Text Analytics

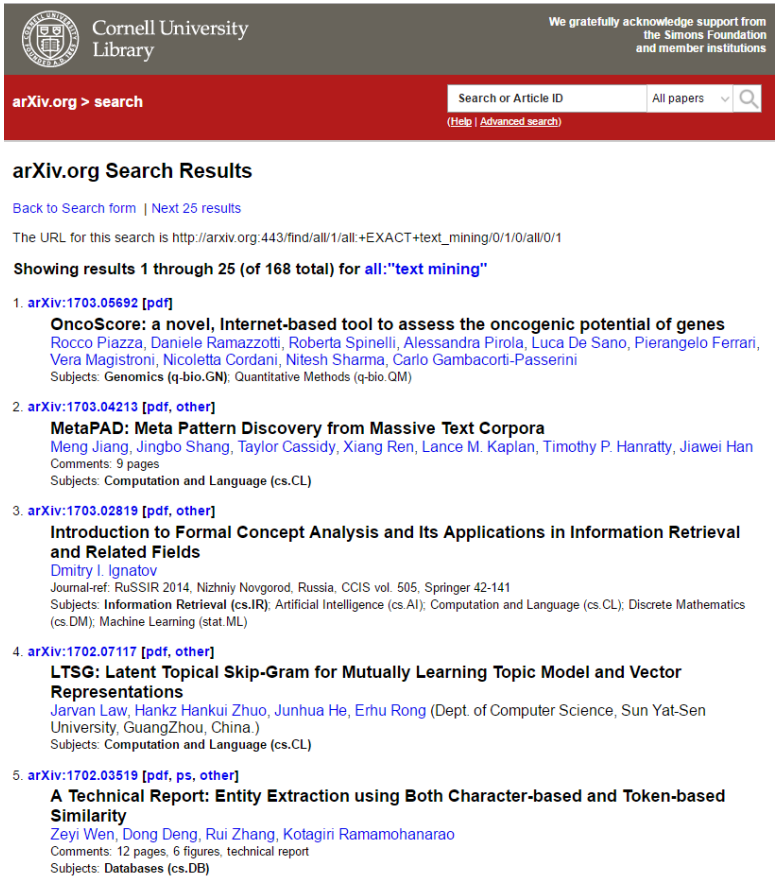
Pilsung Kang

School of Industrial Management Engineering

Korea University

Text Analytics

- What we have done so far…



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

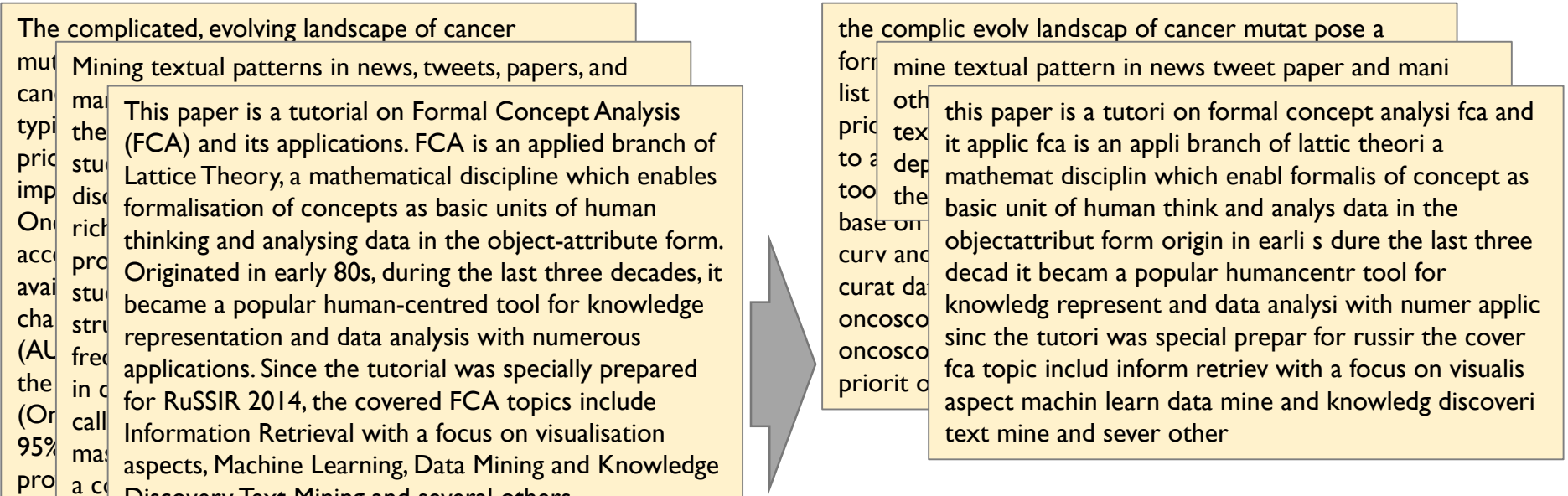
Mining textual patterns in news, tweets, papers, and

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.

