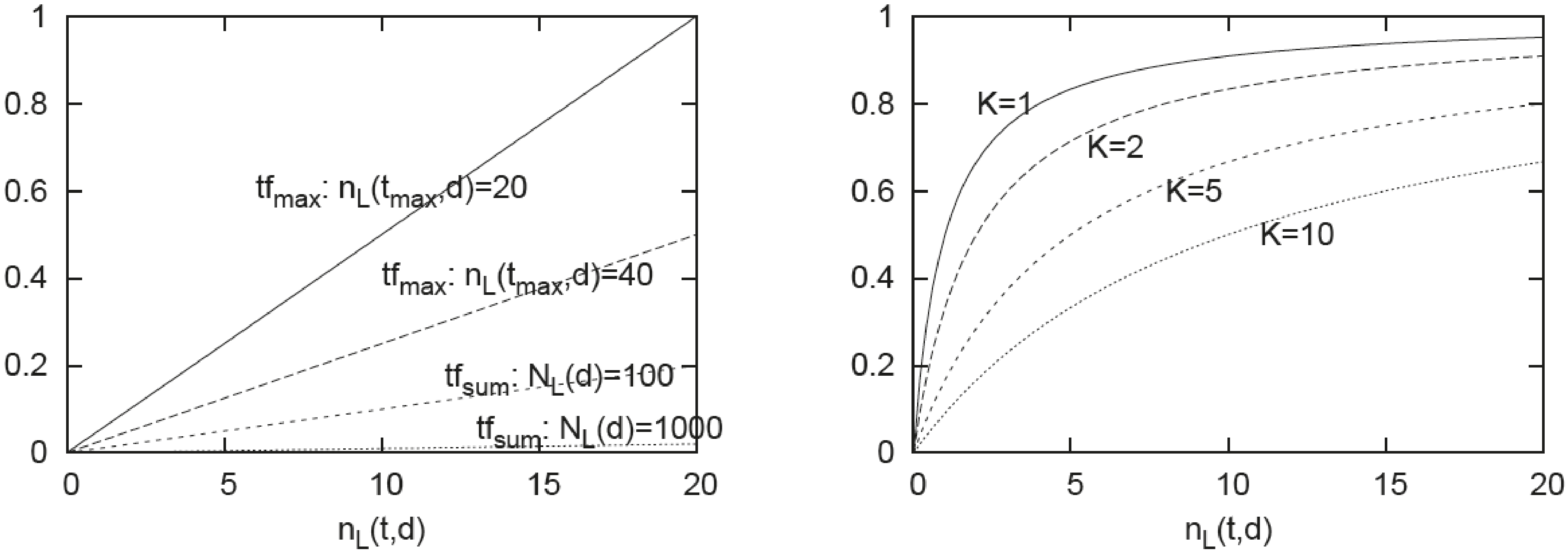
$$\mathrm{TF}_{\mathrm{BM25},k_1,b}(t,d) := \mathrm{lf}_{\mathrm{BM25},k_1,b}(t,d) := \frac{n_L(t,d)}{n_L(t,d)+k_1\cdot(b\cdot\mathrm{pivdl}(d,c)+(1-b))} \tag{2.6}$$

- K_d: (document length)/(average document length)

TF Variants

Roelleke (2013)

- TF Variants



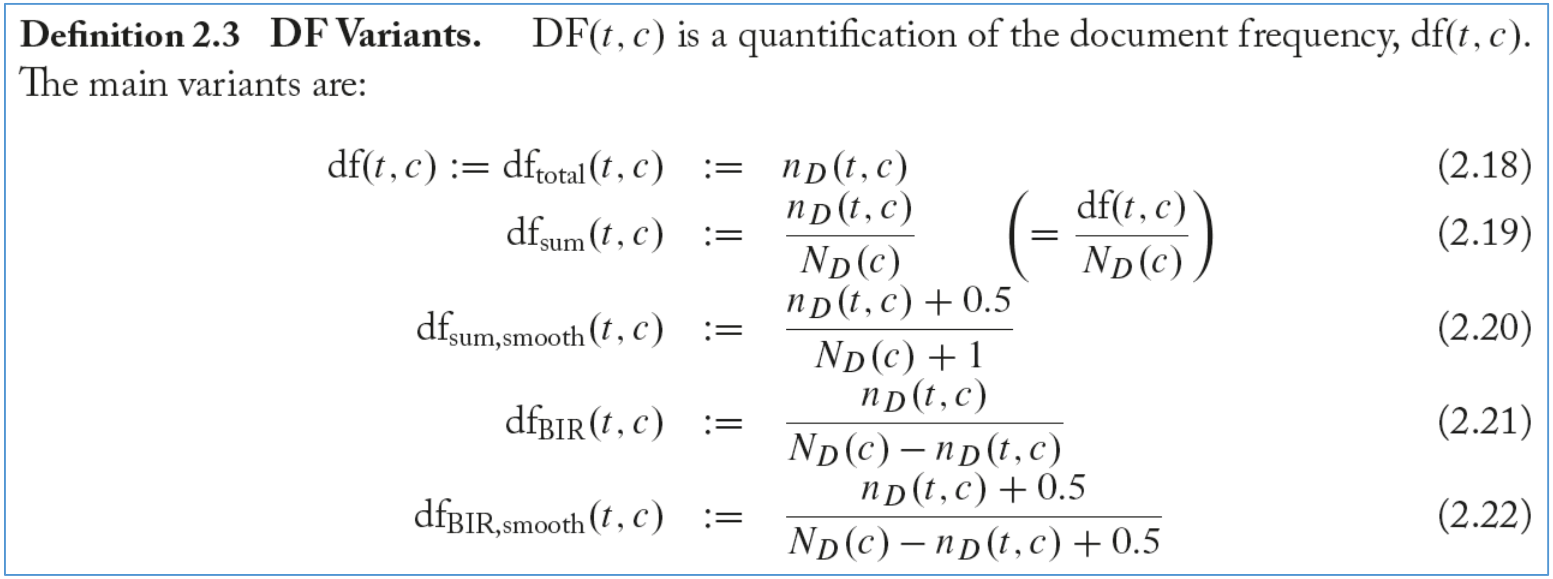
(a) $\mathrm{TF}_{\mathrm{sum}}$ and $\mathrm{TF}_{\mathrm{max}}$

