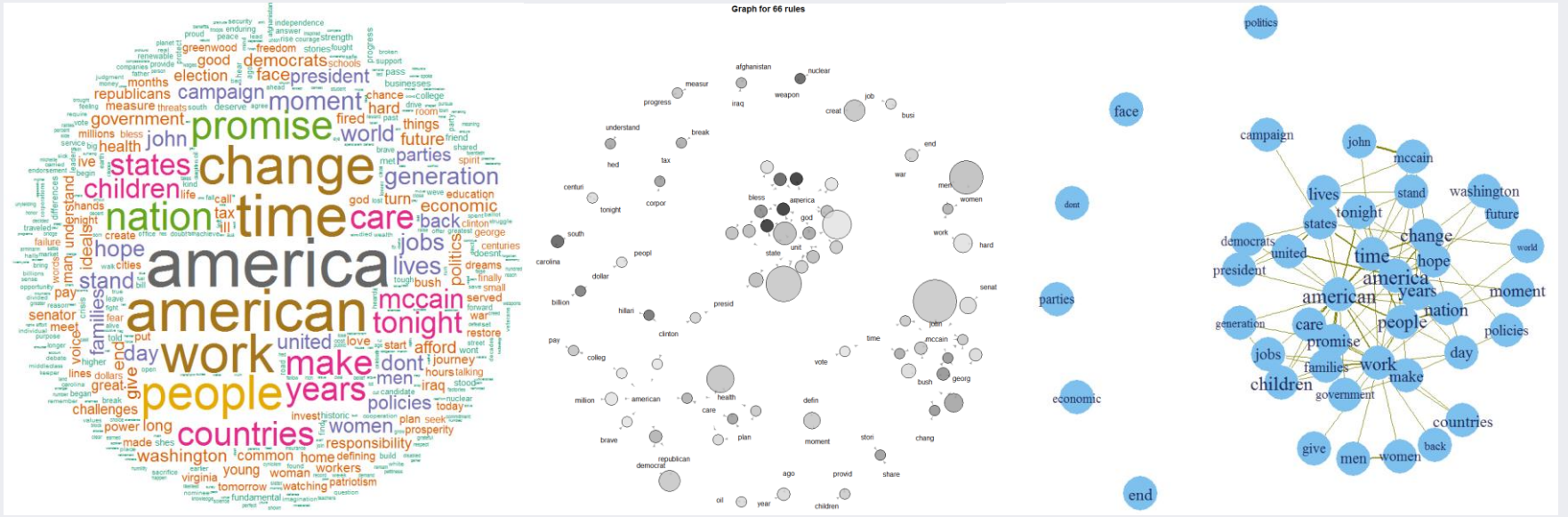


Lecture 5: Text Data Analysis

Pilsung Kang

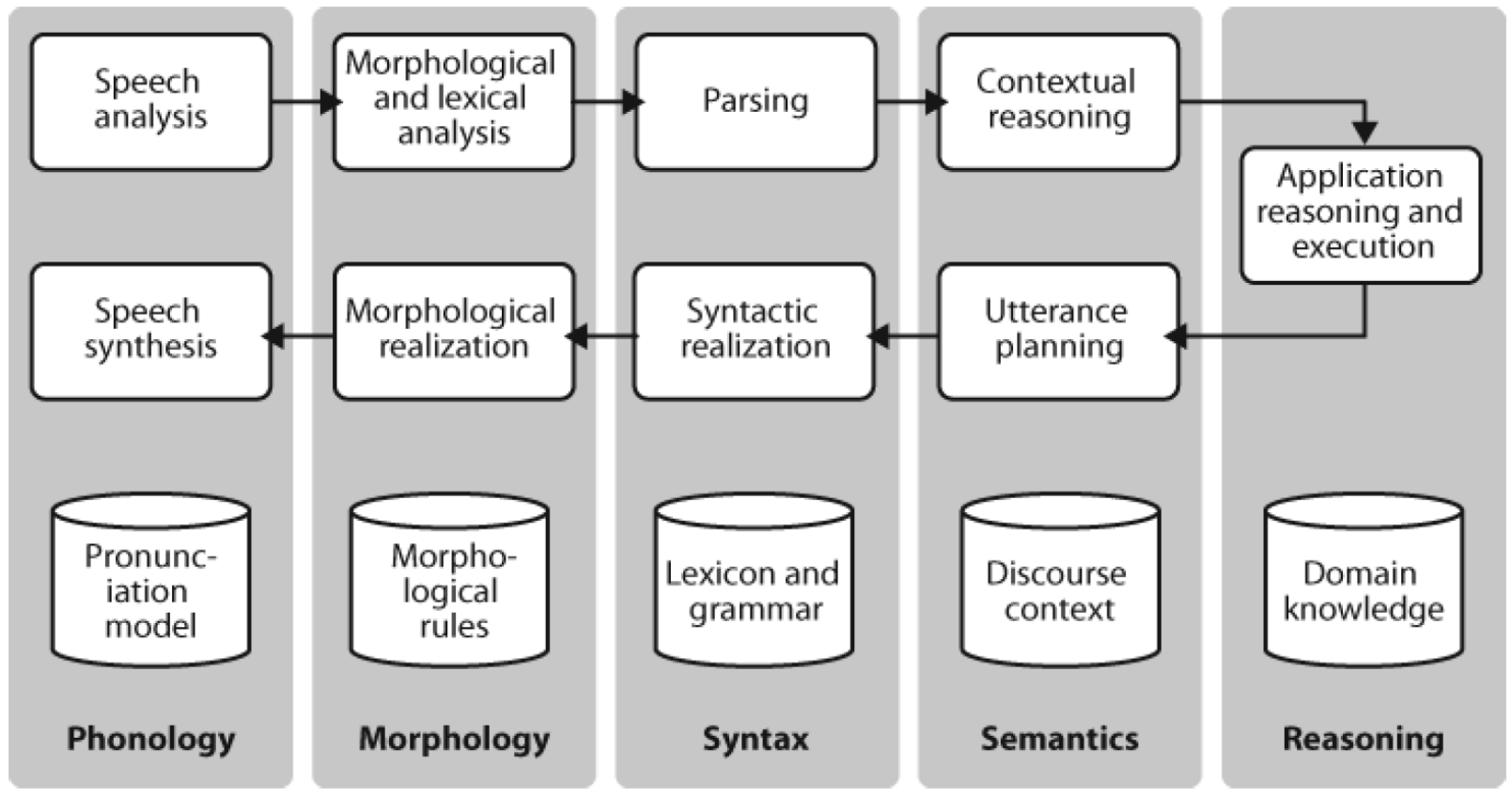
School of Industrial Management Engineering

Korea University

AGENDA

Text Preprocessing

- Natural Language Processing



Text Preprocessing

- Classical categorization of NLP

Classical Categorization

To deal with the complexity of natural langauge, it is typically regarded on several levels (cf. Jurafsky & Martin):

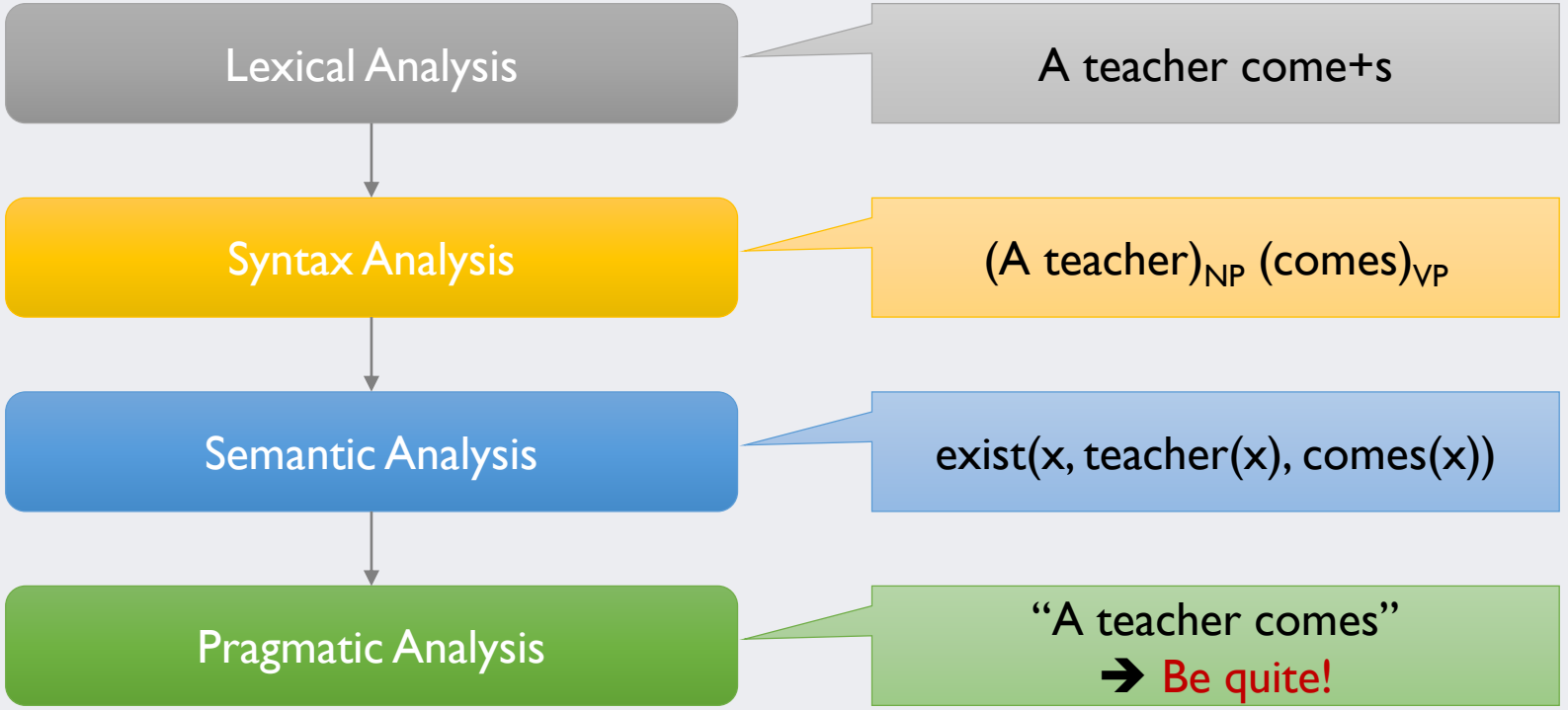
Phonology	the study of linguistic sounds
Morphology	the study of meaningful components of words
Syntax	the study of structural relationships between words
Semantics	the study of meaning
Pragmatics	the study of how language is used to accomplish goals
Discourse	the study of larger linguistic units

Importance for Text Mining

- *Phonology* only concerns spoken language
- *Discourse*, *Pragmatics*, and even *Semantics* is still rarely used

Text Preprocessing

- An example of NLP



Text Preprocessing: Lexical Analysis

- Goals of lexical analysis
  - ✓ Convert a sequence of characters into a sequence of tokens, i.e., meaningful character strings.
    - In natural language processing, morpheme is a basic unit
    - In text mining, word is commonly used as a basic unit for analysis
- Process of lexical analysis
  - ✓ Tokenizing
  - ✓ Part-of-Speech (POS) tagging
  - ✓ Additional analysis: named entity recognition (NER), noun phrase recognition, sentence split, chunking, etc.

Lexical Analysis 1: Sentence Splitting

Witte (2016)

- Sentence is very important in NLP, but it is not critical for some Text Mining tasks

Mark Sentence Boundaries

Detects sentence units. Easy case:

- often, sentences end with ".", "!", or "?"

Hard (or annoying) cases:

- difficult when a "." do not indicate an EOS:
  "MR. X", "3.14", "Y Corp.", ...
- we can detect common abbreviations ("U.S."), but what if a sentence ends with one?
  "...announced today by the U.S. The..
- Sentences can be *nested* (e.g., within quotes)

Correct sentence boundary is important

for many downstream analysis tasks:

- POS-Taggers maximize probabilites of tags within a sentence
- Summarization systems rely on correct detection of sentence

Lexical Analysis 2: Tokenization

- Text is split into basic units called Tokens
  - ✓ word tokens, number tokens, space tokens, …

```
> crude[[1]]
<<PlainTextDocument (metadata: 15)>>
Diamond Shamrock Corp said that
effective today it had cut its contract prices for crude oil by
1.50 dlrs a barrel.
    The reduction brings its posted price for West Texas
Intermediate to 16.00 dlrs a barrel, the copany said.
    "The price reduction today was made in the light of falling
oil product prices and a weak crude oil market," a company
spokeswoman said.
    Diamond is the latest in a line of U.S. oil companies that
have cut its contract, or posted, prices over the last two days
citing weak oil markets.
 Reuter
```

```
> MC_tokenizer(crude[[1]])
  [1] "Diamond"     "Shamrock"    "Corp"        "said"        "that"
  [6] "effective"   "today"       "it"          "had"         "cut"
 [11] "its"         "contract"    "prices"      "for"         "crude"
 [16] "oil"         "by"          ""            ""            ""
 [21] ""            ""            "dlrs"        "a"           "barrel"
 [26] ""            ""            ""            ""            ""
 [31] "The"         "reduction"   "brings"      "its"         "posted"
 [36] "price"       "for"         "West"        "Texas"       "Intermediate"
 [41] "to"          ""            ""            ""            ""
 [46] ""            ""            "dlrs"        "a"           "barrel"
 [51] ""            "the"         "copany"      "said"        ""
 [56] ""            ""            ""            ""            ""
 [61] "The"         "price"       "reduction"   "today"       "was"
 [66] "made"        "in"          "the"         "light"       "of"
 [71] "falling"     "oil"         "product"     "prices"      "and"
 [76] "a"           "weak"        "crude"       "oil"         "market"
 [81] ""            ""            "a"           "company"     "spokeswoman"
 [86] "said"        ""            ""            ""            ""
 [91] ""            "Diamond"     "is"          "the"         "latest"
 [96] "in"          "a"           "line"        "of"          "U"
[101] "S"           ""            "oil"         "companies"   "that"
[106] "have"        "cut"         "its"         "contract"    ""
[111] "or"          "posted"      ""            "prices"      "over"
[116] "the"         "last"        "two"         "days"        "citing"
[121] "weak"        "oil"         "markets"     ""            ""
[126] "Reuter"
```

```
> scan_tokenizer(crude[[1]])
  [1] "Diamond"     "Shamrock"    "Corp"        "said"        "that"
  [6] "effective"   "today"       "it"          "had"         "cut"
 [11] "its"         "contract"    "prices"      "for"         "crude"
 [16] "oil"         "by"          "1.50"        "dlrs"        "a"
 [21] "barrel."     "The"         "reduction"   "brings"      "its"
 [26] "posted"      "price"       "for"         "West"        "Texas"
 [31] "Intermediate" "to"         "16.00"       "dlrs"        "a"
 [36] "barrel,"     "the"         "copany"      "said."       "\"The"
 [41] "price"       "reduction"   "today"       "was"         "made"
 [46] "in"          "the"         "light"       "of"          "falling"
 [51] "oil"         "product"     "prices"      "and"         "a"
 [56] "weak"        "crude"       "oil"         "market,\""   "a"
 [61] "company"     "spokeswoman" "said."       "Diamond"     "is"
 [66] "the"         "latest"      "in"          "a"           "line"
 [71] "of"          "U.S."        "oil"         "companies"   "that"
 [76] "have"        "cut"         "its"         "contract,"   "or"
 [81] "posted,"     "prices"      "over"        "the"         "last"
 [86] "two"         "days"        "citing"      "weak"        "oil"
 [91] "markets."    "Reuter"
```

	MC	Scan
Space	Not removed	Removed
Punctuation	Removed	Not removed
Numbers	Removed	Not removed
Special characters	Removed	Not removed

Lexical Analysis 3: Morphological Analysis

- Morphological Variants: Stemming and Lemmatization

Morphological Variants

Words are changed through a morphological process called *inflection*:

- typically indicates changes in case, gender, number, tense, etc.
- example *car* → cars, *give* → *gives, gave, given*

Goal: "normalize" words

Stemming and Lemmatization

Two main approaches to normalization:

Stemming reduce words to a *base form*

Lemmatization reduce words to their *lemma*

Main difference: stemming just finds **any** base form, which doesn't even need to be a word in the language! Lemmatization find the actual *root* of a word, but requires morphological analysis.