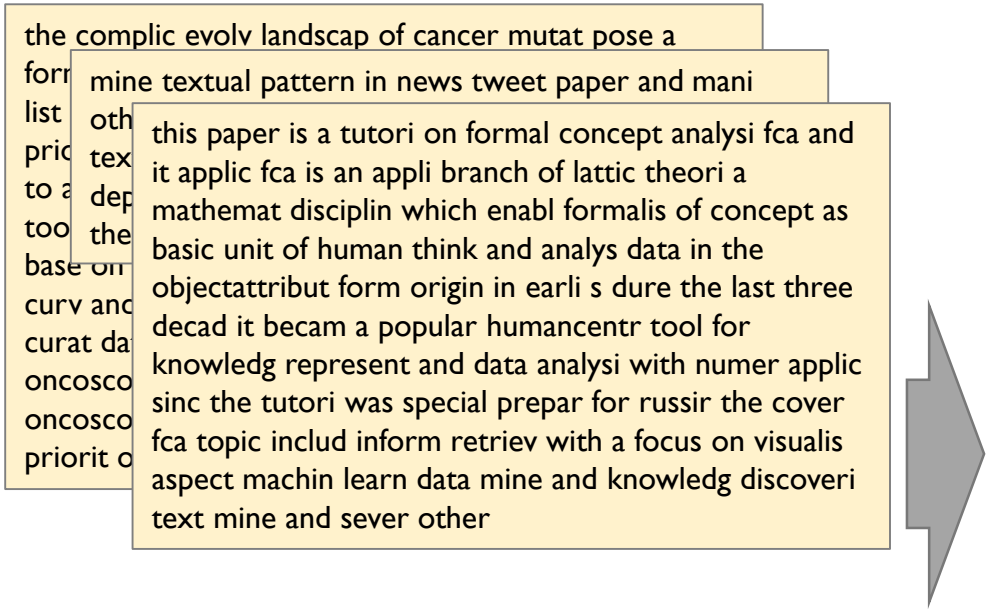
Text Analytics

- Preprocessing with some NLP techniques



Text Analytics

- What we will do…
  - ✓ Transform unstructured data into structured data



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

Text Analytics

- Install and load necessary packages

```
install.packages("textstem")
install.packages("tm")
install.packages("wordcloud")
install.packages("plyr")
install.packages("igraph")
install.packages("treemap")

library(textstem)
library(tm)
library(wordcloud)
library(plyr)
library(igraph)
library(dplyr)
library(treemap)
```

Text Analytics

- Load two datasets
  - ✓ Dataset 1: The abstracts of 500 arXiv papers with the search term "natural language processing“
  - ✓ Dataset 2: The abstracts of 500 arXiv papers with the search term "image processing"

```
load("arxiv_NLP.RData")
load("arxiv_Image_Processing.RData")

nlp_papers <- paste(arxiv_nlp$title, arxiv_nlp$abstract)
nlp_papers[1]

image_papers <- paste(arxiv_image_processing$title,
arxiv_image_processing$abstract)
image_papers[1]

# Remove invalid characters
nlp_papers <- sapply(nlp_papers,function(row) iconv(row, "latin1", "ASCII",
sub=""))
image_papers <- sapply(image_papers,function(row) iconv(row, "latin1",
"ASCII", sub=""))
```

Text Analytics 1: Tokenization

- 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.

Text Analytics 1: 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

Text Analytics 1: Tokenization

- Even tokenization can be difficult
  - ✓ Is John's sick one token or two?
    - If one → problems in parsing (where is the verb?)
    - If two → what do we do with John's house?
  - ✓ What to do with hyphens?
    - database vs. data-base vs. data base
  - ✓ What to do with "C++", "A/C", ":-)", "…", "ㅋ ㅋ ㅋ ㅋ ㅋ ㅋ ㅋ ㅋ"?
  - ✓ Some languages do not use whitespace (e.g., Chinese)

2013年5月，习主席在视察成都战区时，郑重提出在适当时候召开全军政治工作会议，并明确提出到古田召开这次会议，以更好弘扬我党我军的光荣传统和优良作风。6月，总政治部向中央军委提交《关于筹备召开全军政治工作会议的请示》，提出要通过召开会议形成一个指导性文件。习主席随即批示同意，明确要求这个文件要充分体现深厚的历史积淀和政治意蕴，能够管一个时期，起到历史性作用。

- Consistent tokenization is important for all later processing steps.

Text Analytics 2: 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.

Text Analytics 2: 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

Text Analytics 2: 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

Text Analytics 2: 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

Text Analytics 2: Morphological Analysis

- Lemmatization

```
nlp_lemma <- NULL
image_lemma <- NULL

# Lemmatization
for (i in 1:length(nlp_papers)){
    cat("Lemmatizing", i, "-th paper. \n")
    nlp_lemma[i] <- lemmatize_strings(nlp_papers[i])
}

for (j in 1:length(image_papers)){
    cat("Lemmatizing", j, "-th paper. \n")
    image_lemma[j] <- lemmatize_strings(image_papers[j])
}
```

Text Analytics 2: Morphological Analysis

- Before lemmatization

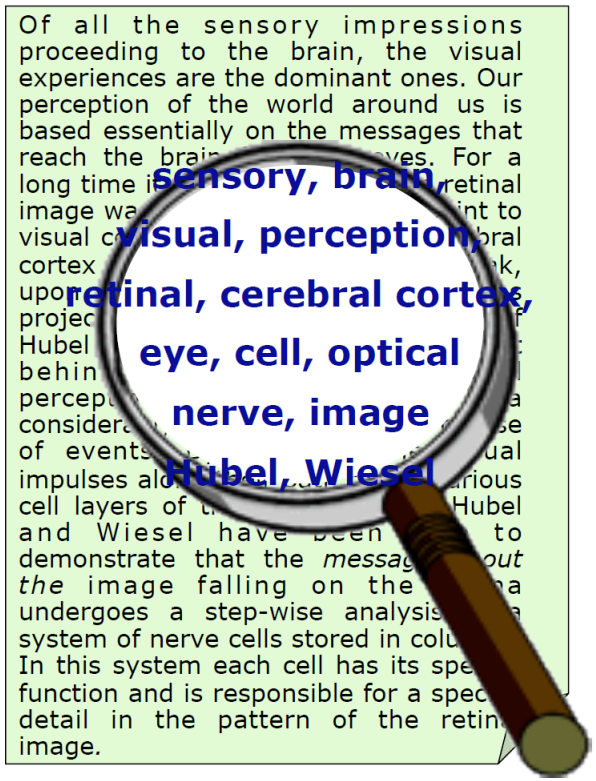
```
> nlp_papers[1]
[1] "Towards non-toxic landscapes: Automatic toxic comment detection using DNN The spectacular expans
ion of the Internet led to the development of a new research problem in the natural language processi
ng field: automatic toxic comment detection, since many countries prohibit hate speech in public medi
a. There is no clear and formal definition of hate, offensive, toxic and abusive speeches. In this ar
ticle, we put all these terms under the \"umbrella\" of toxic speech. The contribution of this paper
 is the design of binary classification and regression-based approaches aiming to predict whether a c
omment is toxic or not. We compare different unsupervised word representations and different DNN clas
sifiers. Moreover, we study the robustness of the proposed approaches to adversarial attacks by addin
g one (healthy or toxic) word. We evaluate the proposed methodology on the English Wikipedia Detox co
rpus. Our experiments show that using BERT fine-tuning outperforms feature-based BERT, Mikolov's word
 embedding or fastText representations with different DNN classifiers."
```