(b) $\mathrm{TF}_{\mathrm{frac}}$

DF & IDF Variants

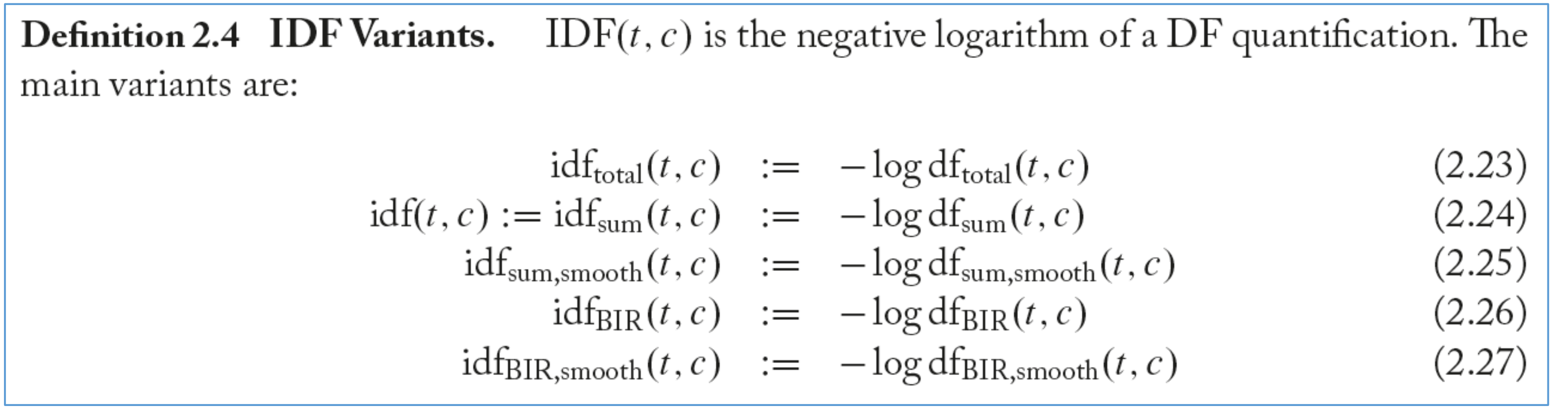
Roelleke (2013)

- DF & IDF Variants



Definition 2.3 DF Variants. DF(t,c) is a quantification of the document frequency, df(t,c). The main variants are:

$$
\begin{aligned}
\mathrm{df}(t,c) := \mathrm{df}_{\mathrm{total}}(t,c) &:= n_D(t,c) && (2.18)\\
\mathrm{df}_{\mathrm{sum}}(t,c) &:= \frac{n_D(t,c)}{N_D(c)} \qquad \left(= \frac{\mathrm{df}(t,c)}{N_D(c)}\right) && (2.19)\\
\mathrm{df}_{\mathrm{sum,smooth}}(t,c) &:= \frac{n_D(t,c)+0.5}{N_D(c)+1} && (2.20)\\
\mathrm{df}_{\mathrm{BIR}}(t,c) &:= \frac{n_D(t,c)}{N_D(c)-n_D(t,c)} && (2.21)\\
\mathrm{df}_{\mathrm{BIR,smooth}}(t,c) &:= \frac{n_D(t,c)+0.5}{N_D(c)-n_D(t,c)+0.5} && (2.22)
\end{aligned}
$$



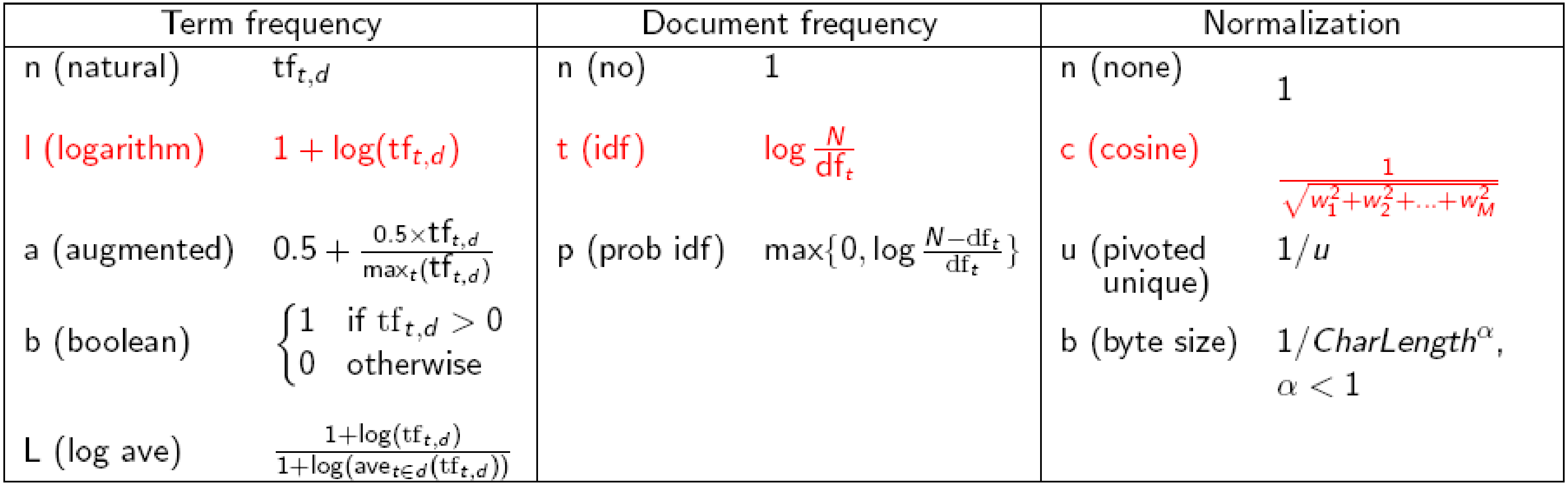
Definition 2.4 IDF Variants. IDF(t,c) is the negative logarithm of a DF quantification. The main variants are:

$$
\begin{aligned}
\mathrm{idf}_{\mathrm{total}}(t,c) &:= -\log \mathrm{df}_{\mathrm{total}}(t,c) && (2.23)\\
\mathrm{idf}(t,c) := \mathrm{idf}_{\mathrm{sum}}(t,c) &:= -\log \mathrm{df}_{\mathrm{sum}}(t,c) && (2.24)\\
\mathrm{idf}_{\mathrm{sum,smooth}}(t,c) &:= -\log \mathrm{df}_{\mathrm{sum,smooth}}(t,c) && (2.25)\\
\mathrm{idf}_{\mathrm{BIR}}(t,c) &:= -\log \mathrm{df}_{\mathrm{BIR}}(t,c) && (2.26)\\
\mathrm{idf}_{\mathrm{BIR,smooth}}(t,c) &:= -\log \mathrm{df}_{\mathrm{BIR,smooth}}(t,c) && (2.27)
\end{aligned}
$$