Lexical Analysis 3: Morphological Analysis

- Stemming

Stemming

Commonly used in Information Retrieval:

- Can be achieved with rule-based algorithms, usually based on suffix-stripping
- Standard algorithm for English: the *Porter* stemmer
- Advantages: simple & fast
- Disadvantages:
  - Rules are language-dependent
  - Can create words that do not exist in the language, e.g., *computers* $\longrightarrow$ *comput*
  - Often reduces different words to the same stem, e.g., *army, arm* $\longrightarrow$ *arm*
    stocks, stockings $\longrightarrow$ *stock*
- Stemming for German: German stemmer in the full-text search engine *Lucene*, *Snowball* stemmer with German rule file

Lexical Analysis 3: Morphological Analysis

- Lemmatization

Lemmatization

Lemmatization is the process of deriving the base form, or *lemma*, of a word from one of its inflected forms. This requires a morphological analysis, which in turn typically requires a *lexicon*.

- Advantages:
  - identifies the *lemma* (root form), which is an actual word
  - less errors than in stemming
- Disadvantages:
  - more complex than stemming, slower
  - requires additional language-dependent resources
- While stemming is good enough for Information Retrieval, Text Mining often requires lemmatization
  - Semantics is more important (we need to distinguish an *army* and an *arm*!)
  - Errors in low-level components can multiply when running downstream

Lexical Analysis 3: Morphological Analysis

- Stemming vs. Lemmatization

Word	Stemming	Lemmatization
Love	Lov	Love
Loves	Lov	Love
Loved	Lov	Love
Loving	Lov	Love
Innovation	Innovat	Innovation
Innovations	Innovat	Innovation
Innovate	Innovat	Innovate
Innovates	Innovat	Innovate
Innovative	Innovat	Innovative

AGENDA

01 Theory 1: Text Preprocessing

02 Theory 2: Text Representation

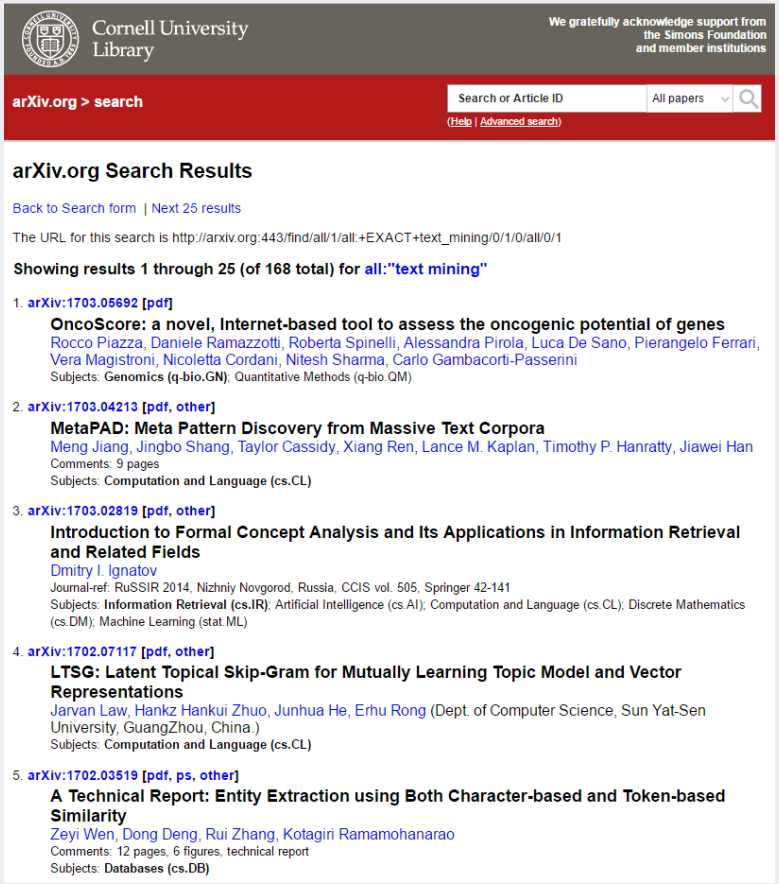
03 Theory 3: Text Summarization

04 Case Study 1: Obama's present vs. past

05 Case Study 2: Obama vs. Romney

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

Mining textual patterns in news, tweets, papers, and

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.

What We Have Done So Far…

Preprocessing with some NLP techniques

The complicated, evolving landscape of cancer

Mining textual patterns in news, tweets, papers, and

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

mine textual pattern in news tweet paper and mani

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

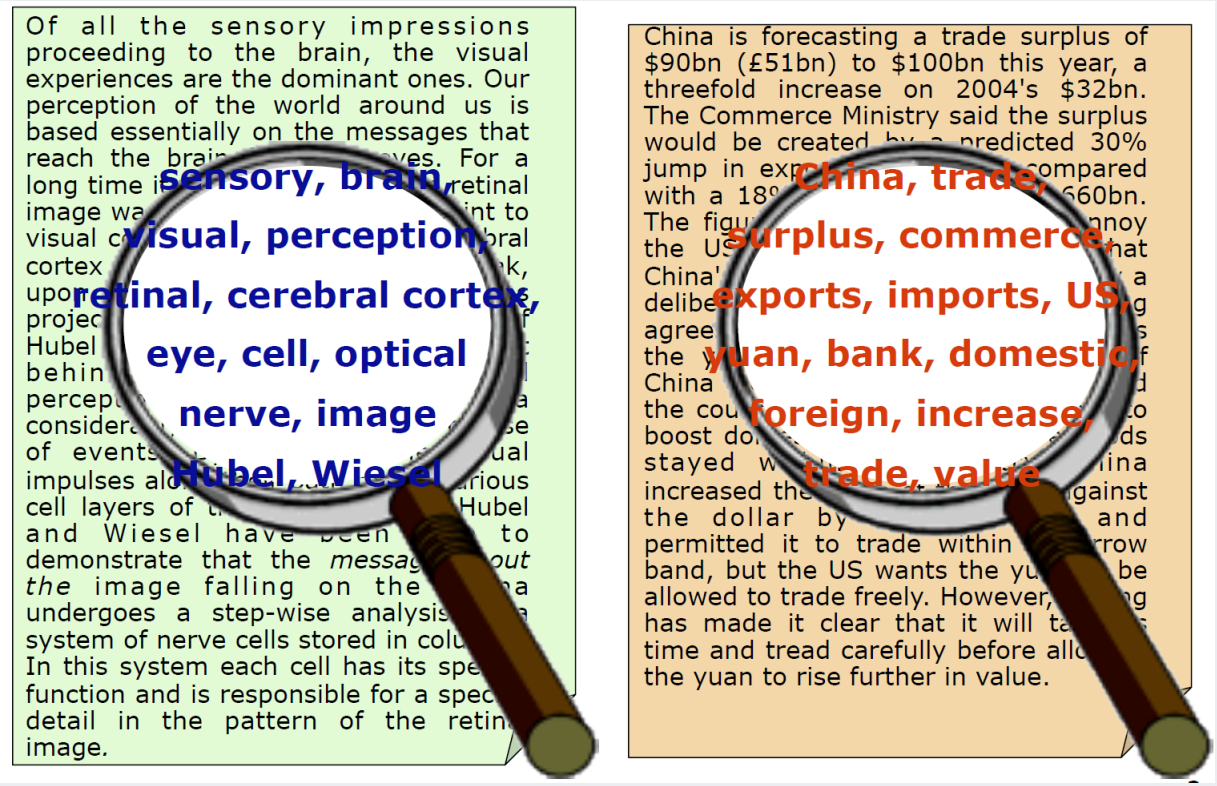
mine textual pattern in news tweet paper and mani

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					

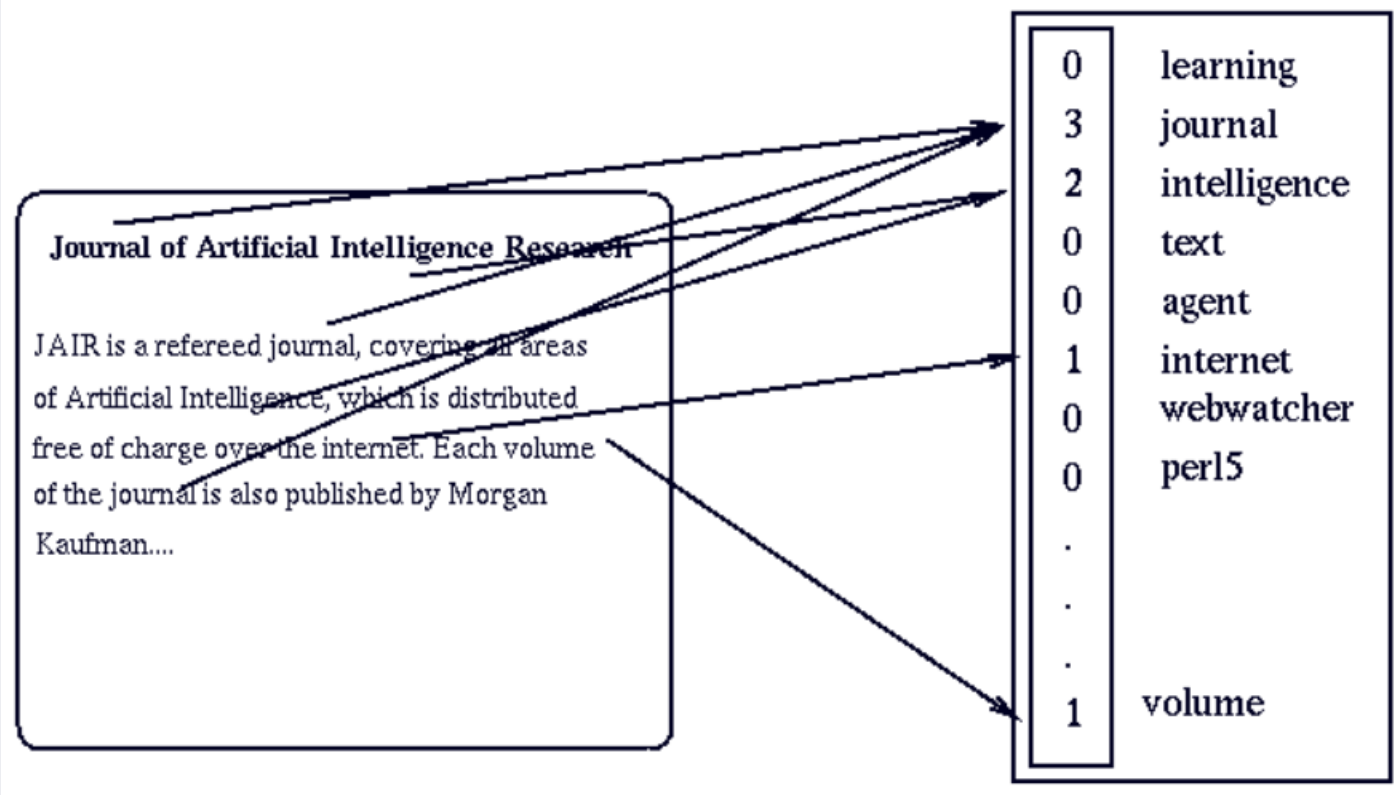
Bag of Words: Motivation

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



Bag of Words: Idea

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



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"	

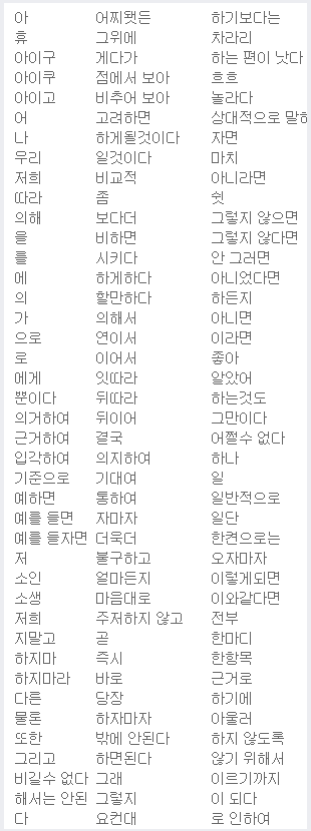
Stop Words

- 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

어찌됏든
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 낫다
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
펄렁
똥안
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
어찌됏어
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
어느해
어느 년도
라 해도
언젠가
어떤것
어느것
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

...