- After lemmatization

```
> nlp_lemma[1]
[1] "Towards non - toxic landscape : Automatic toxic comment detection use DNN The spectacular expansi
on of the Internet lead to the development of a new research problem in the natural language process
 field: automatic toxic comment detection, since many country prohibit hate speech in public medium.
 There be no clear and formal definition of hate, offensive, toxic and abusive speech. In this articl
e, we put all this term under the \" umbrella \" of toxic speech. The contribution of this paper be t
he design of binary classification and regression - base approach aim to predict whether a comment be
 toxic or not. We compare different unsupervised word representation and different DNN classifier. Mo
reover, we study the robustness of the propose approach to adversarial attack by add one ( healthy or
 toxic ) word. We evaluate the propose methodology on the English Wikipedia Detox corpus. Our experim
ent show that use BERT fine - tune outperform feature - base BERT, Mikolov's word embed or fastText r
epresentation with different DNN classifier."
```

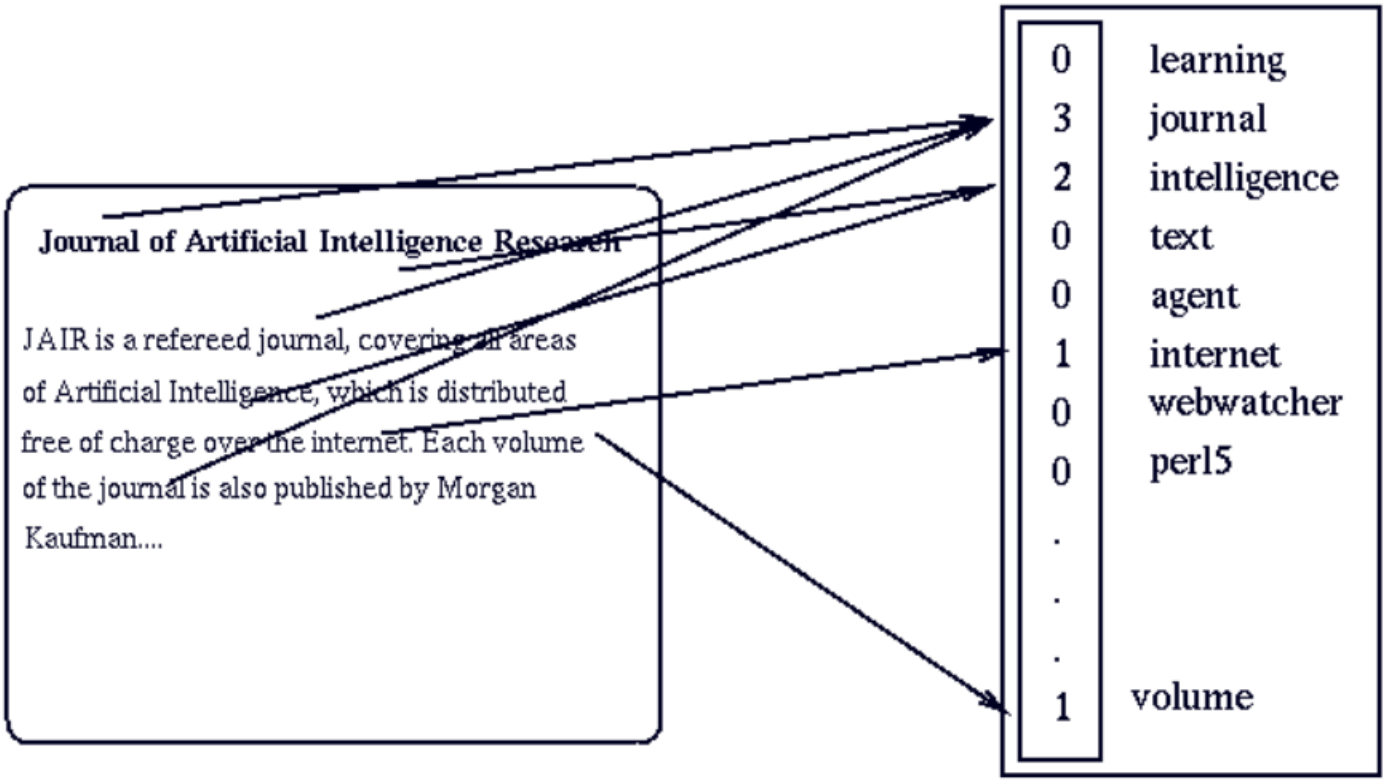
Text Representation: Bag-of-Words

- 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?