TF-IDF Variants Summary

Roelleke (2013)

- The most commonly used TF-IDF in general

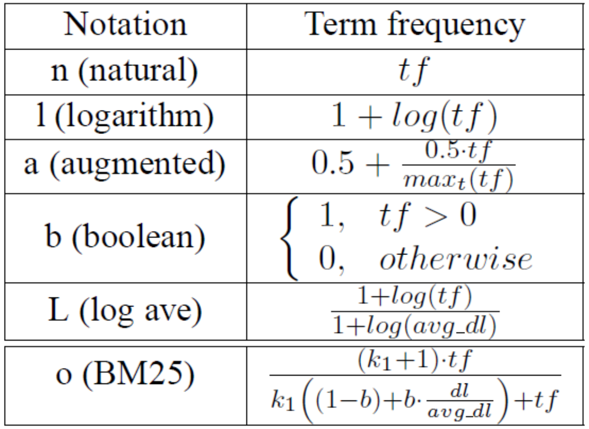


Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log\frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0,\log\frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases}1 & \text{if } \mathrm{tf}_{t,d}>0\\0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

Effects of TF-IDF Variants

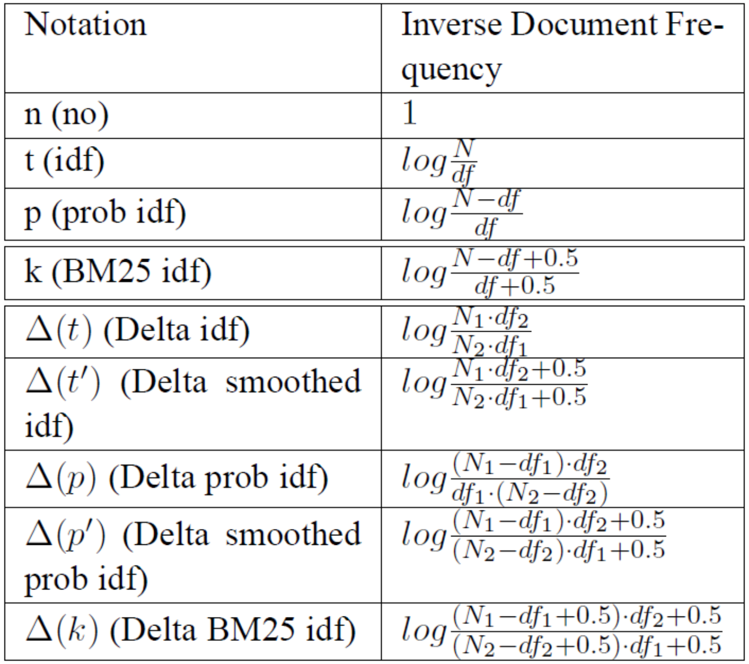
- Comparative Study (Paltoglou and Thelwall, 2010)
  - ✓ Task 1: Classification of 2,000 movie reviews: positive vs. negative
  - ✓ Task 2: Multi-Domain Sentiment Data set (MDSD)
    - Four different product types: books, electronics, DVDs, and kitchen appliances
    - 1,000 positive & 1,000 negative for each type, 8,000 in total

Term Frequency



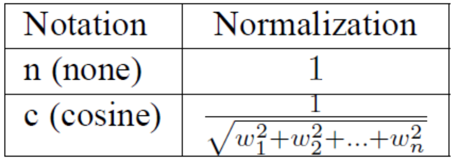
Notation	Term frequency
n (natural)	tf
l (logarithm)	$1 + log(tf)$
a (augmented)	$0.5 + \frac{0.5 \cdot tf}{max_t(tf)}$
b (boolean)	$\begin{cases} 1, & tf > 0 \\ 0, & otherwise \end{cases}$
L (log ave)	$\frac{1+log(tf)}{1+log(avg\_dl)}$
o (BM25)	$\frac{(k_1+1)\cdot tf}{k_1\left((1-b)+b\cdot\frac{dl}{avg\_dl}\right)+tf}$

Inverse Document Frequency



Notation	Inverse Document Frequency
n (no)	1
t (idf)	$log\frac{N}{df}$
p (prob idf)	$log\frac{N-df}{df}$
k (BM25 idf)	$log\frac{N-df+0.5}{df+0.5}$
$\Delta(t)$ (Delta idf)	$log\frac{N_1\cdot df_2}{N_2\cdot df_1}$
$\Delta(t')$ (Delta smoothed idf)	$log\frac{N_1\cdot df_2+0.5}{N_2\cdot df_1+0.5}$
$\Delta(p)$ (Delta prob idf)	$log\frac{(N_1-df_1)\cdot df_2}{df_1\cdot(N_2-df_2)}$
$\Delta(p')$ (Delta smoothed prob idf)	$log\frac{(N_1-df_1)\cdot df_2+0.5}{(N_2-df_2)\cdot df_1+0.5}$
$\Delta(k)$ (Delta BM25 idf)	$log\frac{(N_1-df_1+0.5)\cdot df_2+0.5}{(N_2-df_2+0.5)\cdot df_1+0.5}$