Word Weighting: Term-Frequency (TF)

Nayak & Raghavan (2014)

- Term frequency $tf_{t,d}$
  - ✓ 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%>

<산공 강의 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
  - ✓ $idf_t = \log_{10}(N/df_t)$
  - ✓ We use $\log(N/df_t)$ instead of N/df_t to "dampen" the effect of idf
- IDF example with N = 1 million

term	df_t	idf_t
calpurnia	1	6
animal	100	5
sunday	1,000	4
fly	10,000	3
under	100,000	2
the	1,000,000	1

Word Weighting: TF-IDF

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

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

More important if the term occur more frequently in a document

More important if the term occur less frequently in the other document

- ✓ 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 $R^{|V|}$

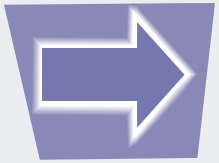
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0.35
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

  - ✓ 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

AGENDA

01 Theory 1: Text Preprocessing

02 Theory 2: Text Representation

03 Theory 3: Text Summarization

04 Case Study 1: Obama's present vs. past

05 Case Study 2: Obama vs. Romney

Wordcloud (Tagcloud)

- Wordcloud
  - ✓ A tool used to visually show the popularity of words in a collection of documents and how often they have been used
  - ✓ Conceptually resemble histograms, but can represent more items
- The way it works
  - ✓ The more a word is presented, the larger it will appear within the cloud
  - ✓ Words that are similar appear next to each other in the cloud
- Various algorithms/designs exist

http://www.edudemic.com/9-word-cloud-generators-that-arent-wordle/

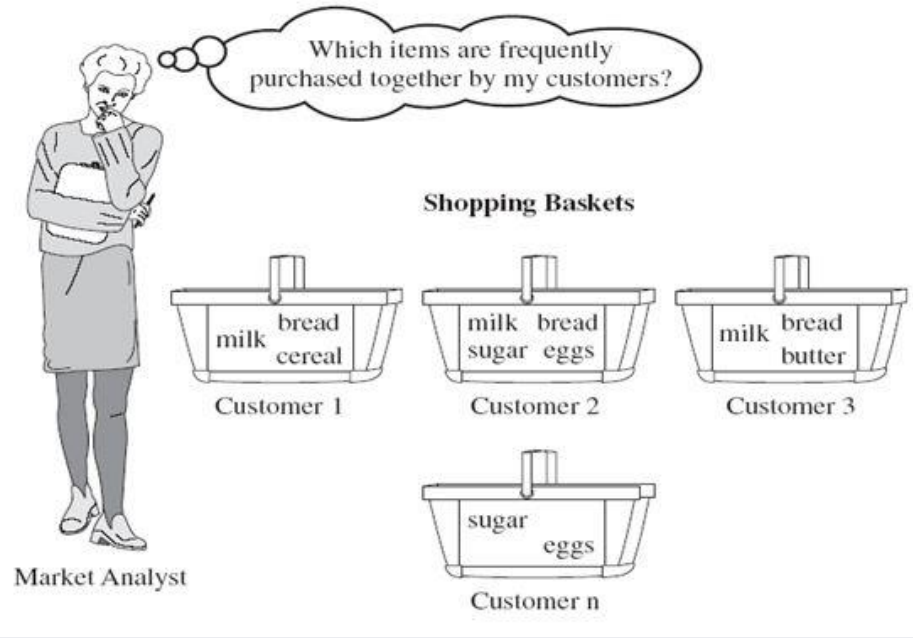
Wordcloud

- Creation of wordcloud
  - ✓ The font size of a word in a wordcloud is determined by its incidence.
  - ✓ For smaller frequencies, one can specify font size directly, from one to whatever the maximum font size.
  - ✓ For larger values, a scaling should be made
  - ✓ An example of font size computation

$$font\ size\ (w(i)) = \frac{freq(w(i)) - min.freq(w, D)}{max.freq(w, D) - min.freq(w, D)} + constant$$

Association rules

- Goal:
  - ✓ Produce rules that define "what goes with what"
  - ✓ "If X was purchased, then Y was also purchased" (in Market Basket Analysis; MBA)
  - ✓ "If a word X is presented in a sentence (phrase), then a word Y is also presented in the same sentence (phrase)" (in Text Mining; TM)




```
> inspect(ares.pruned)
    lhs                rhs          support confidence      lift
1   {기적}          => {한강}    0.04494382  1.0000000 22.250000
2   {기술}          => {과학}    0.04494382  1.0000000 17.800000
3   {경제,
     기술}          => {창조}    0.03370787  1.0000000 11.125000
4   {경제,
     과학}          => {창조}    0.04494382  1.0000000 11.125000
5   {국민,
     신뢰}          => {정부}    0.03370787  1.0000000 11.125000
6   {융성}          => {문화}    0.03370787  1.0000000  8.900000
7   {과학}          => {창조}    0.04494382  0.8000000  8.900000
8   {민국}          => {대한}    0.13483146  1.0000000  7.416667
9   {창조}          => {경제}    0.08988764  1.0000000  6.846154
10  {오늘}          => {민국}    0.04494382  0.8000000  5.933333
11  {오늘}          => {대한}    0.04494382  0.8000000  5.933333
12  {과학}          => {경제}    0.04494382  0.8000000  5.476923
13  {존경}          => {여러분}  0.04494382  0.8000000  5.085714
14  {경제부흥,
     국민}          => {행복}    0.03370787  1.0000000  4.944444
15  {여러분,
     희망}          => {시대}    0.03370787  1.0000000  4.944444
```

Association rules

- Features
  - ✓ Rows are transactions (in MBA) or sentences/phrase (in TM)
  - ✓ A Synthetic Example
    - 6 keywords in 10 sentences

Sentence	Word 1	Word 2	Word 3	Word 4
S1	Love	Movie	Football	
S2	Movie	Watch		
S3	Movie	Sleep		
S4	Love	Movie	Watch	
S5	Love	Sleep		
S6	Movie	Sleep		
S7	Movie	Watch		
S8	Love	Movie	Sleep	Football
S9	Love	Movie	Sleep	
S10	Party			

Association rules

- Terminology
  - ✓ Antecedent – "IF" part
  - ✓ Consequent – "THEN" part
  - ✓ Item set – the items comprising the antecedent or consequent
  - ✓ Antecedent and consequent are disjoint (have no items in common)

- Generating rules
  - ✓ Many rules are possible (e.g., for sentence 1)
    - If Love is presented, then Movie is also presented.
    - If Love and Movie are presented, then Football is also presented.
    - If Football is presented, then Love is also presented.
    - etc.

Association rules: Performance measures

For the rule A → B

- Support

$$Support\ (A) = P(A)\ or\ Support\ (A \rightarrow B) = P(A, B)$$

  ✓ Used to find the frequent item sets

- Confidence

$$Confidence\ (A \rightarrow B) = \frac{P(A, B)}{P(A)}$$

  ✓ Used to generate meaningful rules

- Lift

$$Lift\ (A \rightarrow B) = \frac{P(A, B)}{P(A) \times P(B)}$$

  ✓ Used to evaluate the statistical significance of the generated rules

  - If lift = 1, then the antecedent and the consequents are statistically independent
  - If lift > 1, then the rule is useful in finding consequent item sets

Association rules

- How to generate effective association rules?
  - ✓ Ideally, create all possible combinations of items and see what rules are effective and what rules are not.
  - ✓ Computation time grows exponentially as the number of items increases.

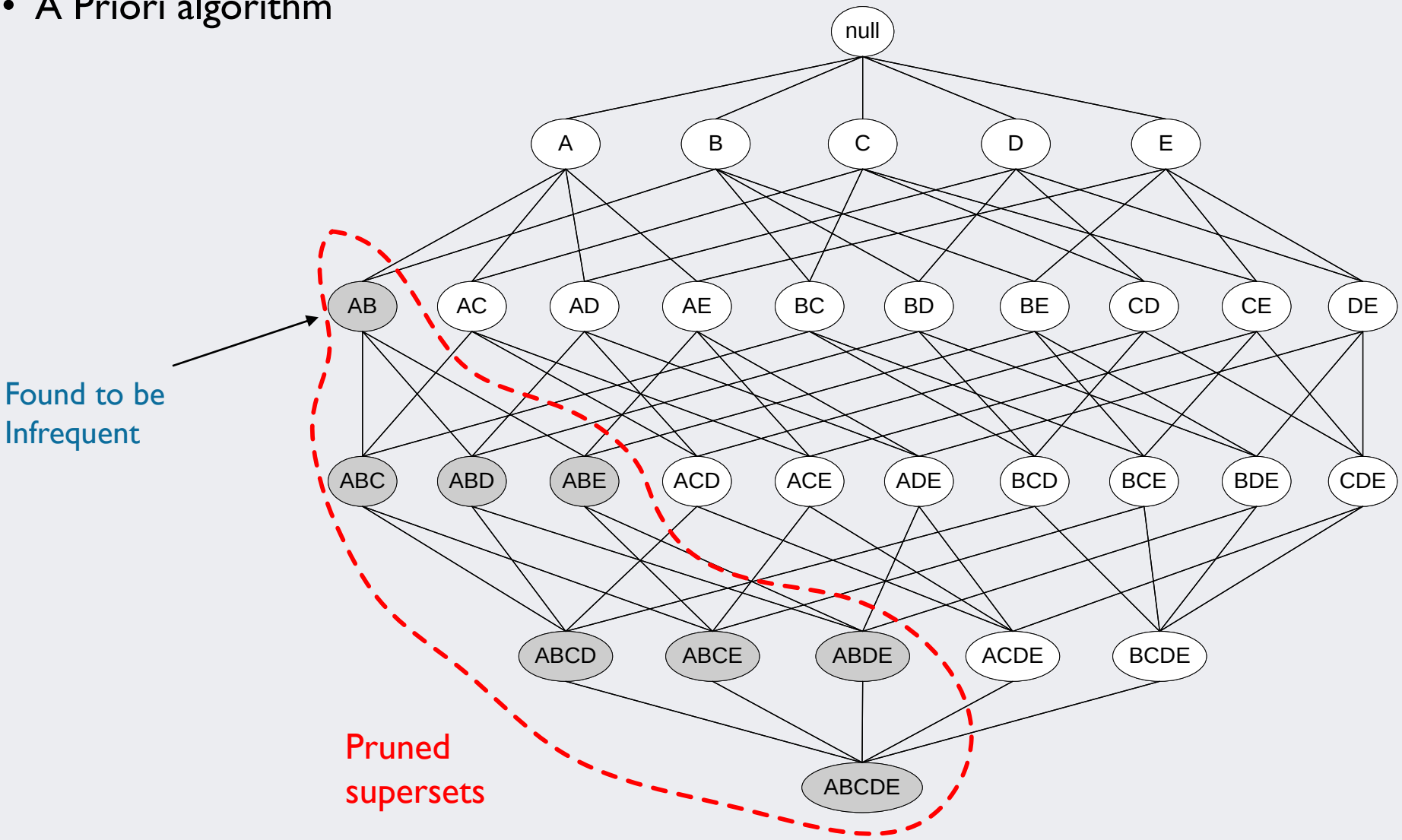
- Brute-force approach
  - ✓ List all possible association rules
  - ✓ Compute the support and confidence for each rule
  - ✓ Prune rules that fail the minimum support and minimum confidence threshold
  - ✓ Computationally prohibitive!

Association rules

- A priori algorithm
  - ✓ Consider only "frequent item sets"
  - ✓ Support
    - Criterion for item set frequency P(A)
    - #(%) of sentences that include the antecedent (both the antecedent and the consequent)
    - Support for the item set {Love, Movie} is 4 out of 10 sentences, or 40%
  - ✓ Support of an itemset never exceeds the support of its subsets, which is known as anti-monotone property of support.

Association rules

- A Priori algorithm



Association Rules: Generating Frequent Item Sets

I

- Set a minimum support criterion

  - Set the minimum support to 2 sentences or 20%

Sentence	Word 1	Word 2	Word 3	Word 4
S1	Love	Movie	Football	
S2	Movie	Watch		
S3	Movie	Sleep		
S4	Love	Movie	Watch	
S5	Love	Sleep		
S6	Movie	Sleep		
S7	Movie	Watch		
S8	Love	Movie	Sleep	Football
S9	Love	Movie	Sleep	
S10	Party			

Association Rules: Generating Frequent Item Sets

2

- Generate the list of one-item sets that meets the support criterion

- Support {Movie} = 8/10 = 80%
- Support {Love} = 5/10 = 50%
- Support {Sleep} = 5/10 = 50%
- Support {Watch} = 3/10 = 30%
- Support {Football} = 2/10 = 20%
- Support {Party} = 1/10 = 10%

Party is removed because it does not meet the minimum support criterion

Association Rules: Generating Frequent Item Sets

3

- Use the life of one-item sets to generate list of two-item sets that meet the support criterion

	Movie	Love	Sleep	Watch	Football
Movie		**40%**	**40%**	**20%**	**20%**
Love			**30%**	0%	**20%**
Sleep				0%	10%
Watch					0%
Football					

- Among the 10 possible item sets, six of them are still found to be frequent

Association Rules: Generating Frequent Item Sets

- Use the list of two-item sets to generate the three-item sets.
- Continue up through k-item sets.

Set-size	Word 1	Word 2	Word 3	..	Word 6
1	Movie				
1	Love				
1	Sleep				
1	Watch				
1	Football				
2	Movie	Love			
2	Movie	Sleep			
2	Movie	Watch			
…	…	…			

4

Association Rules

- A priori algorithm
  - ✓ Let k=1
  - ✓ Generate frequent itemsets of length 1
  - ✓ Repeat until no new frequent itemsets are identified
    - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Prune candidate itemsets containing subsets of length k that are infrequent
    - Count the support of each candidate by scanning the DB
    - Eliminate candidates that are infrequent, leaving only those that are frequent

Association Rules: Result

- Generated Rules

Rule: If all Antecedent items are purchased, then with Confidence percentage Consequent items will also be purchased.

Row ID	Confidence %	Antecedent (A)	Consequent (C)	Support for A	Support for C	Support for A & C	Lift Ratio
6	100	Football	Love & Movie	2	4	2	2.5
2	100	Football	Love	2	5	2	2
4	100	Movie & Football	Love	2	5	2	2
3	100	Football	Movie	2	8	2	1.25
5	100	Love & Football	Movie	2	8	2	1.25
7	100	Watch	Movie	3	8	3	1.25
1	80	Love	Movie	5	8	4	1
8	80	Sleep	Movie	5	8	4	1

✓ Interpretation

- Support for A & C = 2 → There are two sentences that the words Football, Love, and Movie are presented together.
- 100% Confidence → If Football is presented in a sentence, then it is always that Love and Movie are also presented.
- Lift Ratio 2.5 → The association between the two item sets are 2.5 stronger than when they are assumed to be statistically independent.