Text Representation: Bag-of-Words

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



Text Representation: Bag-of-Words

- 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

Text Representation: Bag-of-Words

- 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

Text Preprocessing

- Remove unnecessary information: to lower case

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

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

✓ Before

```
> nlp_papers[1]
[1] "Towards non-toxic landscapes: Automatic toxic comment detection using DNN The spectacular expans
ion of the Internet led to the development of a new research problem in the natural language processi
ng field: automatic toxic comment detection, since many countries prohibit hate speech in public medi
a. There is no clear and formal definition of hate, offensive, toxic and abusive speeches. In this ar
ticle, we put all these terms under the \"umbrella\" of toxic speech. The contribution of this paper
 is the design of binary classification and regression-based approaches aiming to predict whether a c
omment is toxic or not. We compare different unsupervised word representations and different DNN clas
sifiers. Moreover, we study the robustness of the proposed approaches to adversarial attacks by addin
g one (healthy or toxic) word. We evaluate the proposed methodology on the English Wikipedia Detox co
rpus. Our experiments show that using BERT fine-tuning outperforms feature-based BERT, Mikolov's word
 embedding or fastText representations with different DNN classifiers."
```

Text Preprocessing

- Remove unnecessary information: to lower case

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

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

✓ After

```
> nlp_corpus[[1]][1]
$content
[1] "towards non - toxic landscape: automatic toxic comment detection use dnn the spectacular expansi
on of the internet lead to the development of a new research problem in the natural language process
 field: automatic toxic comment detection, since many country prohibit hate speech in public medium.
 there be no clear and formal definition of hate, offensive, toxic and abusive speech. in this articl
e, we put all this term under the \" umbrella \" of toxic speech. the contribution of this paper be t
he design of binary classification and regression - base approach aim to predict whether a comment be
 toxic or not. we compare different unsupervised word representation and different dnn classifier. mo
reover, we study the robustness of the propose approach to adversarial attack by add one ( healthy or
 toxic ) word. we evaluate the propose methodology on the english wikipedia detox corpus. our experim
ent show that use bert fine - tune outperform feature - base bert, mikolov's word embed or fasttext r
epresentation with different dnn classifier."
```

Text Preprocessing

- Remove unnecessary information: remove punctuations

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

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

```
> nlp_corpus[[1]][1]
$content
[1] "towards non  toxic landscape automatic toxic comment detection use dnn the spectacular expansion
 of the internet lead to the development of a new research problem in the natural language process fi
eld automatic toxic comment detection since many country prohibit hate speech in public medium there
 be no clear and formal definition of hate offensive toxic and abusive speech in this article we put
 all this term under the  umbrella  of toxic speech the contribution of this paper be the design of b
inary classification and regression  base approach aim to predict whether a comment be toxic or not w
e compare different unsupervised word representation and different dnn classifier moreover we study t
he robustness of the propose approach to adversarial attack by add one  healthy or toxic  word we eva
luate the propose methodology on the english wikipedia detox corpus our experiment show that use bert
 fine  tune outperform feature  base bert mikolovs word embed or fasttext representation with differe
nt dnn classifier"
```

Text Preprocessing

- Remove unnecessary information: remove numbers

```
# 3. remove numbers
nlp_corpus <- tm_map(nlp_corpus, content_transformer(removeNumbers))
nlp_corpus[[5]][1]

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

✓ Before

```
> nlp_corpus[[5]][1]
$content
[1] "a subword level language model for bangla language language model be at the core of natural lang
uage process the ability to represent natural language give rise to its application in numerous nlp t
ask include text classification summarization and translation research in this area be very limit in
 bangla due to the scarcity of resource except for some count  base model and very recent neural lang
uage model be propose which be all base on word and limit in practical task due to their high perplex
ity this paper attempt to approach this issue of perplexity and propose a subword level neural langua
ge model with the awd  lstm architecture and various other technique suitable for train in bangla lan
guage the model be train on a corpus of bangla newspaper article of a appreciable size consist of muc
h than 285 million word token the performance comparison with various other model depict the signific
ant reduction in perplexity the propose model provide reach as low as 39 84 in just 20 epoch"
```

Text Preprocessing

- Remove unnecessary information: remove numbers

```
# 3. remove numbers
nlp_corpus <- tm_map(nlp_corpus, content_transformer(removeNumbers))
nlp_corpus[[5]][1]

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

✓ After

```
> nlp_corpus[[5]][1]
$content
[1] "a subword level language model for bangla language language model be at the core of natural lang
uage process the ability to represent natural language give rise to its application in numerous nlp t
ask include text classification summarization and translation research in this area be very limit in
 bangla due to the scarcity of resource except for some count  base model and very recent neural lang
uage model be propose which be all base on word and limit in practical task due to their high perplex
ity this paper attempt to approach this issue of perplexity and propose a subword level neural langua
ge model with the awd  lstm architecture and various other technique suitable for train in bangla lan
guage the model be train on a corpus of bangla newspaper article of a appreciable size consist of muc
h than  million word token the performance comparison with various other model depict the significant
 reduction in perplexity the propose model provide reach as low as   in just  epoch"
```

Text Preprocessing: Stopwords

- 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

Text Preprocessing: Stopwords

- 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"	

Text Preprocessing: Stopwords

- 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					

Text Preprocessing: Stopwords

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

아 어찌됏든 하기보다는
휴 그위에 차라리
아이구 게다가 하는 편이 낫다
아이쿠 점에서 보아 흐흐
아이고 비추어 보아 놀라다
어 고려하면 상대적으로 말하
나 하게될것이다 자면
우리 일것이다 마치
저희 비교적 아니라면
따라 좀 쉿
의해 보다더 그렇지 않으면
을 비하면 그렇지 않다면
를 시키다 안 그러면
에 하게하다 아니었다면
의 할만하다 하든지
가 의해서 아니면
으로 연이서 이라면
로 이어서 좋아
에게 잇따라 알았어
뿐이다 뒤따라 하는것도
의거하여 뒤이어 그만이다
근거하여 결국 어쩔수 없다
입각하여 의지하여 하나
기준으로 기대여 일
예하면 통하여 일반적으로
예를 들면 자마자 일단
예를 들자면 더욱더 한켠으로는
저 불구하고 오자마자
소인 얼마든지 이렇게되면
소생 마음대로 이와같다면
저희 주저하지 않고 전부
지말고 곧 한마디
하지마 즉시 한항목
하지마라 바로 근거로
다른 당장 하기에
물론 하자마자 아울러
또한 밖에 안된다 하지 않도록
그리고 하면된다 않기 위해서
비길수 없다 그래 이르기까지
해서는 안된 그렇지 이 되다
다 요컨대 로 인하여