Normalization



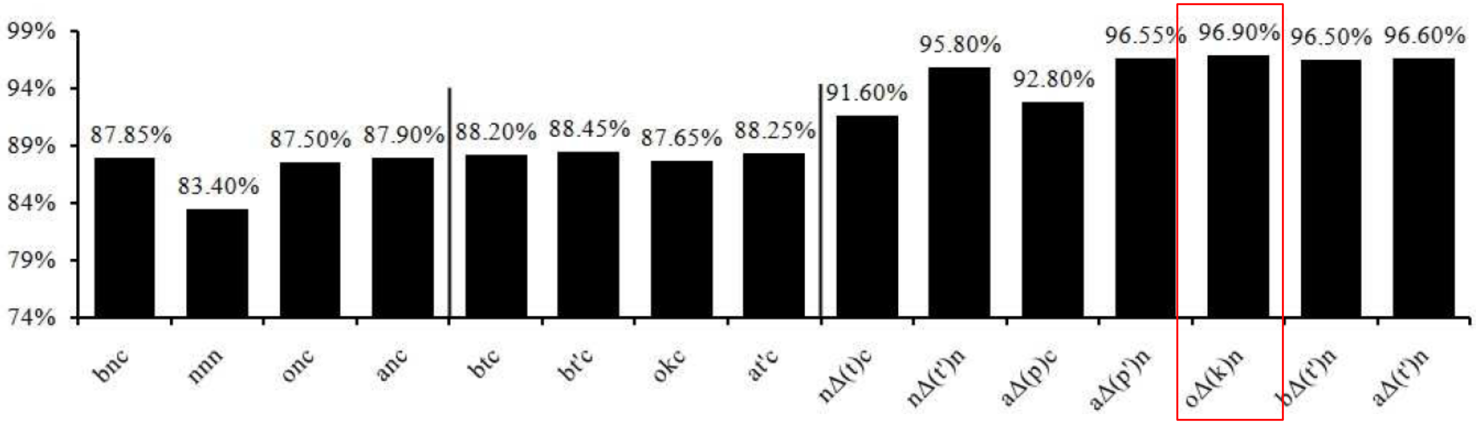
Notation	Normalization
n (none)	1
c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+...+w_n^2}}$

Effects of TF-IDF Variants

Paltoglou and Thelwall (2010)

- Experimental Result 1: Movie Reviews
  - ✓ Base classifier: support vector machine (SVM)

Data set	#Documents	#Terms	#Unique Terms	Average #Terms per Document
Movie Reviews	2,000	1,336,883	39,399	668
Multi-Domain Sentiment Dataset (MDSD)	8,000	1,741,085	455,943	217
BLOGS06	17,898	51,252,850	367,899	2,832



o (BM25)	$\frac{(k_1+1)\cdot tf}{k_1\left((1-b)+b\cdot\frac{dl}{avg\_dl}\right)+tf}$

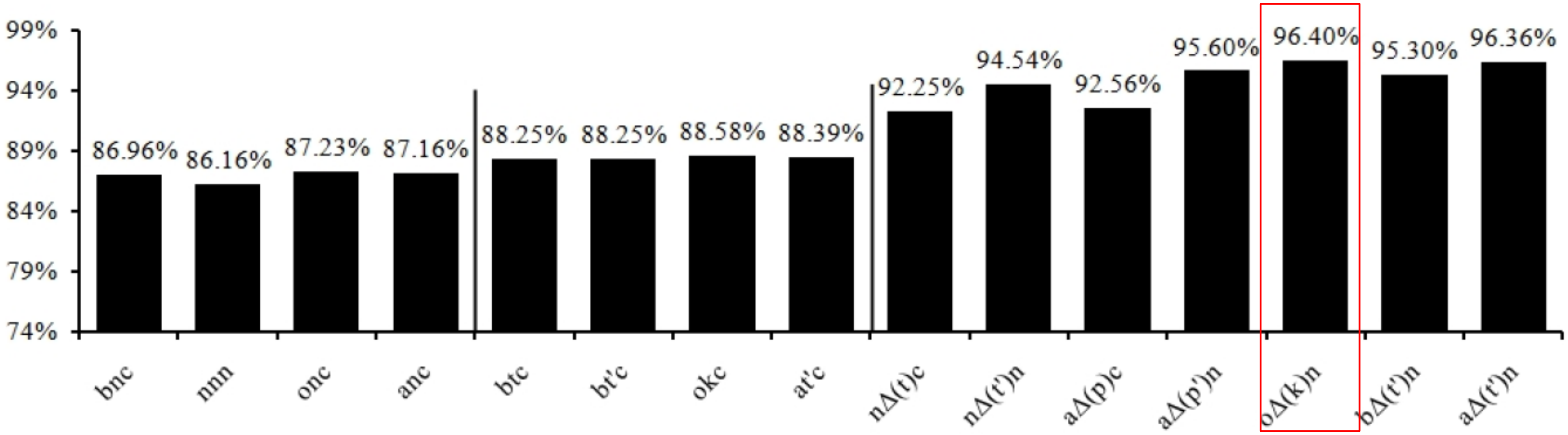
$\Delta(k)$ (Delta BM25 idf)	$log\frac{(N_1-df_1+0.5)\cdot df_2+0.5}{(N_2-df_2+0.5)\cdot df_1+0.5}$

n (none)	1

Effects of TF-IDF Variants

Paltoglou and Thelwall (2010)

- Experimental Result 2: MDSD
  - ✓ Base classifier: support vector machine (SVM)



o (BM25)	$\frac{(k_1+1)\cdot tf}{k_1\left((1-b)+b\cdot\frac{dl}{avg\_dl}\right)+tf}$

$\Delta(k)$ (Delta BM25 idf)	$log\frac{(N_1-df_1+0.5)\cdot df_2+0.5}{(N_2-df_2+0.5)\cdot df_1+0.5}$