AGENDA

01 Theory 1: Text Preprocessing

02 Theory 2: Text Representation

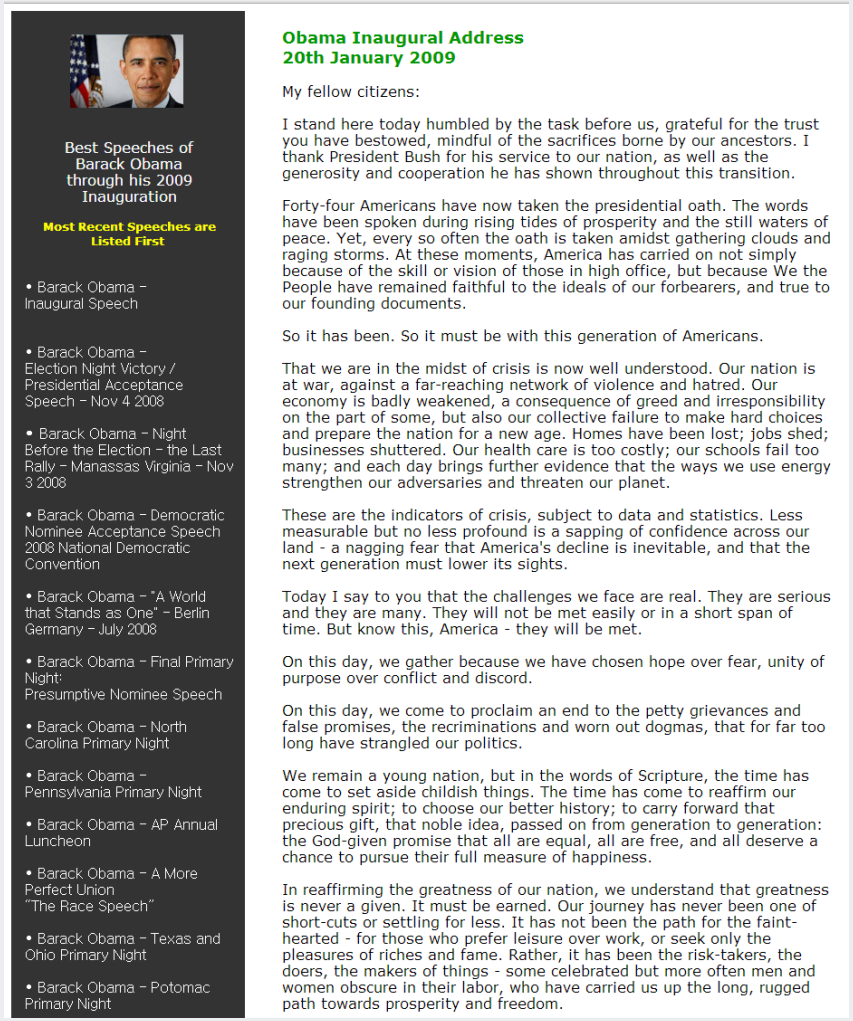
03 Theory 3: Text Summarization

04 Case Study 1: Obama's present vs. past

05 Case Study 2: Obama vs. Romney

Obama vs. Obama

- http://www.obamaspeeches.com/



Best Speeches of Barack Obama through his 2009 Inauguration

Most Recent Speeches are Listed First

- Barack Obama – Inaugural Speech
- Barack Obama – Election Night Victory / Presidential Acceptance Speech – Nov 4 2008
- Barack Obama – Night Before the Election – the Last Rally – Manassas Virginia – Nov 3 2008
- Barack Obama – Democratic Nominee Acceptance Speech 2008 National Democratic Convention
- Barack Obama – "A World that Stands as One" – Berlin Germany – July 2008
- Barack Obama – Final Primary Night: Presumptive Nominee Speech
- Barack Obama – North Carolina Primary Night
- Barack Obama – Pennsylvania Primary Night
- Barack Obama – AP Annual Luncheon
- Barack Obama – A More Perfect Union "The Race Speech"
- Barack Obama – Texas and Ohio Primary Night
- Barack Obama – Potomac Primary Night

Obama Inaugural Address
20th January 2009

My fellow citizens:

I stand here today humbled by the task before us, grateful for the trust you have bestowed, mindful of the sacrifices borne by our ancestors. I thank President Bush for his service to our nation, as well as the generosity and cooperation he has shown throughout this transition.

Forty-four Americans have now taken the presidential oath. The words have been spoken during rising tides of prosperity and the still waters of peace. Yet, every so often the oath is taken amidst gathering clouds and raging storms. At these moments, America has carried on not simply because of the skill or vision of those in high office, but because We the People have remained faithful to the ideals of our forbearers, and true to our founding documents.

So it has been. So it must be with this generation of Americans.

That we are in the midst of crisis is now well understood. Our nation is at war, against a far-reaching network of violence and hatred. Our economy is badly weakened, a consequence of greed and irresponsibility on the part of some, but also our collective failure to make hard choices and prepare the nation for a new age. Homes have been lost; jobs shed; businesses shuttered. Our health care is too costly; our schools fail too many; and each day brings further evidence that the ways we use energy strengthen our adversaries and threaten our planet.

These are the indicators of crisis, subject to data and statistics. Less measurable but no less profound is a sapping of confidence across our land - a nagging fear that America's decline is inevitable, and that the next generation must lower its sights.

Today I say to you that the challenges we face are real. They are serious and they are many. They will not be met easily or in a short span of time. But know this, America - they will be met.

On this day, we gather because we have chosen hope over fear, unity of purpose over conflict and discord.

On this day, we come to proclaim an end to the petty grievances and false promises, the recriminations and worn out dogmas, that for far too long have strangled our politics.

We remain a young nation, but in the words of Scripture, the time has come to set aside childish things. The time has come to reaffirm our enduring spirit; to choose our better history; to carry forward that precious gift, that noble idea, passed on from generation to generation: the God-given promise that all are equal, all are free, and all deserve a chance to pursue their full measure of happiness.

In reaffirming the greatness of our nation, we understand that greatness is never a given. It must be earned. Our journey has never been one of short-cuts or settling for less. It has not been the path for the faint-hearted - for those who prefer leisure over work, or seek only the pleasures of riches and fame. Rather, it has been the risk-takers, the doers, the makers of things - some celebrated but more often men and women obscure in their labor, who have carried us up the long, rugged path towards prosperity and freedom.

Obama in 2009

Obama in 2002

Obama vs. Obama

- Install necessary packages

```
# Install necessary packages
install.packages("tm", dependencies = TRUE)
install.packages("wordcloud", dependencies = TRUE)
install.packages("plyr", dependencies = TRUE)
install.packages("igraph", dependencies = TRUE)

library(tm)
library(wordcloud)
library(plyr)
library(igraph)
```

✓ Package "tm": provides basic functions for text mining

✓ Package "wordcloud" & "igraph": help to draw wordclouds and keywords network

Obama vs. Obama

- Install necessary packages

```
# Load the dataset
ObamaSpeech <- read.csv("speechBylineTable_utf8.csv", stringsAsFactors = FALSE)
```

✓ stringsAsFactors: strings are stored as factors without this option

	X	idx	line	contents
1	1	1	1	Obama Inaugural Address 20th January 2009My fellow c...
2	2	1	2	I thank President Bush for his service to our nation, as ...
3	3	1	3	Forty-four Americans have now taken the presidential o...
4	4	1	4	The words have been spoken during rising tides of pros...
5	5	1	5	Yet, every so often the oath is taken amidst gathering cl...
6	6	1	6	At these moments, America has carried on not simply be...
7	7	1	7	So it has been.
8	8	1	8	So it must be with this generation of Americans.
9	9	1	9	That we are in the midst of crisis is now well understood.
10	10	1	10	Our nation is at war, against a far-reaching network of v...
11	11	1	11	Our economy is badly weakened, a consequence of gree...
12	12	1	12	Homes have been lost; jobs shed; businesses shuttered.
13	13	1	13	Our health care is too costly; our schools fail too many; ...
14	14	1	14	These are the indicators of crisis, subject to data and sta...
15	15	1	15	Less measurable but no less profound is a sapping of co...

Obama vs. Obama

- Select the most recent 5 and past 5 speeches

```
# Select the first five speeches
idx1 <- which(ObamaSpeech$idx <= 5)
Speech1 <-as.data.frame(ObamaSpeech$contents[idx1])
names(Speech1) <- "sentence"

# Select the last five speeches
idx2 <- which(ObamaSpeech$idx > 97)
Speech2 <-as.data.frame(ObamaSpeech$contents[idx2])
names(Speech2) <- "sentence"
```

Data	
ObamaSpeech	8596 obs. of 4 variables
Speech1	628 obs. of 1 variable
Speech2	380 obs. of 1 variable
Values	
idx1	int [1:628] 1 2 3 4 5 6 7 8 9 10 ...
idx2	int [1:380] 8217 8218 8219 8220 8221 8222 8223 8224 8225 …

Obama vs. Obama

- Construct a corpus

```
# Construct a corpus # VectorSource specifies that the source is character vectors.
myCorpus1 <- Corpus(VectorSource(Speech1$sentence))
myCorpus1[[1]][1]

myCorpus2 <- Corpus(VectorSource(Speech2$sentence))
myCorpus2[[1]][1]
```

```
> myCorpus1[[1]][1]
$content
[1] "Obama Inaugural Address 20th January 2009My fellow citizens:I stand here today humbled by the ta
sk before us, grateful for the trust you have bestowed, mindful of the sacrifices borne by our ancest
ors."
```

```
> myCorpus2[[1]][1]
$content
[1] "A<U+00A0> REMARKS OF SENATOR OBAMA AT TECHNET Tuesday, March 8, 2005Remarks at TechNetComplete T
ext  Thank you John Doer."
```

Obama vs. Obama

- Data preprocessing 1: To lower case

```
# Data preprocessing # 1: to lower case
myCorpus1 <- tm_map(myCorpus1, content_transformer(tolower))
myCorpus1[[1]][1]

myCorpus2 <- tm_map(myCorpus2, content_transformer(tolower))
myCorpus2[[1]][1]
```

```
> myCorpus1[[1]][1]
$content
[1] "obama inaugural address 20th january 2009my fellow citizens:i stand here today humbled by the ta
sk before us, grateful for the trust you have bestowed, mindful of the sacrifices borne by our ancest
ors."
```

```
> myCorpus2[[1]][1]
$content
[1] "a<u+00a0> remarks of senator obama at technet tuesday, march 8, 2005remarks at technetcomplete t
ext  thank you john doer."
```

Obama vs. Obama

- Data preprocessing 2: Remove punctuation

```
# 2: remove puntuations
myCorpus1 <- tm_map(myCorpus1, content_transformer(removePunctuation))
myCorpus1[[1]][1]

myCorpus2 <- tm_map(myCorpus2, content_transformer(removePunctuation))
myCorpus2[[1]][1]
```

```
> myCorpus1 <- tm_map(myCorpus1, content_transformer(removePunctuation))
> myCorpus1[[1]][1]
$content
[1] "obama inaugural address 20th january 2009my fellow citizensi stand here today humbled by the tas
k before us grateful for the trust you have bestowed mindful of the sacrifices borne by our ancestors
"
```

```
> myCorpus2 <- tm_map(myCorpus2, content_transformer(removePunctuation))
> myCorpus2[[1]][1]
$content
[1] "au00a0 remarks of senator obama at technet tuesday march 8 2005remarks at technetcomplete text
thank you john doer"
```

Obama vs. Obama

- Data preprocessing 3: Remove numbers

```
# 3. remove numbers
myCorpus1 <- tm_map(myCorpus1, content_transformer(removeNumbers))
myCorpus1[[1]][1]

myCorpus2 <- tm_map(myCorpus2, content_transformer(removeNumbers))
myCorpus2[[1]][1]
```

```
> myCorpus1 <- tm_map(myCorpus1, content_transformer(removeNumbers))
> myCorpus1[[1]][1]
$content
[1] "obama inaugural address th january my fellow citizensi stand here today humbled by the task befo
re us grateful for the trust you have bestowed mindful of the sacrifices borne by our ancestors"
```

```
> myCorpus2 <- tm_map(myCorpus2, content_transformer(removeNumbers))
> myCorpus2[[1]][1]
$content
[1] "aua remarks of senator obama at technet tuesday march  remarks at technetcomplete text  thank yo
u john doer"
```

Obama vs. Obama

- Data preprocessing 4: Remove stopwords

```
# 4. remove stopwords (SMART stopwords list)
# Add "obama" to the stopwords list
myStopwords <- c(stopwords("SMART"), "obama")

myCorpus1 <- tm_map(myCorpus1, removeWords, myStopwords)
myCorpus1[[1]][1]

myCorpus2 <- tm_map(myCorpus2, removeWords, myStopwords)
myCorpus2[[1]][1]
```

```
> myCorpus1 <- tm_map(myCorpus1, removeWords, myStopwords)
> myCorpus1[[1]][1]
$content
[1] " inaugural address  january  fellow citizensi stand  today humbled   task   grateful   trust   b
estowed mindful   sacrifices borne   ancestors"
```

```
> myCorpus2 <- tm_map(myCorpus2, removeWords, myStopwords)
> myCorpus2[[1]][1]
$content
[1] "aua remarks  senator   technet tuesday march  remarks  technetcomplete text    john doer"
```

Obama vs. Obama

- Data preprocessing 5: Stemming

```
# 5. Stemming
stemCorpus1 <- tm_map(myCorpus1, stemDocument)
myCorpus1[[1]][1]

stemCorpus2 <- tm_map(myCorpus2, stemDocument)
myCorpus2[[1]][1]
```

```
> stemCorpus1 <- tm_map(myCorpus1, stemDocument)
> myCorpus1[[1]][1]
$content
[1] " inaugural address  january  fellow citizensi stand  today humbled   task   grateful   trust   b
estowed mindful   sacrifices borne   ancestors"
```

```
> stemCorpus2 <- tm_map(myCorpus2, stemDocument)
> myCorpus2[[1]][1]
$content
[1] "aua remarks  senator   technet tuesday march  remarks  technetcomplete text    john doer"
```

Obama vs. Obama

- Construct Term-Document matrix

```
# Construct Term-Document Matrix
myTDM1 <- TermDocumentMatrix(stemCorpus1, control = list(minWordLength = 1))
myTDM2 <- TermDocumentMatrix(stemCorpus2, control = list(minWordLength = 1))

# Check the Term-Document Matrix
myTDM1
myTDM2

as.matrix(myTDM1)[11:30,11:30]
as.matrix(myTDM2)[11:30,11:30]
```

```
> myTDM1
<<TermDocumentMatrix (terms: 1621, documents: 628)>>
Non-/sparse entries: 4967/1013021
Sparsity           : 100%
Maximal term length: 16
Weighting          : term frequency (tf)
> myTDM2
<<TermDocumentMatrix (terms: 1318, documents: 380)>>
Non-/sparse entries: 3055/497785
Sparsity           : 99%
Maximal term length: 16
Weighting          : term frequency (tf)
```

Obama vs. Obama

- Construct Term-Document matrix

```
> as.matrix(myTDM1)[11:30,11:30]
           Docs
Terms       11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
  mind       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  sacrific   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  stand      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  task       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  today      0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  trust      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  bush       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  cooper     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  generos    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  nation     1  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0
  presid     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  servic     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  shown      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  transit    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  american   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  fortyfour  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  oath       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  presidenti 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  peac       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  prosper    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
```

```
> as.matrix(myTDM2)[11:30,11:30]
           Docs
Terms       11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
  children   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  discuss    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  futur      0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
  great      0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  honor      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  join       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  opportun   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  play       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  role       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  technolog  0  0  0  0  1  0  1  0  0  1  0  0  0  1  0  0  1  0  0  0
  today      0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
  youv       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  begin      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  ceo        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  countri    0  0  0  0  2  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0
  inspir     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  leadership 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  leav       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  organ      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  recogn     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Obama vs. Obama

- Find high frequency words (cut-off: 15)

```
# High frequent words
findFreqTerms(myTDM1, lowfreq=15)
findFreqTerms(myTDM2, lowfreq=15)
```

```
> findFreqTerms(myTDM1, lowfreq=15)
 [1] "stand"      "nation"     "presid"     "american"   "america"    "moment"     "peopl"
 [8] "generat"    "economi"    "make"       "job"        "care"       "day"        "face"
[15] "time"       "hope"       "end"        "long"       "polit"      "promis"     "work"
[22] "men"        "women"      "world"      "live"       "year"       "state"      "countri"
[29] "afford"     "famili"     "govern"     "give"       "futur"      "readi"      "back"
[36] "unit"       "chang"      "children"   "turn"       "elect"      "tonight"    "democrat"
[43] "mccain"     "campaign"   "washington" "polici"     "parti"      "dont"       "john"
[50] "tax"
> findFreqTerms(myTDM2, lowfreq=15)
 [1] "john"       "children"   "countri"    "year"       "famili"     "job"        "theyr"
 [8] "state"      "american"   "work"       "america"    "dream"      "hope"       "school"
[15] "peopl"      "bill"       "make"       "bankruptci" "compani"    "war"
```

Obama vs. Obama

- Highly associated words with "freedom" and "american" (cut-off: 15)