뿐만 아니라 다시 말하자면 까닭으로
만이 아니다 바꿔 말하면 이유만으로
만은 아니다 즉 이로 인하여
막론하고 구체적으로 그래서
관계없이 말하자면 이 때문에
그치지 않다 시작하여 그러므로
그러나 시초에 그런 까닭에
그런데 이상 알 수 있다
하지만 허 결론을 낼 수 있
든간에 헉 다
논하지 않다 허걱 으로 인하여
따지지 않다 바와같이 있다
설사 해도좋다 어떤것
비록 해도된다 관계가 있다
더라도 게다가 관련이 있다
아니면 더구나 연관되다
만 못하다 하물며 어떤것들
하는 편이 낫다 와르르 에 대해
다 팍 이리하여
불문하고 퍽 그리하여
향하여 펄렁 여부
향해서 똥안 하기보다는
향하다 이래 하느니
쪽으로 하고있었다 하면 할수록
틈타 이었다 운운
이용하여 에서 이러이러하다
타다 로부터 하구나
오르다 까지 하도다
제외하고 예하면 다시말하면
이 외에 했어요 다음으로
이 밖에 해요 에 있다
하여야 함께 에 달려 있다
비로소 같이 우리
한다면 몰라 더불어 우리들
도 마저 오히려
외에도 마저도 하기는한데
이곳 양자 어떻게
여기 모두 어떻해
부터 습니다 어찌됏어
기점으로 가까스로 어때
따라서 하려고하다 어째서

할 생각이다 즈음하여 본대로
하려고하다 다른 자
이리하여 다른 방면으로 이
그리하여 해봐요 이쪽
그렇게 함으 습니까 여기
로써 했어요 이것
하지만 말할것도 없고 이번
일때 무릎쓰고 이렇게말하자면
할때 개의치않고 이런
앞에서 하는것만 못하다 이러한
중에서 하는것이 낫다 이와 같은
보는데서 매 요만큼
으로써 매번 요만한 것
로써 들 얼마 안 되는 것
까지 모 이만큼
해야한다 어느것 이 정도의
일것이다 어느 이렇게 많은 것
반드시 로써 이와 같다
할줄알다 갖고말하자면 이때
할수있다 어디 이렇구나
할수있어 어느쪽 것과 같이
임에 틀림없 어느것 끼익
다 어느해 삐걱
한다면 어느 년도 따위
등 라 해도 와 같은 사람들
등등 언젠가 부류의 사람들
제 어떤것 왜냐하면
겨우 어느것 중의하나
단지 저기 오직
다만 저쪽 오로지
할뿐 저것 에 한하다
딩동 그때 하기만 하면
댕그 그럼 도착하다
대해서 그러면 까지 미치다
대하여 요만한걸 도달하다
대하면 그래 정도에 이르다
훨씬 그때 할 지경이다
얼마나 저것만큼 결과에 이르다
얼마만큼 그저 관해서는
얼마큼 이르기까지 여러분
남짓 할 줄 안다 하고 있다
여 할 힘이 있다 한 후

얼마간 너 혼자
약간 너희 자기
다소 당신 자기집
좀 어찌 자신
조금 설마 우에 종합한것과
다수 차라리 같이
몇 할지언정 총적으로 보면
얼마 할지라도 총적으로 말하면
지만 할망정 총적으로
하물며 할지언정 대로 하다
또한 구토하다 으로서
그러나 게우다 참
그렇지만 토하다 그만이다
하지만 메쓰겁다 할 따름이다
이외에도 옆사람 쿵
대해 말하자 훼 탕탕
면 쳇 광광
뿐이다 의거하여 둥둥
다음에 근거하여 봐
반대로 의해 봐라
반대로 말하 따라 아이야
자면 힘입어 아니
이와 반대로 그 와아
바꾸어서 말 다음 응
하면 버금 아이
바꾸어서 한 두번째로 참나
다면 기타 년
만약 첫번째로 월
그렇지않으 나머지는 일
면 그중에서 령
까악 견지에서 영
툭 형식으로 쓰여 일
딱 입장에서 이
삐걱거리다 위해서 삼
보드득 단지 사
비걱거리다 의해되다 오
꽈당 하도록시키다 육
응당 뿐만아니라 륙
해야한다 반대로 칠
에 가서 전후 팔
각 전자 구

Text Preprocessing

- Remove unnecessary information: remove stopwords
  - ✓ You can add your own stopwords to an existing stopwords list

```
# 4. remove stopwords (SMART stopwords list)
arxiv_stopwords <- c("process", "paper", "model", "learn", "propose", "task",
                     "method", "show", "approach", "result", "system")
myStopwords <- c(stopwords("SMART"), arxiv_stopwords)

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

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

```
> nlp_corpus[[1]][1]
$content
[1] "   toxic landscape automatic toxic comment detection  dnn  spectacular expansion   internet lead
   development    research problem   natural language process field automatic toxic comment detection
   country prohibit hate speech  public medium    clear  formal definition  hate offensive toxic  abu
sive speech   article  put   term    umbrella   toxic speech  contribution      design  binary classi
fication  regression  base approach aim  predict   comment  toxic    compare  unsupervised word repre
sentation   dnn classifier   study  robustness   propose approach  adversarial attack  add   healthy
 toxic  word  evaluate  propose methodology   english wikipedia detox corpus  experiment show   bert
 fine  tune outperform feature  base bert mikolovs word embed  fasttext representation   dnn classifi
er"
```

Text Preprocessing

- Bag-of-Words Representation: Construct a Term-Document Matrix