n (none)	1

AGENDA

01 Bag of Words

02 Word Weighting

03 N-Grams

N-Grams

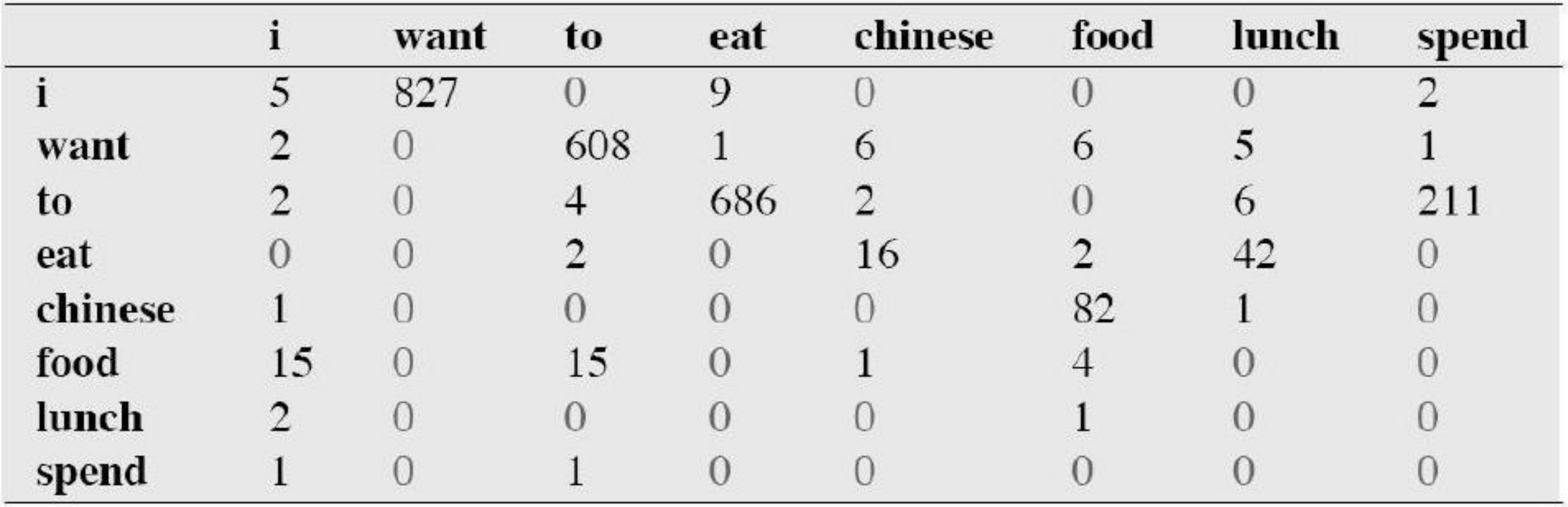
- N-Gram-based Language Models in NLP
  - ✓ Use the previous N-1 words in a sequence to predict the next word

$$P(w_n|w_{n-1}, w_{n-2},\ldots, w_1) = \frac{P(w_n,w_{n-1},w_{n-2},\ldots, w_1)}{P(w_{n-1},w_{n-2},\ldots, w_1)}$$

  - ✓ Q) One of the hottest topics in artificial intelligence is deep ______
    - blue vs. frying vs. learning ?
- N-Gram in Text Mining
  - ✓ Some phrases are very useful in text clustering/categorization!
    - Six sigma, supply chain management, big data, etc.
  - ✓ Term-frequency for n-grams can be utilized.
  - ✓ Domain-dependent.

N-Grams

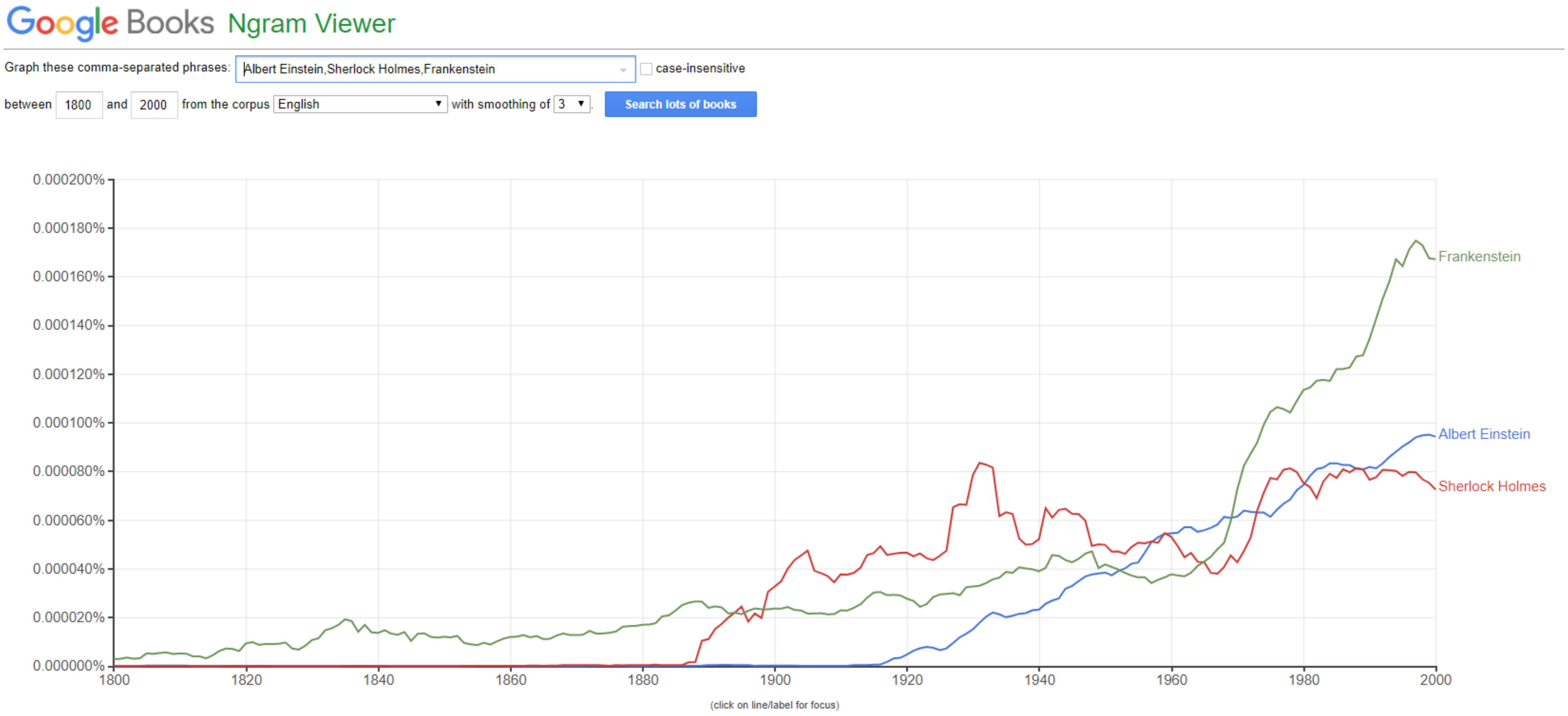
- Bigram example
  - ✓ Total counts in a corpus



	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

N-Grams

- Google Books Ngram Viewer (https://books.google.com/ngrams)