```
# Which words are associated with "freedom"?
findAssocs(myTDM1, 'freedom', 0.35)
findAssocs(myTDM2, 'freedom', 0.35)

# Which words are associated with "american"?
findAssocs(myTDM1, 'america', 0.3)
findAssocs(myTDM2, 'america', 0.3)
```

```
> findAssocs(myTDM1, 'freedom', 0.35)
$freedom
     doer     labor     maker    obscur   risktak       rug   control     favor      spin
     0.41      0.41      0.41      0.41      0.41      0.41      0.41      0.41      0.41
  unmatch    falter   horizon      test promiseit     treat     yearn      ceil     glass
     0.41      0.41      0.41      0.41      0.41      0.41      0.41      0.41      0.41
  shatter       eye     carri
     0.41      0.40      0.36

> findAssocs(myTDM2, 'freedom', 0.35)
$freedom
     banker       belov    shopkeep      teenag      beacon       magic    persever scholarship
       0.41        0.41        0.41        0.41        0.41        0.41        0.41        0.41
       envi        wedg        dare        defi       delta     distant        fire       funni
       0.41        0.41        0.41        0.41        0.41        0.41        0.41        0.41
     mekong    millwork       naval         odd      patrol       shore      skinni       slave
       0.41        0.41        0.41        0.41        0.41        0.41        0.41        0.41
    arsenal   democraci        evil     triumph      defens     lifetim       occas
       0.41        0.41        0.41        0.41        0.41        0.41        0.41
```

Obama vs. Obama

- Highly associated words with "freedom" and "american" (cut-off: 15)

```
# Which words are associated with "freedom"?
findAssocs(myTDM1, 'freedom', 0.35)
findAssocs(myTDM2, 'freedom', 0.35)

# Which words are associated with "american"?
findAssocs(myTDM1, 'america', 0.3)
findAssocs(myTDM2, 'america', 0.3)
```

```
> findAssocs(myTDM1, 'america', 0.3)
$america
blue  red unit serv
0.43 0.43 0.38 0.34

> findAssocs(myTDM2, 'america', 0.3)
$america
  asian  latino    unit conserv   liber
   0.65    0.65    0.46    0.39    0.39
```

Obama vs. Obama

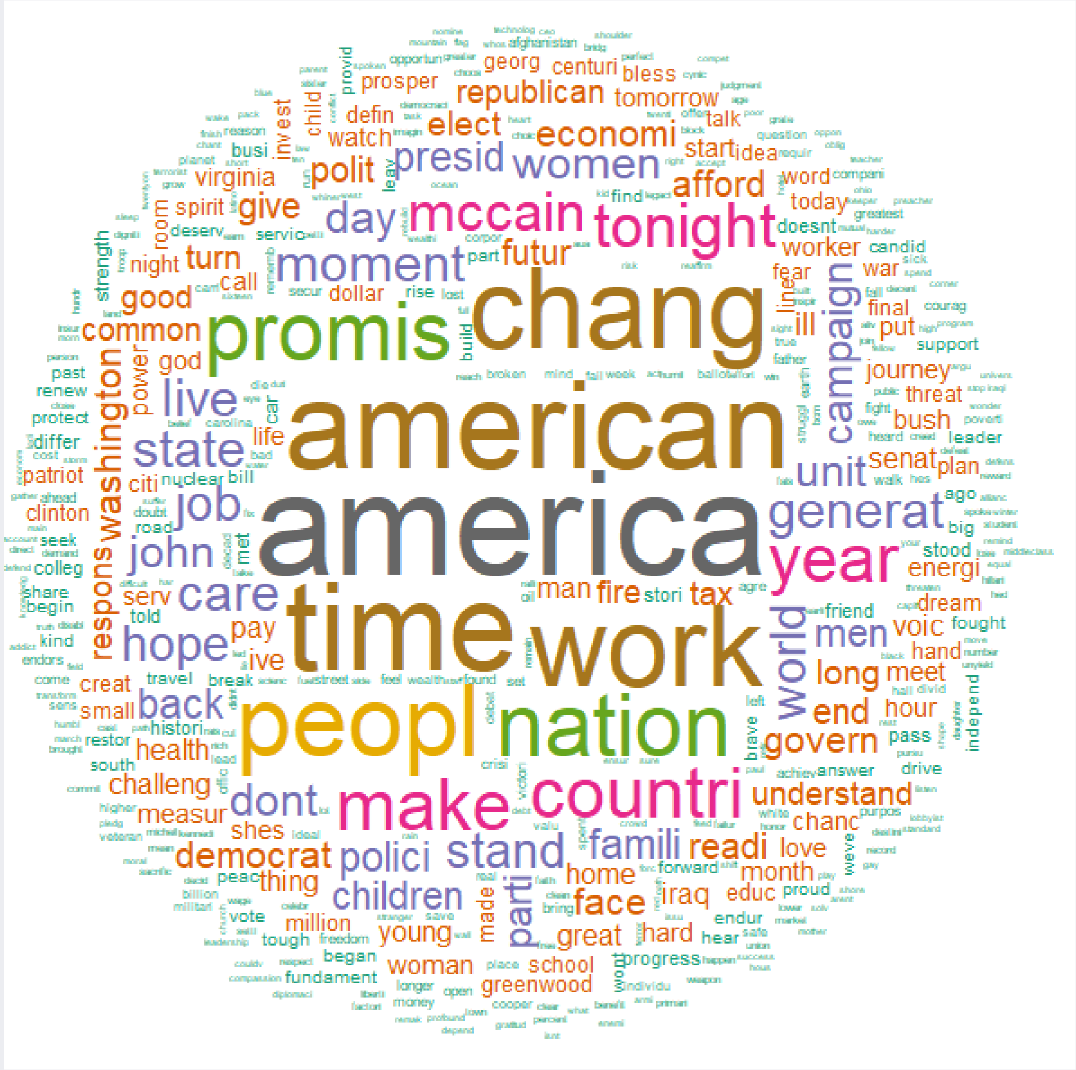
- Construct the wordcloud with the first five speeches

```
# Construct a Word Cloud for the first 5 speeches
wcmat1 <- as.matrix(myTDM1)
# calculate the frequency of words
word_freq1 <- sort(rowSums(wcmat1), decreasing=TRUE)
myNames1 <- names(word_freq1)
wcdat1 <- data.frame(word=myNames1, freq=word_freq1)
pal <- brewer.pal(8, "Dark2")
wordcloud(wcdat1$word, wcdat1$freq, min.freq=3, scale = c(5, 0.1),
          rot.per = 0.1, col=pal, random.order=F)
```

- Function "wordcloud": draw a wordcloud with the following options
  - ✓ Arg. 1: word to be displayed
  - ✓ Arg. 2: word frequency
  - ✓ Arg. 3: minimum frequency of a word to be displayed
  - ✓ Arg. 4: relative size of the words in the wordcloud
  - ✓ Arg. 5: percentage of word to be rotated

Obama vs. Obama

- Wordcloud for the first five speeches



Obama vs. Obama

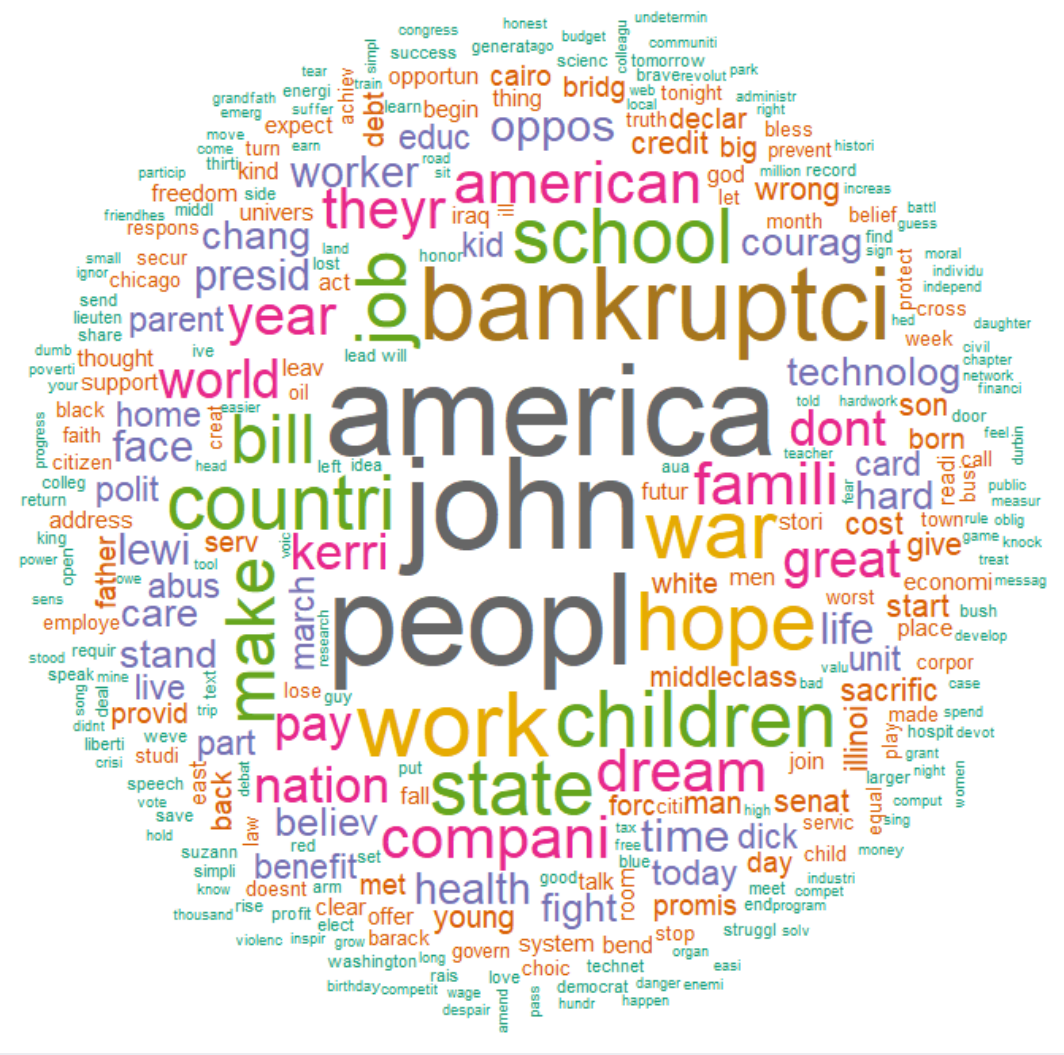
- Construct the wordcloud with the second five speeches

```
# Construct a Word Cloud for the last 5 speeches
wcmat2 <- as.matrix(myTDM2)
word_freq2 <- sort(rowSums(wcmat2), decreasing=TRUE)
myNames2 <- names(word_freq2)
wcdat2 <- data.frame(word=myNames2, freq=word_freq2)
pal <- brewer.pal(8, "Dark2")
wordcloud(wcdat2$word, wcdat2$freq, min.freq=3, scale = c(5, 0.1),
          rot.per = 0.1, col=pal, random.order=F)
```

- Function "wordcloud": draw a wordcloud with the following options
  - ✓ Arg. 1: word to be displayed
  - ✓ Arg. 2: word frequency
  - ✓ Arg. 3: minimum frequency of a word to be displayed
  - ✓ Arg. 4: relative size of the words in the wordcloud
  - ✓ Arg. 5: percentage of word to be rotated

Obama vs. Obama

- Wordcloud for the second five speeches



Obama vs. Obama

- Keyword network for the first five speeches

```
# Construct a word network for the first five speeches
# Change it to a Boolean matrix
wcmat1[wcmat1 >= 1] <- 1

# find the words that are used more than 15 times
freq_idx1 <- which(rowSums(wcmat1) > 15)
freq_wcmat1 <- wcmat1[freq_idx1,]

# Transform into a term-term adjacency matrix
termMatrix1 <- freq_wcmat1 %*% t(freq_wcmat1)

# inspect terms numbered 5 to 10
termMatrix1[1:10,5:10]
```

- ✓ Step 1: convert the word count matrix to a binary matrix
- ✓ Step 2: find the words that are used more than 15 sentences
- ✓ Step 3: transform the word count matrix to the adjacency matrix
- ✓ Step 4: check some part of the adjacency matrix

Obama vs. Obama

- Keyword network for the first five speeches
  - ✓ Step 4: Check some part of the adjacency matrix

```
> termMatrix1[1:10,5:10]
           Terms
Terms       america moment peopl generat economi make
  stand           4      0     0       0       0    0
  nation          5      5     7       2       2    3
  presid          1      0     0       0       2    2
  american        2      2    15       5       2    5
  america        62      4     4       2       0    2
  moment          4     19     2       3       1    1
  peopl           4      2    42       1       1    4
  generat         2      3     1      23       0    0
  economi         0      1     1       0      16    2
  make            2      1     4       0       2   33
```

Obama vs. Obama

- Keyword network for the first five speeches

```
# Build a graph from the above matrix
g1 <- graph.adjacency(termMatrix1, weighted=T, mode = "undirected")

# remove loops
g1 <- simplify(g1)

# set labels and degrees of vertices
V(g1)$label <- V(g1)$name
V(g1)$degree <- degree(g1)
g1 <- delete.edges(g1, which(E(g1)$weight <= 3))
```

✓ Step 6: create a graph object

✓ Step 7: remove loops in the graph

✓ Step 8: set labels and degrees of vertices

✓ Step 9: delete edges with the weight less than 3

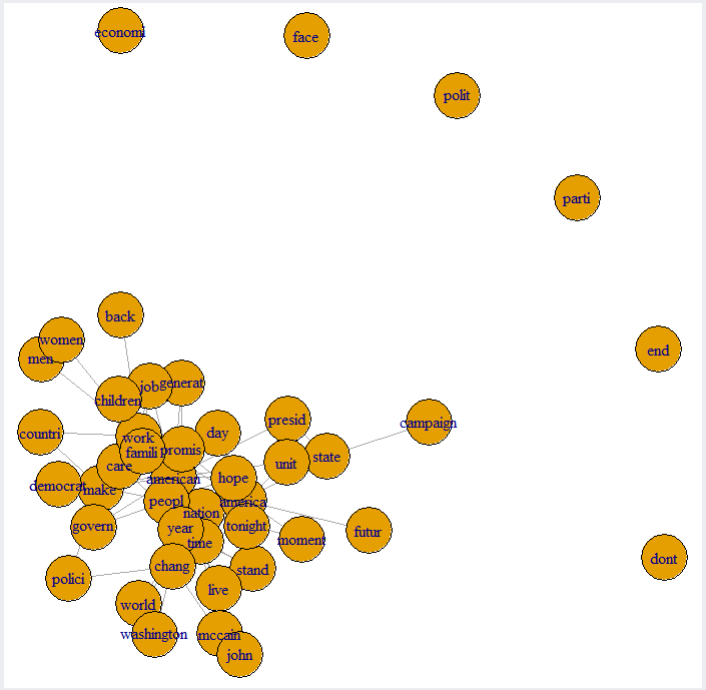
Obama vs. Obama

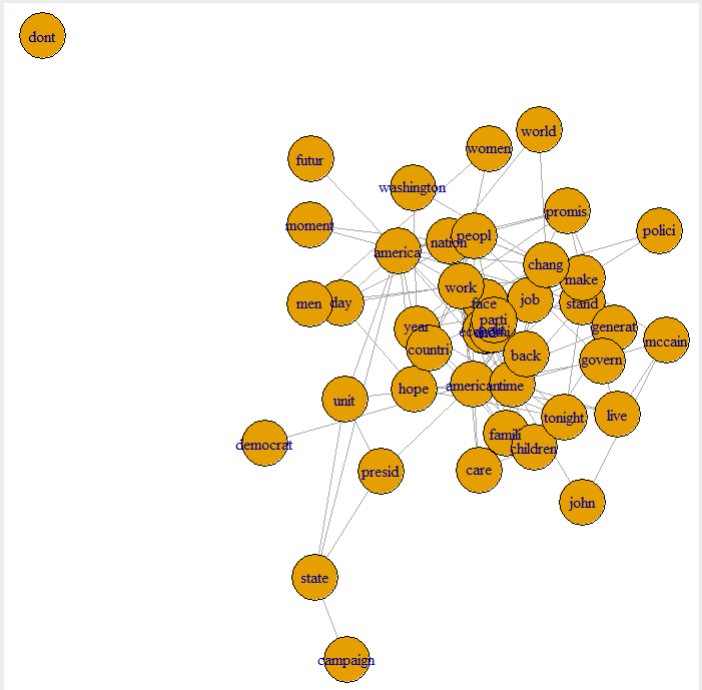
- Keyword network for the first five speeches