```
# Construct Term-Document Matrix
nlpTDM <- TermDocumentMatrix(nlp_corpus, control = list(minWordLength = 2))
imageTDM <- TermDocumentMatrix(image_corpus, control = list(minWordLength =
2))

# Check the Term-Document Matrix
as.matrix(nlpTDM)[11:30,11:30]
as.matrix(imageTDM)[11:30,11:30]
```

```
> as.matrix(nlpTDM)[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
  binary          0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0
  classification  0  0  0  1  1  0  0  2  0  0  0  0  0  1  0  0  0  0  6  0
  classifier      0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  1  0
  clear           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  comment         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  compare         1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
  contribution    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  corpus          0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  1  0  0  0  0
  country         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  definition      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  design          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  detection       0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
  detox           0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  development     0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0
  dnn             0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  embed           0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  2  0  9  0
  english         0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0
  evaluate        0  1  0  0  0  0  1  1  0  0  0  2  0  0  1  0  0  0  0  0
  expansion       0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
  experiment      0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  1
```

```
> as.matrix(imageTDM)[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
  datum           0  2  1  1  0  2  1  0  0  1  0  1  0  0  0  1  1  0  0  0
  design          0  1  1  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  development     0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  1  0
  device          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  domain          0  0  0  0  1  0  0  0  0  0  0  0  3  0  0  0  0  0  2  0
  ease            0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  easily          0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  edge            0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
  effective       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  environment     0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
  fog             0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  framework       0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0
  gateway         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  grow            0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  healthcare      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  heterogeneous   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  igatelink       0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  image           1  1  9  4  1  1  2  1  2  7  5  2  1 10  4  9  2  1  1 12
  important       0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  2  0  0  0
  increasingly    0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Text Analytics

- Highly Frequent Tokens

```
# High frequent words
nlpTDM_matrix <- as.matrix(nlpTDM)
nlp_word_count <- rowSums(nlpTDM_matrix)
nlp_word_count <- sort(nlp_word_count, decreasing = TRUE)
nlp_word_count[1:30]
```

```
> nlp_word_count[1:30]
    language      natural        train        datum         word         base
        1035          641          449          428          392          384
     network       neural         text         deep  performance representation
         359          321          304          274          272          253
     dataset         good          nlp  information        state      problem
         250          237          234          223          219          200
        work      machine          art       domain        large classification
         191          190          183          177          174          170
   attention   embeddings      feature     analysis      achieve    technique
         170          162          161          161          158          156
```

Text Analytics

- Highly Frequent Tokens

```
# High frequent words
imageTDM_matrix <- as.matrix(imageTDM)
image_word_count <- rowSums(imageTDM_matrix)
image_word_count <- sort(image_word_count, decreasing = TRUE)
image_word_count[1:30]
```

```
> image_word_count[1:30]
        image         datum       network          base          deep     algorithm       problem
         1927           490           485           462           366           311           299
         good   application        neural          high         train          time       dataset
          251           248           247           226           224           219           218
      feature   performance  segmentation     technique       present     detection       provide
          214           210           210           207           200           191           168
     analysis          work         study         state       quality         field      function
          163           161           157           151           150           143           140
convolutional   information
          140           140
```

Text Analytics

- Highly Frequent Tokens

```
nlp_freq_words <- findFreqTerms(nlpTDM, lowfreq=100)
image_freq_words <- findFreqTerms(imageTDM, lowfreq=100)

intersect(nlp_freq_words, image_freq_words)
setdiff(nlp_freq_words, image_freq_words)
setdiff(image_freq_words, nlp_freq_words)
```

✓ Words appearing more than 100 times in the both datasets

```
> intersect(nlp_freq_words, image_freq_words)
 [1] "base"          "classification" "compare"       "feature"       "problem"
 [6] "study"         "improve"        "technique"     "accuracy"      "algorithm"
[11] "information"   "performance"    "provide"       "dataset"       "domain"
[16] "application"   "architecture"   "high"          "neural"        "train"
[21] "analysis"      "apply"          "art"           "datum"         "deep"
[26] "image"         "large"          "network"       "state"         "work"
[31] "good"          "present"        "time"          "achieve"       "demonstrate"
[36] "recognition"   "computer"       "vision"        "machine"       "set"
[41] "challenge"
```

Text Analytics

- Highly Frequent Tokens

```
nlp_freq_words <- findFreqTerms(nlpTDM, lowfreq=100)
image_freq_words <- findFreqTerms(imageTDM, lowfreq=100)

intersect(nlp_freq_words, image_freq_words)
setdiff(nlp_freq_words, image_freq_words)
setdiff(image_freq_words, nlp_freq_words)
```

✓ Words appearing more than 100 times in the "nlp" dataset but not in the "image" dataset

```
> setdiff(nlp_freq_words, image_freq_words)
 [1] "bert"           "corpus"         "embed"          "evaluate"       "experiment"
 [6] "language"       "natural"        "representation" "research"       "speech"
[11] "word"           "generate"       "nlp"            "understand"     "test"
[16] "text"           "exist"          "level"          "recent"         "label"
[21] "pre"            "make"           "embeddings"     "knowledge"      "human"
[26] "semantic"       "sequence"       "transformer"    "context"        "sentence"
[31] "question"       "graph"          "attention"      "answer"
```

Text Analytics

- Highly Frequent Tokens

```
nlp_freq_words <- findFreqTerms(nlpTDM, lowfreq=100)
image_freq_words <- findFreqTerms(imageTDM, lowfreq=100)

intersect(nlp_freq_words, image_freq_words)
setdiff(nlp_freq_words, image_freq_words)
setdiff(image_freq_words, nlp_freq_words)
```