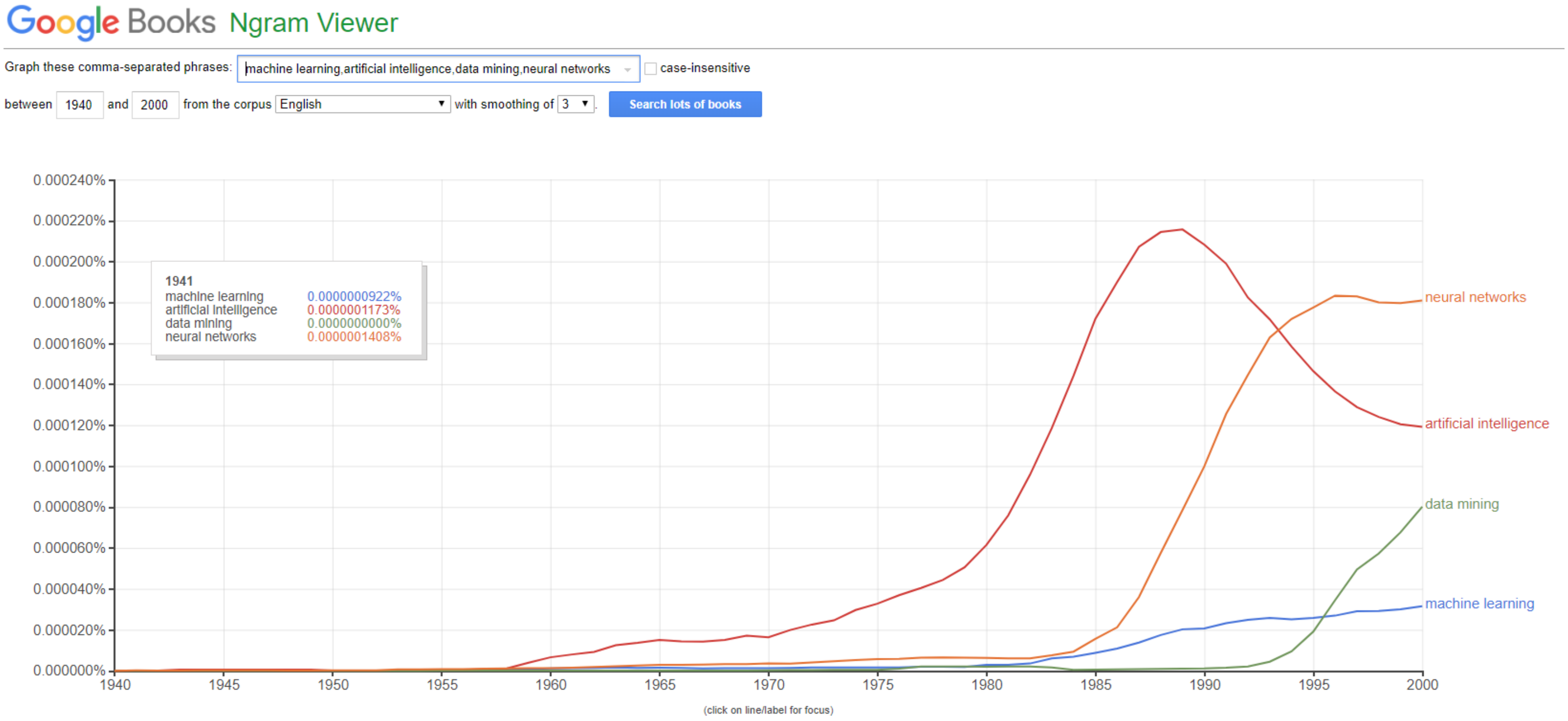
Google Books Ngram Viewer

Graph these comma-separated phrases: Albert Einstein,Sherlock Holmes,Frankenstein □ case-insensitive

between 1800 and 2000 from the corpus English with smoothing of 3 .

N-Grams

- Google Books Ngram Viewer (https://books.google.com/ngrams)
  - ✓ Ngram frequencies for "artificial intelligence", "machine learning", "data mining", and "neural networks"



Google Books Ngram Viewer

Graph these comma-separated phrases: machine learning,artificial intelligence,data mining,neural networks ☐ case-insensitive

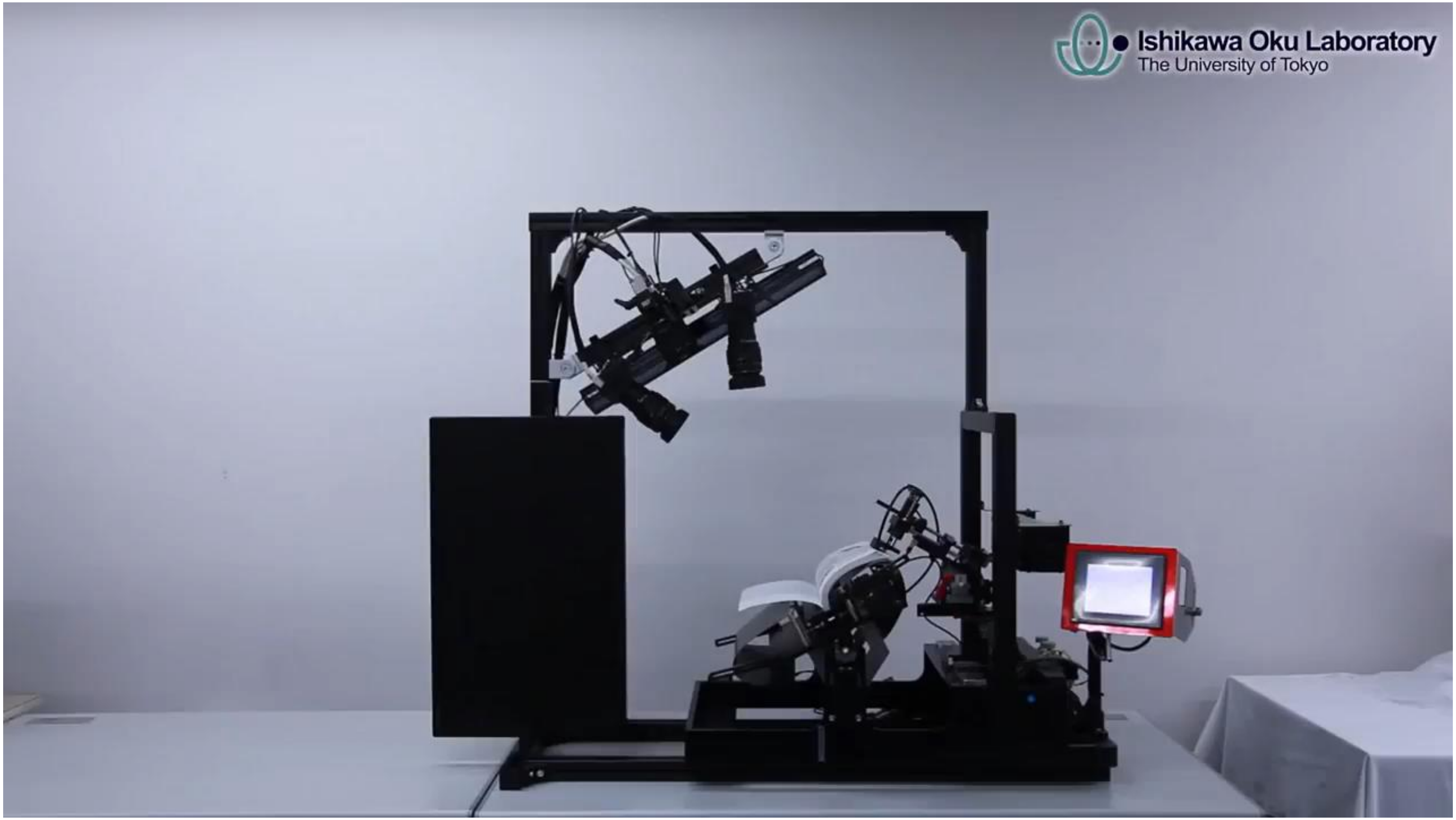
between 1940 and 2000 from the corpus English with smoothing of 3 . Search lots of books