```
# set seed to make the layout reproducible
set.seed(3952)
layout1 <- layout.fruchterman.reingold(g1)
plot(g1, layout=layout1)

# Another layout
plot(g1, layout=layout.kamada.kawai)
```

✓ Step 10: draw networks with two different layouts





Obama vs. Obama

- Keyword network for the first five speeches

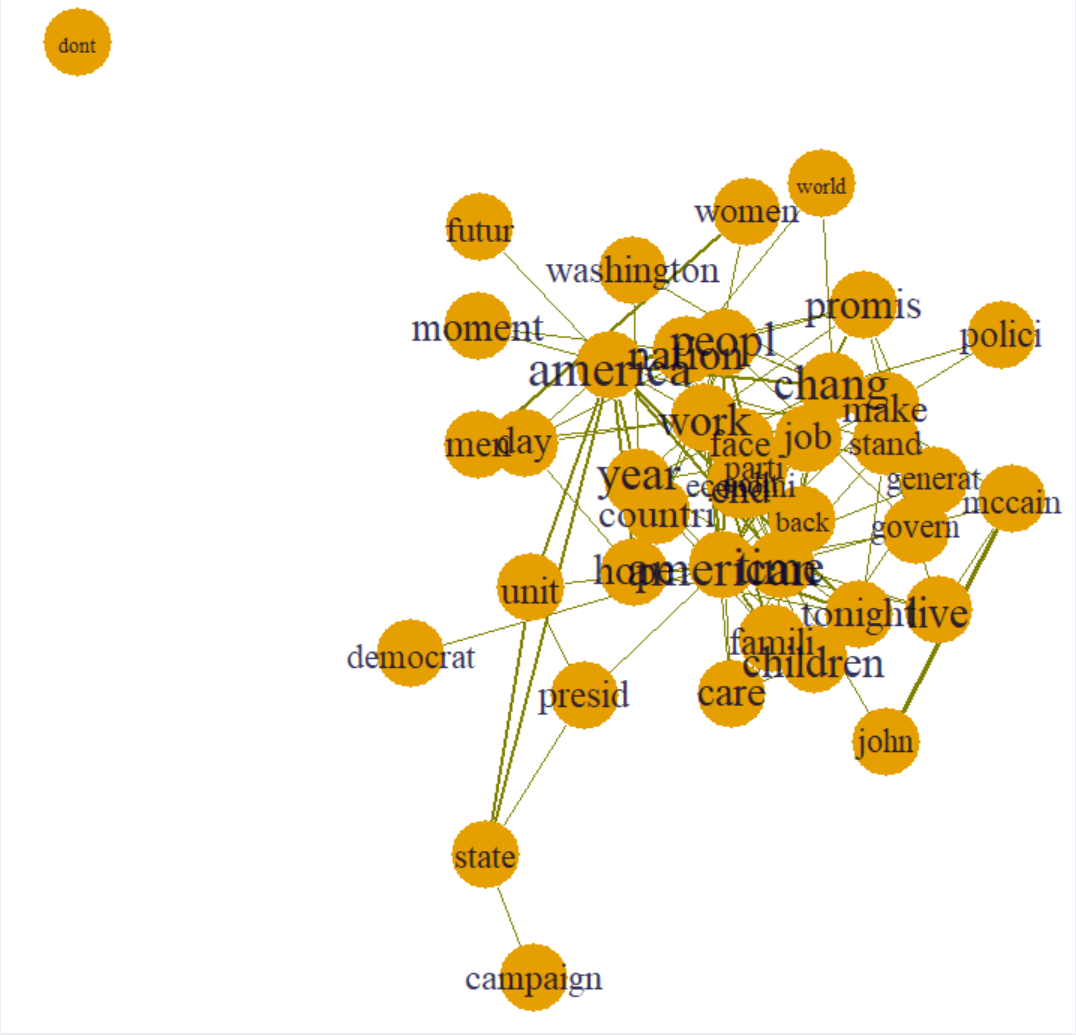
```
# Make the network look better
V(g1)$label.cex <- 2*V(g1)$degree/max(V(g1)$degree)+0.2
V(g1)$label.color <- rgb(0, 0, 0.2, 0.8)
V(g1)$frame.color <- NA
egam1 <- 3*(log(E(g1)$weight+1))/max(log(E(g1)$weight+1))
E(g1)$color <- rgb(0.5, 0.5, 0)
E(g1)$width <- egam1

# plot the graph in layout1
plot(g1, layout=layout.kamada.kawai)
```

✓ Step 11: make the network look better

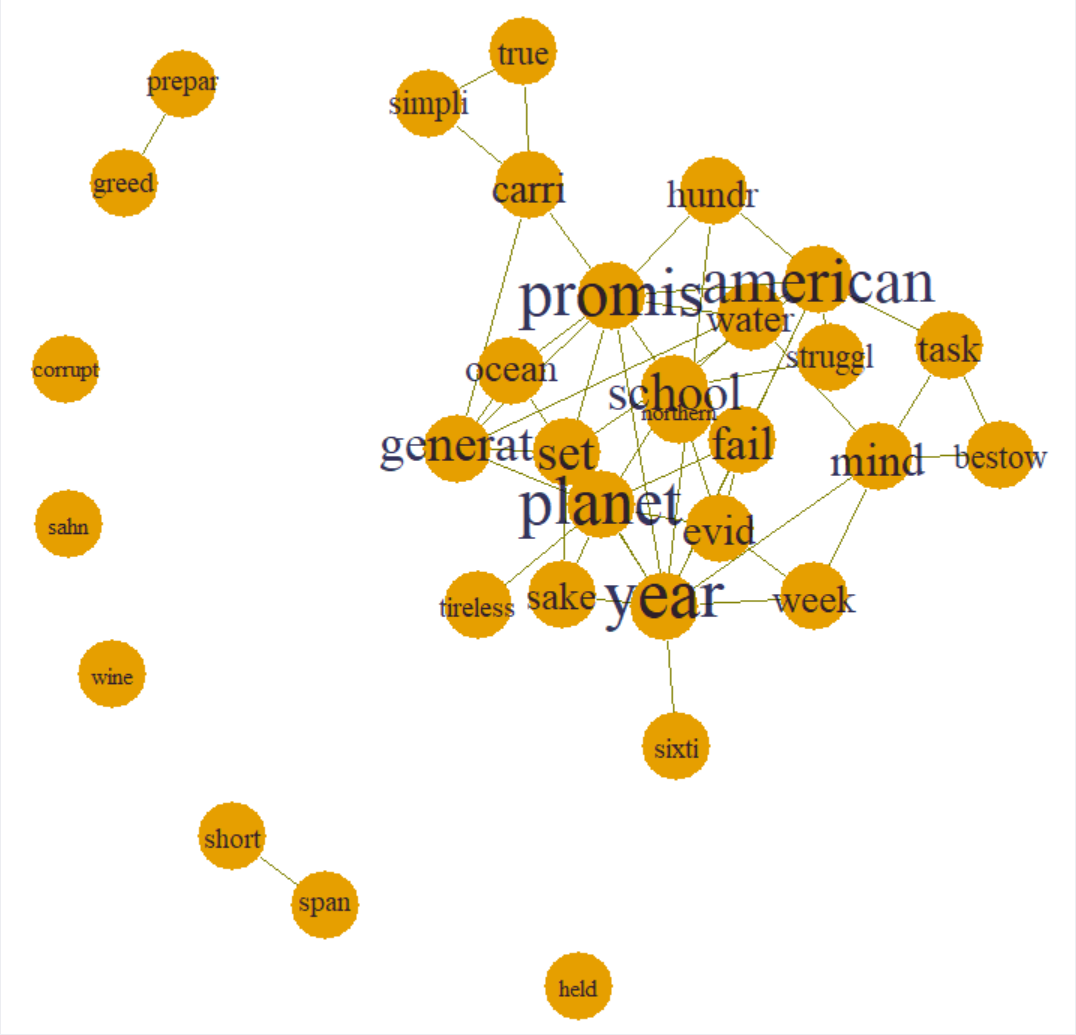
Obama vs. Obama

- Keyword network for the first five speeches



Obama vs. Obama

- Keyword network for the last five speeches



AGENDA

01 Theory 1: Text Preprocessing

02 Theory 2: Text Representation

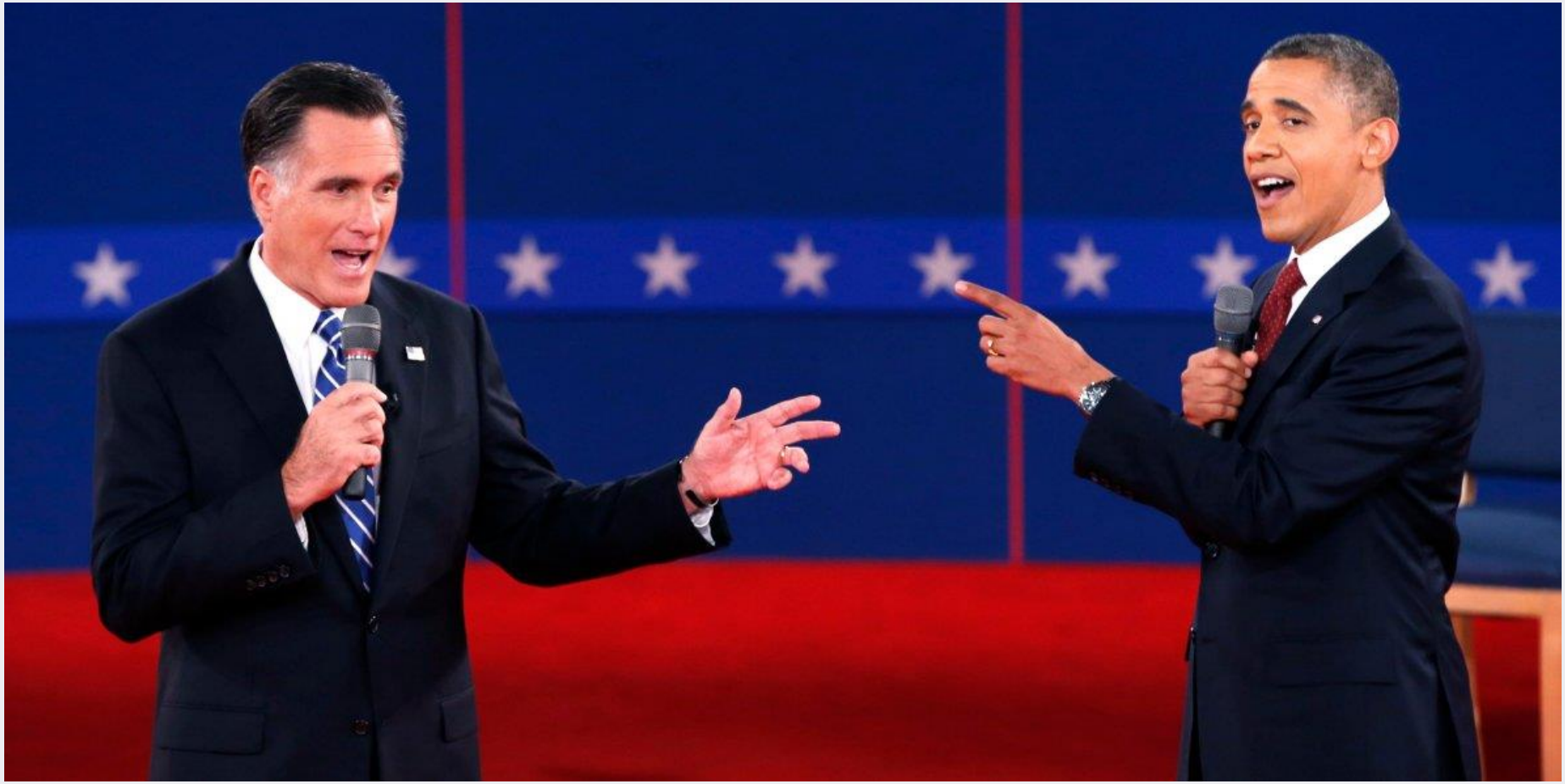
03 Theory 3: Text Summarization

04 Case Study 1: Obama's present vs. past

05 Case Study 2: Obama vs. Romney

Obama vs. Romney

- Obama vs. Romney in 2012



Obama vs. Romney

- Install necessary packages

```
install.packages("tm", dependencies = TRUE)
install.packages("wordcloud", dependencies = TRUE)
install.packages("arules", dependencies = TRUE)
install.packages("arulesViz", dependencies = TRUE)
install.packages("igraph", dependencies = TRUE)

library(tm)
library(wordcloud)
library(arules)
library(arulesViz)
library(igraph)
```

✓ Package "tm": provides basic functions for text mining

✓ Package "wordcloud" & "igraph": help to draw wordclouds and keywords network

✓ Package "arules": helps to find association rules

✓ Package "arulesViz": provide more various figures for the association rules

Obama vs. Romney

- Split the data into Obama's speeches and Romney's speeches
  - ✓ Merge the one's entire speeches and split it to individual sentences

```
# Load the data
load("SpeechData.RData")
# 1. Wordcloud -----------------------------------------------------
# Transform the data into Obama and Romney
obama.idx <- which(SpeechData$Speaker == "Obama")
romney.idx <- which(SpeechData$Speaker == "Romney")

obama.speech <- SpeechData[obama.idx, 2]
romney.speech <- SpeechData[romney.idx, 2]

obama.speech <- paste(obama.speech, collapse = " ")
romney.speech <- paste(romney.speech, collapse = " ")

obama.sentence <- strsplit(obama.speech, ".", fixed = TRUE)
obama.sentence <- as.data.frame(obama.sentence, stringsAsFactors = FALSE)
names(obama.sentence) <- "sentence"

romney.sentence <- strsplit(romney.speech, ".", fixed = TRUE)
romney.sentence <- as.data.frame(romney.sentence, stringsAsFactors = FALSE)
names(romney.sentence) <- "sentence"
```

Obama vs. Romney

- Construct corpuses and preprocess the text data