✓ Words appearing more than 100 times in the "image" dataset but not in the "nlp" dataset

```
> setdiff(image_freq_words, nlp_freq_words)
 [1] "design"        "framework"     "function"      "quality"       "medical"
 [6] "convolutional" "include"       "matrix"        "noise"         "point"
[11] "parameter"     "sample"        "signal"        "structure"     "real"
[16] "low"           "detection"     "object"        "segmentation"  "field"
[21] "space"
```

Text Analytics

- Associated words

```
# Which words are associated with "image"?
findAssocs(nlpTDM, 'image', 0.2)
findAssocs(imageTDM, 'image', 0.2)
```

```
> findAssocs(nlpTDM, 'image', 0.2)
$image
         caption        correlate characterisation           flickr        observers
            0.68             0.53             0.52             0.52             0.52
    unexpectedly           facial       subjective           visual        imitation
            0.52             0.49             0.41             0.38             0.36
     description         fidelity          fifteen             fool       perceptual
            0.33             0.32             0.32             0.32             0.32
        reliably       substitute        criterion       adjective     proportional
            0.32             0.32             0.29             0.29             0.28
          vision            gans          retrieve           metric        judgment
            0.28             0.28             0.27             0.27             0.27
         perturb         generate   reconstruction             coco             dnns
            0.25             0.24             0.24             0.23             0.23
        endeavor         localize       equivalent           sagan    dissimilarity
            0.23             0.23             0.23             0.23             0.23
     neuroscore psychoperceptual        induction              gan         creation
            0.23             0.23             0.22             0.22             0.22
        visually      nonetheless          feature             norm       generative
            0.22             0.22             0.21             0.21             0.20
```

Text Analytics

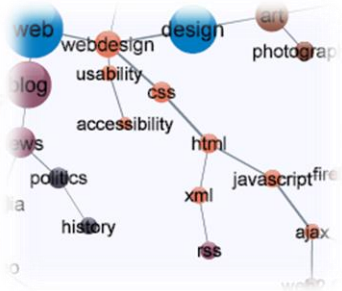
- Associated words

```
# Which words are associated with "image"?
findAssocs(nlpTDM, 'image', 0.2)
findAssocs(imageTDM, 'image', 0.2)
```

```
> findAssocs(imageTDM, 'image', 0.2)
$image
       medical        quality        dataset   registration         manual      treatment
          0.29           0.27           0.24           0.24           0.22           0.22
        option    enhancement       category        curate          diva  heterogeneity
          0.21           0.20           0.20           0.20           0.20           0.20
          lisa   longitudinal   multiplicity parametrically         convey          naked
          0.20           0.20           0.20           0.20           0.20           0.20
     personnel      translate
          0.20           0.20
```

Text Summarization: Wordcloud

- 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/

Text Summarization: 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

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

Text Summarization: Wordcloud

- Word cloud for "nlp" dataset

```
# Word cloud pal <- brewer.pal(12, "Paired")
wordcloud(names(nlp_word_count)[1:200],
            nlp_word_count[1:200],
            min.freq=10,
            scale = c(3, 1),
            rot.per = 0.1,
            col=pal,
            random.order=F)
```

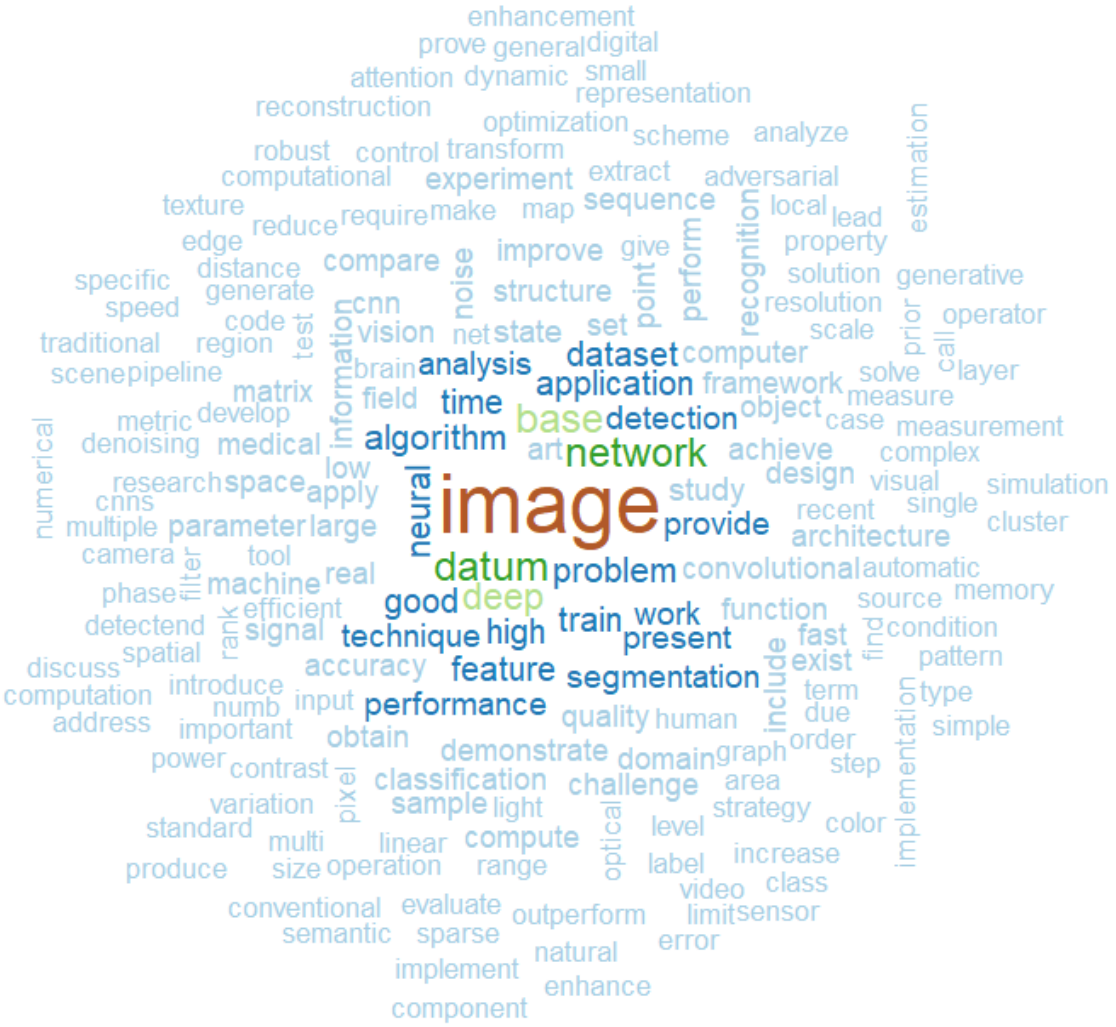
- ✓ Draw a wordcloud with top 200 words
- ✓ Each word must appear at least 10 times
- ✓ scale option controls the size of words in the wordclouds
- ✓ rot.per controls the ratio of words that are rotated