1941

machine learning	0.0000000922%
artificial intelligence	0.0000001173%
data mining	0.0000000000%
neural networks	0.0000001408%

(click on line/label for focus)

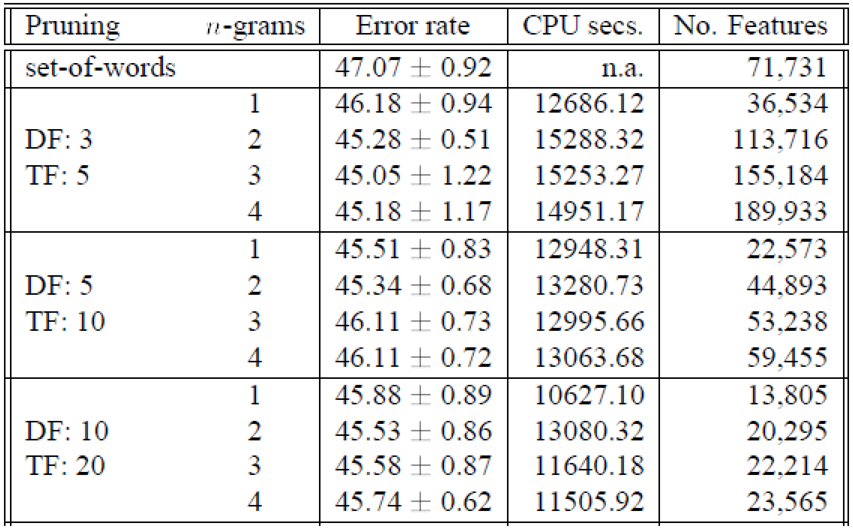
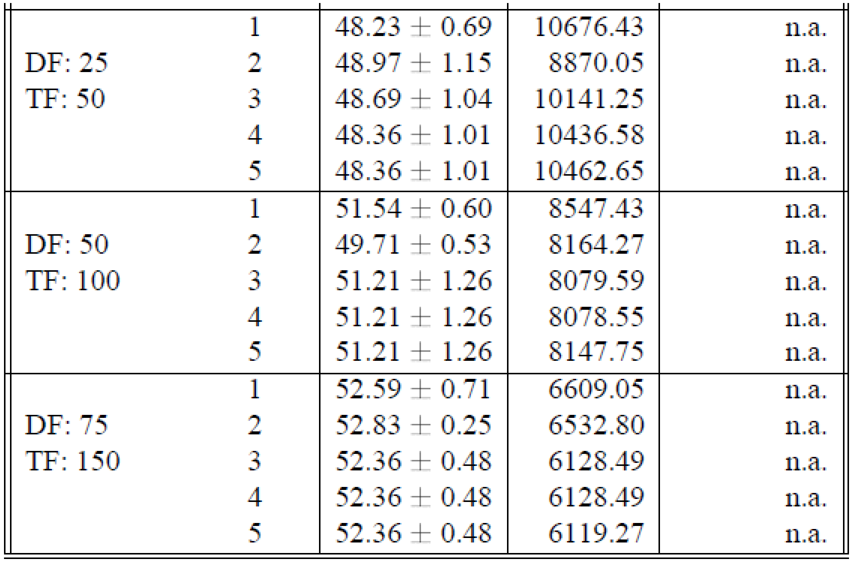
N-Grams



N-Grams

Furnkranz (1998)

- Empirical evaluation
  - ✓ Data sets
    - 20 newsgroup data set: 20,000 articles (1,000 for each category)
    - 21578 REUTERS newswire articles: 21,578 articles with 90 categories
  - ✓ Classification algorithm: RIPPER
- Results for 20 newsgroup dataset