```
# Construct corpuses
# VectorSource specifies that the source is character vectors.
obamaCorpus <- Corpus(VectorSource(obama.sentence$sentence))
romneyCorpus <- Corpus(VectorSource(romney.sentence$sentence))

# Preprocessing # 1: to lower case
obamaCorpus <- tm_map(obamaCorpus, content_transformer(tolower))
romneyCorpus <- tm_map(romneyCorpus, content_transformer(tolower))

# 2: remove puntuations
obamaCorpus <- tm_map(obamaCorpus, content_transformer(removePunctuation))
romneyCorpus <- tm_map(romneyCorpus, content_transformer(removePunctuation))

# 3. remove numbers
obamaCorpus <- tm_map(obamaCorpus, content_transformer(removeNumbers))
romneyCorpus <- tm_map(romneyCorpus, content_transformer(removeNumbers))

# 4. remove stopwords (SMART stopwords list)
myStopwords <- c(stopwords("SMART"), "american", "america")
obamaCorpus <- tm_map(obamaCorpus, removeWords, myStopwords)
romneyCorpus <- tm_map(romneyCorpus, removeWords, myStopwords)

# 5. Stemming
obamaCorpus <- tm_map(obamaCorpus, stemDocument)
romneyCorpus <- tm_map(romneyCorpus, stemDocument)
myStopwords <- c("american", "america")
obamaCorpus <- tm_map(obamaCorpus, removeWords, myStopwords)
romneyCorpus <- tm_map(romneyCorpus, removeWords, myStopwords)
```

Obama vs. Romney

- Create Term-Document Matrix

```
# Term-Document Matrix
obamaTDM <- TermDocumentMatrix(obamaCorpus, control = list(minWordLength = 1))
romneyTDM <- TermDocumentMatrix(romneyCorpus, control = list(minWordLength = 1))

# Term-Document Matrix
obamaTDM
romneyTDM

as.matrix(obamaTDM)[11:30,11:30]
as.matrix(romneyTDM)[11:30,11:30]
```

```
> as.matrix(obamaTDM)[11:30,11:30]
           Docs
Terms       11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
  trust      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  獁\xbf     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  bush       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  cooper     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  generos    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  nation     1  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0
  presid     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  servic     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  shown      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  transit    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  fortyfour  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  oath       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  presidenti 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  peac       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  prosper    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
  rise       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  spoken     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  tide       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  water      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  word       0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
```

```
> as.matrix(romneyTDM)[11:30,11:30]
           Docs
Terms       11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
  paul       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  ryan       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  time       1  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0
  aspect     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  choic      0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  determin   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  histor     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  import     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  intim      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  make       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
  novemb     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  shape      1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  thing      0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  elect      0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0
  matter     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1
  deal       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  chang      0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  citizen    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  consequ    0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  countri    0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
```

Obama vs. Romney

- Find frequently used words

```
# Frequently used words
findFreqTerms(obamaTDM, lowfreq=15)
findFreqTerms(romneyTDM, lowfreq=15)
```

```
> findFreqTerms(obamaTDM, lowfreq=15)
  [1] "stand"      "today"      "bush"       "nation"     "presid"     "prosper"    "word"
  [8] "moment"     "peopl"      "generat"    "war"        "economi"    "hard"       "make"
 [15] "part"       "home"       "job"        "bring"      "care"       "day"        "energi"
 [22] "health"     "school"     "fear"       "challeng"   "face"       "real"       "met"
 [29] "time"       "hope"       "end"        "long"       "polit"      "promis"     "thing"
 [36] "young"      "chanc"      "histori"    "great"      "understand" "journey"    "work"
 [43] "men"        "women"      "life"       "world"      "live"       "struggl"    "power"
 [50] "worker"     "good"       "month"      "week"       "year"       "put"        "begin"
 [57] "start"      "call"       "creat"      "state"      "run"        "colleg"     "plan"
 [64] "question"   "common"     "countri"    "afford"     "famili"     "find"       "govern"
 [71] "small"      "ill"        "opportun"   "man"        "give"       "futur"      "readi"
 [78] "woman"      "iraq"       "respons"    "back"       "unit"       "chang"      "break"
 [85] "final"      "god"        "children"   "race"       "serv"       "told"       "citi"
 [92] "turn"       "dream"      "tonight"    "church"     "voic"       "black"      "democrat"
 [99] "republican" "white"      "elect"      "mccain"     "senat"      "campaign"   "hes"
[106] "love"       "street"     "made"       "didnt"      "washington" "centuri"    "million"
[113] "pay"        "perfect"    "bill"       "polici"     "problem"    "solv"       "wont"
[120] "parti"      "divid"      "stori"      "union"      "reason"     "weve"       "talk"
[127] "georg"      "invest"     "ive"        "carolina"   "south"      "fire"       "dont"
[134] "fight"      "clinton"    "communiti"  "john"       "tax"        "provid"     "educ"
[141] "doesnt"     "compani"    "racial"     "reverend"
```

Obama vs. Romney

- Find frequently used words

```
# Frequently used words
findFreqTerms(obamaTDM, lowfreq=15)
findFreqTerms(romneyTDM, lowfreq=15)
```

```
> findFreqTerms(romneyTDM, lowfreq=15)
  [1] "back"         "great"        "famili"       "kind"         "life"         "time"
  [7] "choic"        "make"         "thing"        "elect"        "chang"        "countri"
 [13] "histori"      "made"         "nation"       "turn"         "year"         "today"
 [19] "creat"        "debt"         "economi"      "futur"        "govern"       "growth"
 [25] "struggl"      "power"        "world"        "big"          "campaign"     "challeng"
 [31] "face"         "peopl"        "bring"        "real"         "ago"          "obama"
 [37] "attack"       "word"         "presid"       "promis"       "polit"        "cut"
 [43] "democrat"     "republican"   "reform"       "secur"        "busi"         "entrepreneur"
 [49] "invest"       "job"          "million"      "polici"       "small"        "obamacar"
 [55] "pay"          "day"          "work"         "put"          "school"       "find"
 [61] "good"         "children"     "home"         "leadership"   "militari"     "strength"
 [67] "call"         "long"         "men"          "women"        "spend"        "rise"
 [73] "stand"        "trade"        "problem"      "forc"         "grow"         "start"
 [79] "tax"          "feder"        "energi"       "offic"        "live"         "econom"
 [85] "plan"         "defens"       "moment"       "heart"        "faith"        "free"
 [91] "middl"        "program"      "state"        "war"          "nuclear"      "support"
 [97] "achiev"       "servic"       "final"        "interest"     "place"        "clear"
[103] "opportun"     "respons"      "share"        "lead"         "friend"       "prosper"
[109] "leader"       "peac"         "build"        "iran"         "centuri"      "confid"
[115] "bless"        "god"          "unit"         "valu"         "commit"       "hope"
[121] "purpos"       "afghanistan"  "honor"        "serv"         "veteran"      "east"
[127] "threat"       "freedom"      "critic"       "alli"         "israel"       "greater"
[133] "develop"      "enterpris"    "aid"          "system"       "guard"        "immigr"
```

Obama vs. Romney

- Word cloud for the Obama's speech

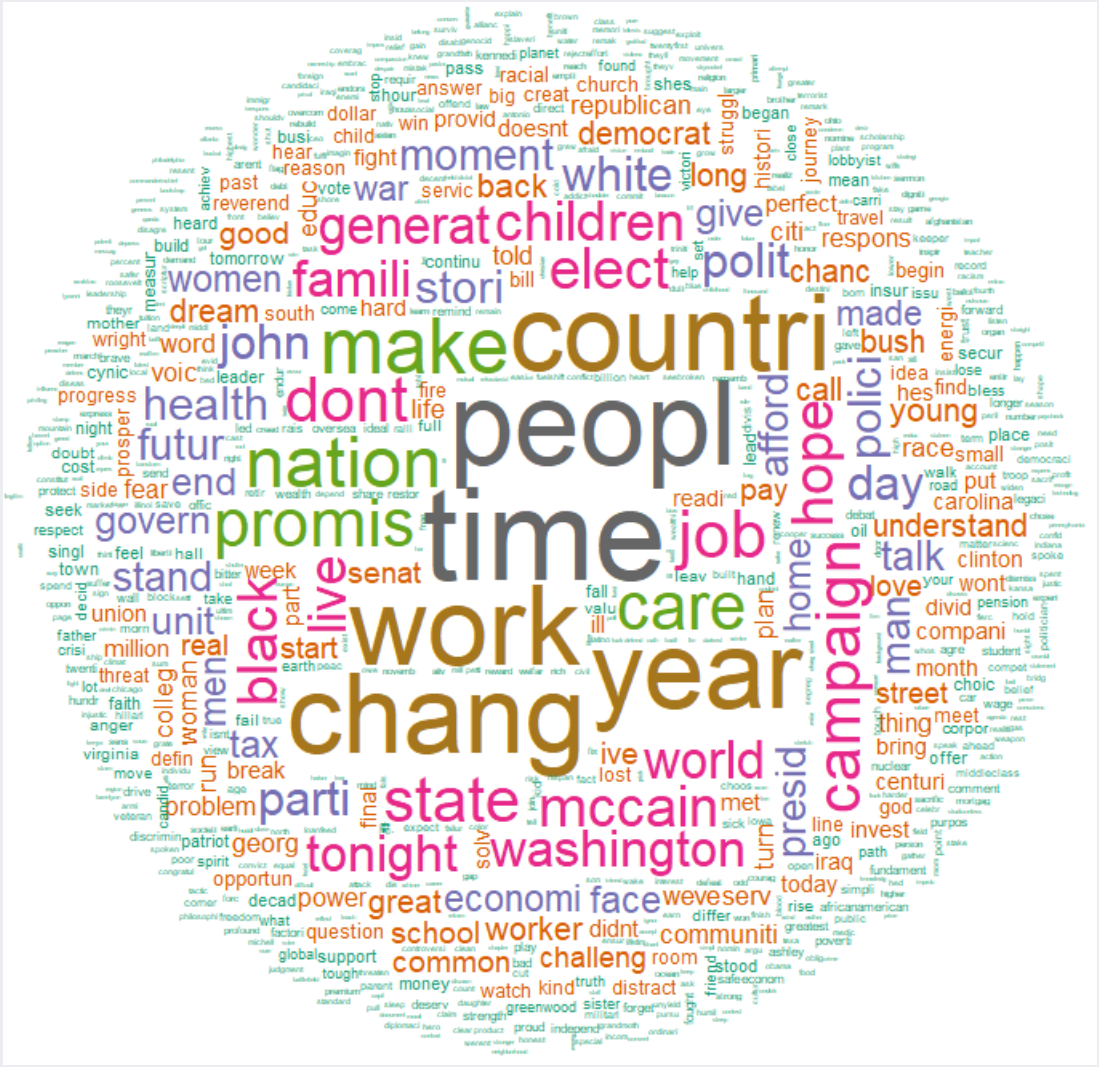
```
# Construct a Word Cloud with Obama's speeches
obama.wcmat <- as.matrix(obamaTDM)

# calculate the frequency of words
obama.word.freq <- sort(rowSums(obama.wcmat), decreasing=TRUE)
obama.keywords <- names(obama.word.freq)
obama.wcdat <- data.frame(word = obama.keywords, freq = obama.word.freq)

pal <- brewer.pal(8, "Dark2")
wordcloud(obama.wcdat$word, obama.wcdat$freq, min.freq=3, scale = c(5, 0.1),
          rot.per = 0.1, col=pal, random.order=F)
```

Obama vs. Romney

- Word cloud for the Obama's speech



Obama vs. Romney

- Word cloud for the Romney's speech

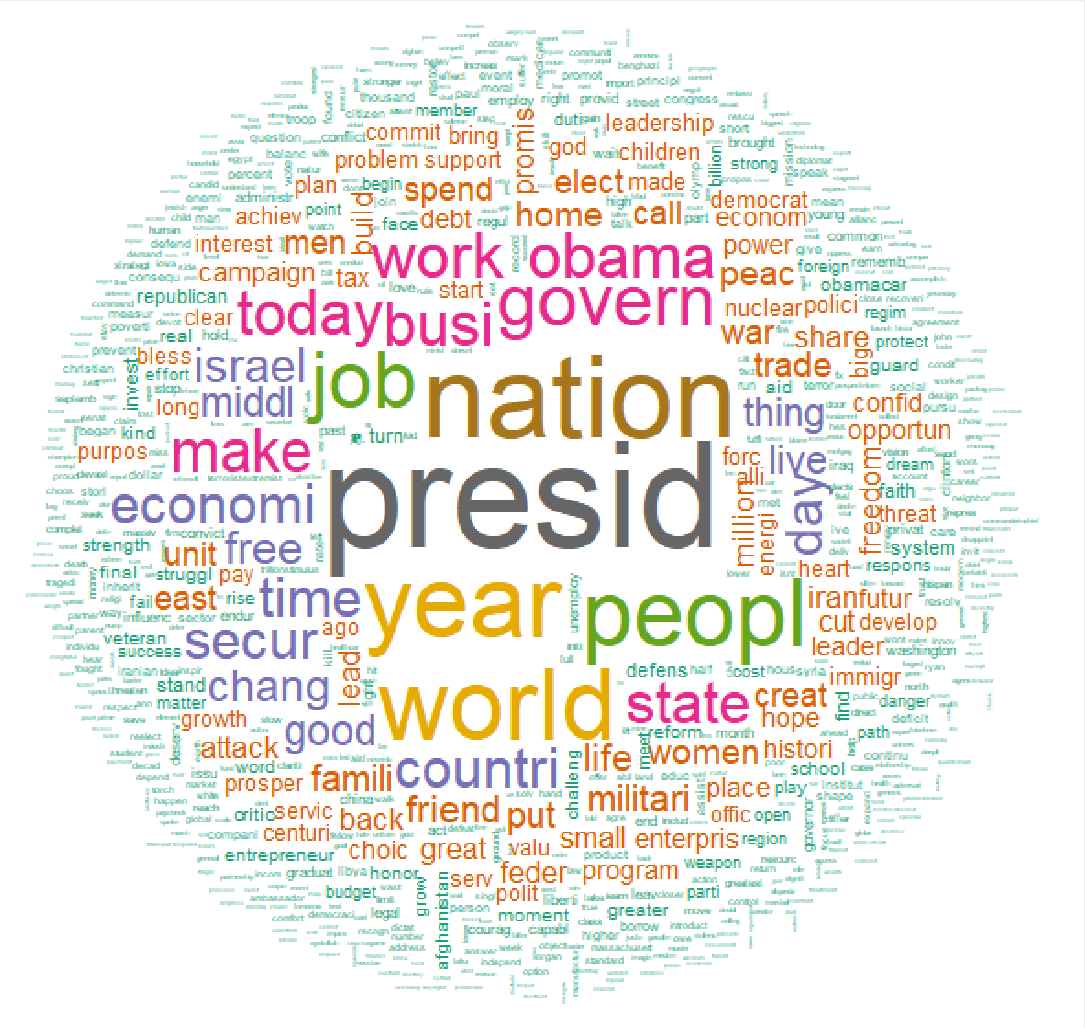
```
# Construct a Word Cloud with Romney's speeches
romney.wcmat <- as.matrix(romneyTDM)

# calculate the frequency of words
romney.word.freq <- sort(rowSums(romney.wcmat), decreasing=TRUE)
romney.keywords <- names(romney.word.freq)
romney.wcdat <- data.frame(word = romney.keywords, freq = romney.word.freq)

pal <- brewer.pal(8, "Dark2")
wordcloud(romney.wcdat$word, romney.wcdat$freq, min.freq=3, scale = c(5, 0.1),
          rot.per = 0.1, col=pal, random.order=F)
```

Obama vs. Romney

- Word cloud for the Romney's speech



Obama vs. Romney

- Association Rule for the Obama's speech

```
# Association Rules for Obama Speeches
obama.tran <- as.matrix(t(obamaTDM))
obama.tran <- as(obama.tran, "transactions")

obama.rules <- apriori(obama.tran, parameter=list(minlen=2,supp=0.007, conf=0.7)) inspect(obama.rules)

# Plot the rules
plot(obama.rules, method="graph")
```

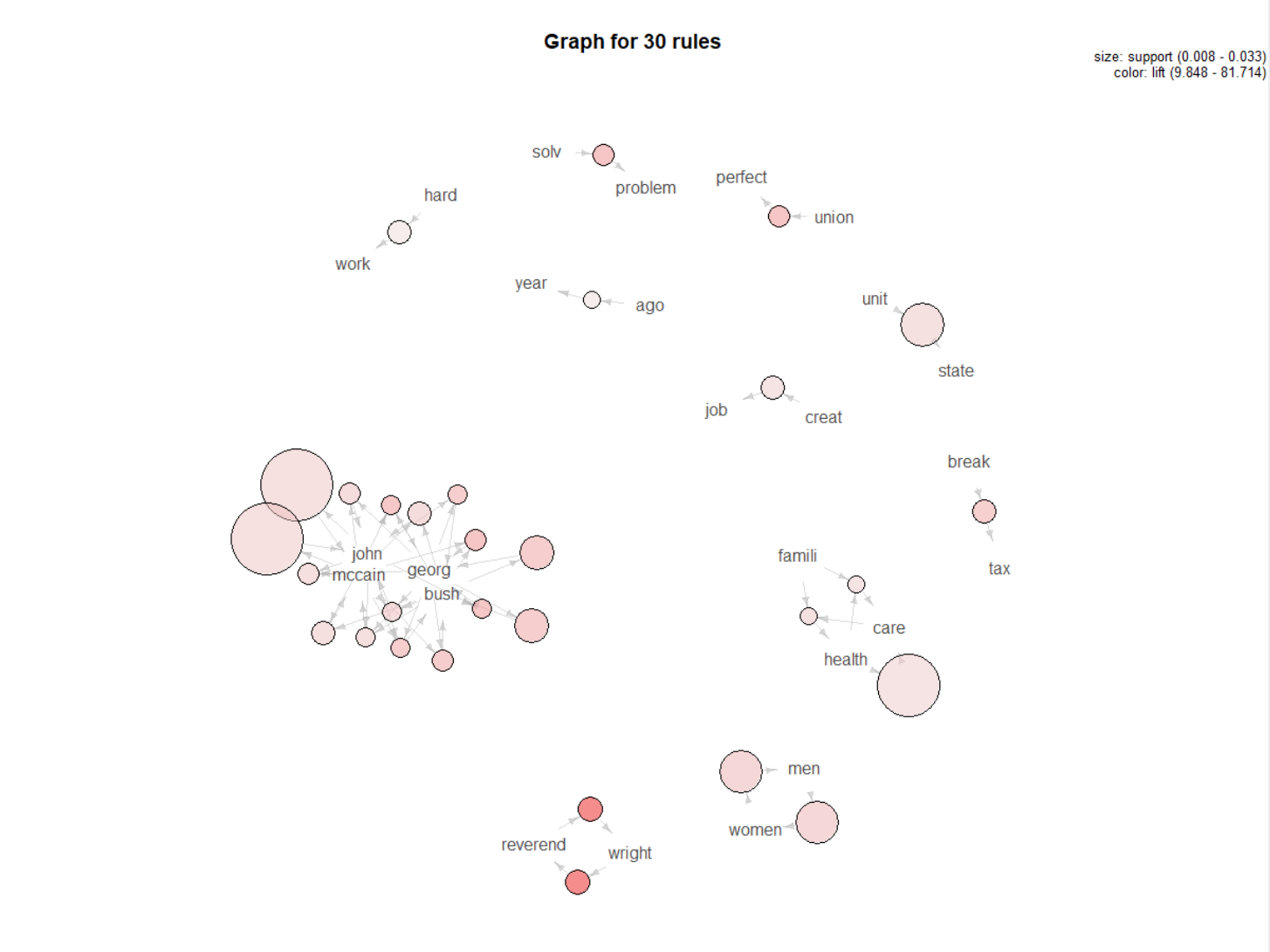
Obama vs. Romney

- Association Rule for the Obama's speech

```
> inspect(obama.rules)
     lhs                      rhs          support     confidence lift     count
[1]  {hard}                => {work}     0.010489510 0.7058824   9.84792 12
[2]  {wright}              => {reverend} 0.011363636 1.0000000  81.71429 13
[3]  {reverend}            => {wright}   0.011363636 0.9285714  81.71429 13
[4]  {union}               => {perfect}  0.009615385 0.7333333  49.34902 11
[5]  {solv}                => {problem}  0.009615385 0.7333333  49.34902 11
[6]  {creat}               => {job}      0.010489510 0.8000000  19.06667 12
[7]  {ago}                 => {year}     0.007867133 0.7500000  10.72500  9
[8]  {georg}               => {bush}     0.015734266 0.9000000  42.90000 18
[9]  {bush}                => {georg}    0.015734266 0.7500000  42.90000 18
[10] {break}               => {tax}      0.010489510 0.8000000  41.60000 12
[11] {women}               => {men}      0.019230769 0.8148148  33.29101 22
[12] {men}                 => {women}    0.019230769 0.7857143  33.29101 22
[13] {unit}                => {state}    0.020104895 0.7666667  21.92667 23
[14] {john}                => {mccain}   0.033216783 0.9743590  24.23188 38
[15] {mccain}              => {john}     0.033216783 0.8260870  24.23188 38
[16] {health}              => {care}     0.028846154 0.8918919  20.40649 33
[17] {georg,john}          => {bush}     0.008741259 0.9090909  43.33333 10
[18] {bush,john}           => {georg}    0.008741259 0.8333333  47.66667 10
[19] {mccain,georg}        => {bush}     0.009615385 0.9166667  43.69444 11
[20] {bush,mccain}         => {georg}    0.009615385 0.8461538  48.40000 11
[21] {georg,john}          => {mccain}   0.009615385 1.0000000  24.86957 11
[22] {mccain,georg}        => {john}     0.009615385 0.9166667  26.88889 11
[23] {bush,john}           => {mccain}   0.010489510 1.0000000  24.86957 12
[24] {bush,mccain}         => {john}     0.010489510 0.9230769  27.07692 12
[25] {health,famili}       => {care}     0.007867133 0.9000000  20.59200  9
[26] {care,famili}         => {health}   0.007867133 0.7500000  23.18919  9
[27] {bush,georg,john}     => {mccain}   0.008741259 1.0000000  24.86957 10
[28] {bush,mccain,georg}   => {john}     0.008741259 0.9090909  26.66667 10
[29] {mccain,georg,john}   => {bush}     0.008741259 0.9090909  43.33333 10
[30] {bush,mccain,john}    => {georg}    0.008741259 0.8333333  47.66667 10
```

Obama vs. Romney

- Association Rule for the Obama's speech



Obama vs. Romney

- Association Rule for the Romney's speech

```
# Association Rules for Romney's speeches
romney.tran <- as.matrix(t(romneyTDM))
romney.tran <- as(romney.tran, "transactions")

romney.rules <- apriori(romney.tran, parameter=list(minlen=2,supp=0.0045, conf=0.7))
inspect(romney.rules)

# Plot the rules
plot(romney.rules, method="graph")
```

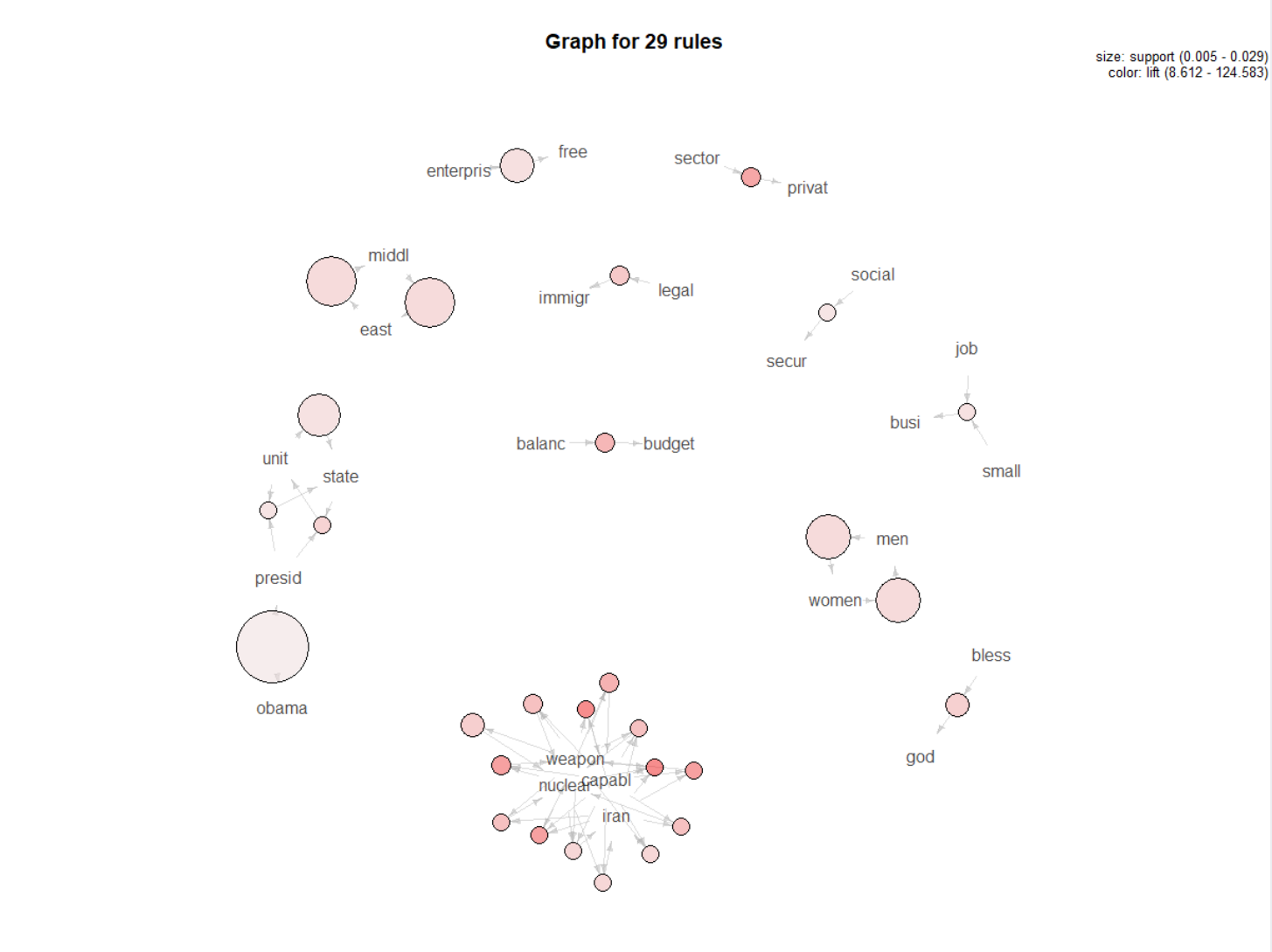
Obama vs. Romney

- Association Rule for the Romney's speech

```
> inspect(romney.rules)
      lhs                          rhs          support     confidence lift       count
[1]   {bless}                   => {god}      0.007357860 0.7333333   57.701754 11
[2]   {social}                  => {secur}    0.004682274 0.7000000   22.265957  7
[3]   {balanc}                  => {budget}   0.005351171 0.8000000   85.428571  8
[4]   {sector}                  => {privat}   0.005351171 0.8000000   99.666667  8
[5]   {legal}                   => {immigr}   0.005351171 0.7272727   67.954545  8
[6]   {weapon}                  => {nuclear}  0.007357860 0.7857143   58.732143 11
[7]   {enterpris}               => {free}     0.012040134 0.7500000   31.145833 18
[8]   {unit}                    => {state}    0.016053512 0.8571429   26.696429 24
[9]   {men}                     => {women}    0.016722408 0.8620690   40.274784 25
[10] {women}                    => {men}      0.016722408 0.7812500   40.274784 25
[11] {east}                     => {middl}    0.019397993 1.0000000   38.333333 29
[12] {middl}                    => {east}     0.019397993 0.7435897   38.333333 29
[13] {obama}                    => {presid}   0.029431438 0.7719298    8.612202 44
[14] {capabl,weapon}            => {nuclear}  0.005351171 1.0000000   74.750000  8
[15] {nuclear,capabl}           => {weapon}   0.005351171 1.0000000  106.785714  8
[16] {nuclear,weapon}           => {capabl}   0.005351171 0.7272727   90.606061  8
[17] {capabl,weapon}            => {iran}     0.004682274 0.8750000   46.718750  7
[18] {iran,capabl}              => {weapon}   0.004682274 1.0000000  106.785714  7
[19] {iran,weapon}              => {capabl}   0.004682274 1.0000000  124.583333  7
[20] {nuclear,capabl}           => {iran}     0.004682274 0.8750000   46.718750  7
[21] {iran,capabl}              => {nuclear}  0.004682274 1.0000000   74.750000  7
[22] {iran,weapon}              => {nuclear}  0.004682274 1.0000000   74.750000  7
[23] {job,small}                => {busi}     0.004682274 1.0000000   29.900000  7
[24] {presid,unit}              => {state}    0.004682274 0.7777778   24.224537  7
[25] {presid,state}             => {unit}     0.004682274 1.0000000   53.392857  7
[26] {nuclear,capabl,weapon}    => {iran}     0.004682274 0.8750000   46.718750  7
[27] {iran,capabl,weapon}       => {nuclear}  0.004682274 1.0000000   74.750000  7
[28] {nuclear,iran,capabl}      => {weapon}   0.004682274 1.0000000  106.785714  7
[29] {nuclear,iran,weapon}      => {capabl}   0.004682274 1.0000000  124.583333  7
```

Obama vs. Romney

- Association Rule for the Romney's speech




ANY
questions?