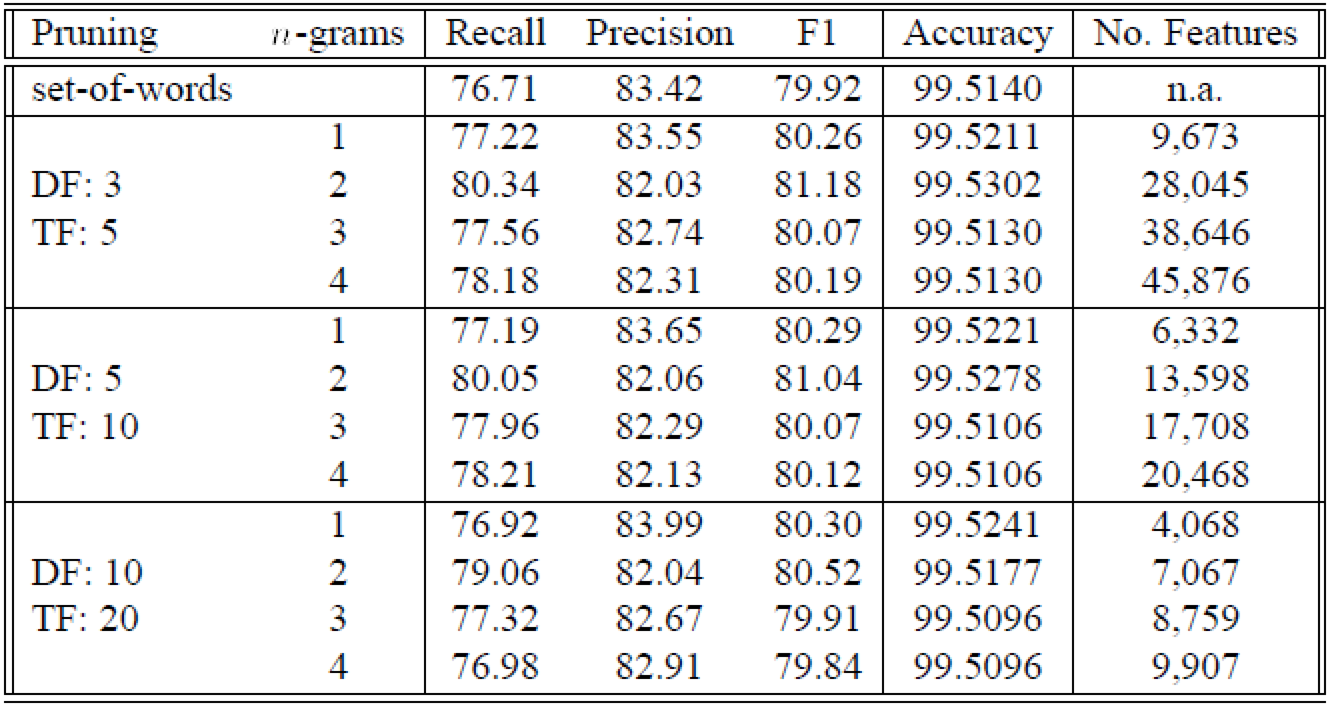



Pruning	n-grams	Error rate	CPU secs.	No. Features
set-of-words		47.07 ± 0.92	n.a.	71,731
DF: 3 TF: 5	1	46.18 ± 0.94	12686.12	36,534
	2	45.28 ± 0.51	15288.32	113,716
	3	45.05 ± 1.22	15253.27	155,184
	4	45.18 ± 1.17	14951.17	189,933
DF: 5 TF: 10	1	45.51 ± 0.83	12948.31	22,573
	2	45.34 ± 0.68	13280.73	44,893
	3	46.11 ± 0.73	12995.66	53,238
	4	46.11 ± 0.72	13063.68	59,455
DF: 10 TF: 20	1	45.88 ± 0.89	10627.10	13,805
	2	45.53 ± 0.86	13080.32	20,295
	3	45.58 ± 0.87	11640.18	22,214
	4	45.74 ± 0.62	11505.92	23,565
DF: 25 TF: 50	1	48.23 ± 0.69	10676.43	n.a.
	2	48.97 ± 1.15	8870.05	n.a.
	3	48.69 ± 1.04	10141.25	n.a.
	4	48.36 ± 1.01	10436.58	n.a.
	5	48.36 ± 1.01	10462.65	n.a.
DF: 50 TF: 100	1	51.54 ± 0.60	8547.43	n.a.
	2	49.71 ± 0.53	8164.27	n.a.
	3	51.21 ± 1.26	8079.59	n.a.
	4	51.21 ± 1.26	8078.55	n.a.
	5	51.21 ± 1.26	8147.75	n.a.
DF: 75 TF: 150	1	52.59 ± 0.71	6609.05	n.a.
	2	52.83 ± 0.25	6532.80	n.a.
	3	52.36 ± 0.48	6128.49	n.a.
	4	52.36 ± 0.48	6128.49	n.a.
	5	52.36 ± 0.48	6119.27	n.a.

N-Grams

Furnkranz (1998)

- Results for 21578 REUTERS
  - ✓ Classification accuracy is the highest with bigram features



Pruning	n-grams	Recall	Precision	F1	Accuracy	No. Features
set-of-words		76.71	83.42	79.92	99.5140	n.a.
DF: 3 TF: 5	1	77.22	83.55	80.26	99.5211	9,673
	2	80.34	82.03	81.18	99.5302	28,045
	3	77.56	82.74	80.07	99.5130	38,646
	4	78.18	82.31	80.19	99.5130	45,876
DF: 5 TF: 10	1	77.19	83.65	80.29	99.5221	6,332
	2	80.05	82.06	81.04	99.5278	13,598
	3	77.96	82.29	80.07	99.5106	17,708
	4	78.21	82.13	80.12	99.5106	20,468
DF: 10 TF: 20	1	76.92	83.99	80.30	99.5241	4,068
	2	79.06	82.04	80.52	99.5177	7,067
	3	77.32	82.67	79.91	99.5096	8,759
	4	76.98	82.91	79.84	99.5096	9,907

ANY
questions?

References

Research Papers & Other materials

- Furnkranz, J. (1998). A Study using N-gram Features for Text Categorization. Austrian Research Institute for Artificial Intelligence Technical Report OEFAI-TR-98-30 Schottengasse.
- Nayak, P. and Raghavan, P. (2014). Lecture 6: Scoring, Term Weighting and the Vector Space Model. http://web.stanford.edu/class/cs276/handouts/lecture6-tfidf-handout-1-per.pdf
- Paltoglou, G. and Thelwall, M. (2010). A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In Proceddings of the 48th Annual Meeting of the Association for Computational Linguistics: 1386-1395.
- Roelleke, T. (2013). Information Retrieval Models: Foundations and Relationships. Morgan & Claypool Publishers.
- 박은정, Supervised Feature Representationsfor Document Classification, PhD Thesis, Seoul National University, 2016.
- 이재천, 김수경, 홍성연. (2015). Data Mining을 이용한 고려대 강의평가 분석: Klue 사이트 강의평가를 기준으로. 2015 캡스톤디자인 II Term Project.