Text Summarization: Wordcloud

- Word cloud for "nlp" dataset

Text Summarization: Wordcloud

- Word cloud for "image" dataset



Text Summarization: Treemap

- Treemap

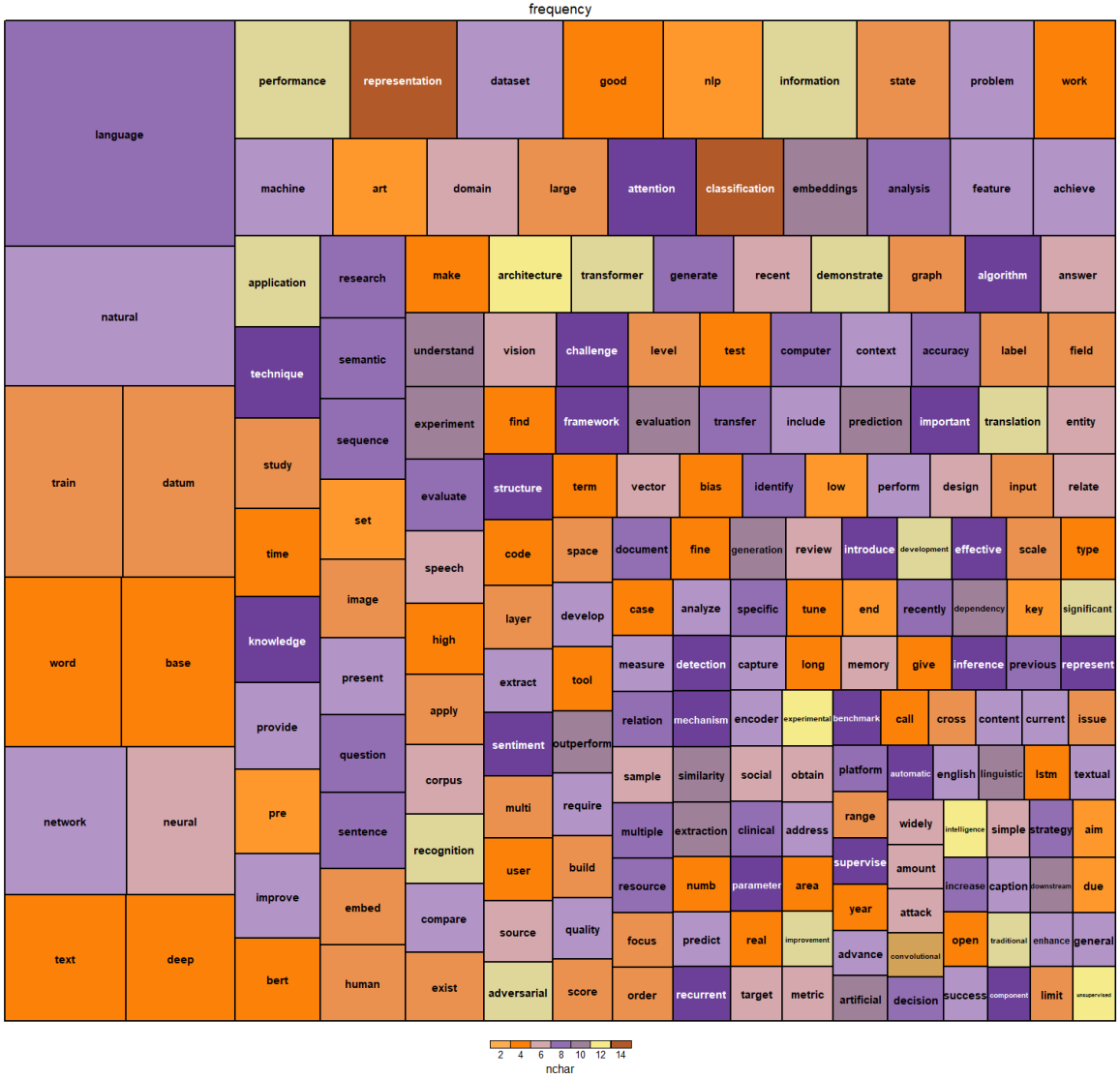
```
# Treemap
nlp_treemap_data <- data.frame(names(nlp_word_count), nlp_word_count,
nchar(names(nlp_word_count)))
names(nlp_treemap_data) <- c("word", "frequency", "nchar")
str(nlp_treemap_data)

treemap(nlp_treemap_data[1:200,],
        index = "word",
        vSize = "frequency",
        vColor = "nchar",
        type = "value",
        palette = brewer.pal(12, "Paired"))
```

✓ Size of rectangle: frequency of words

✓ Color: number of characters in each word

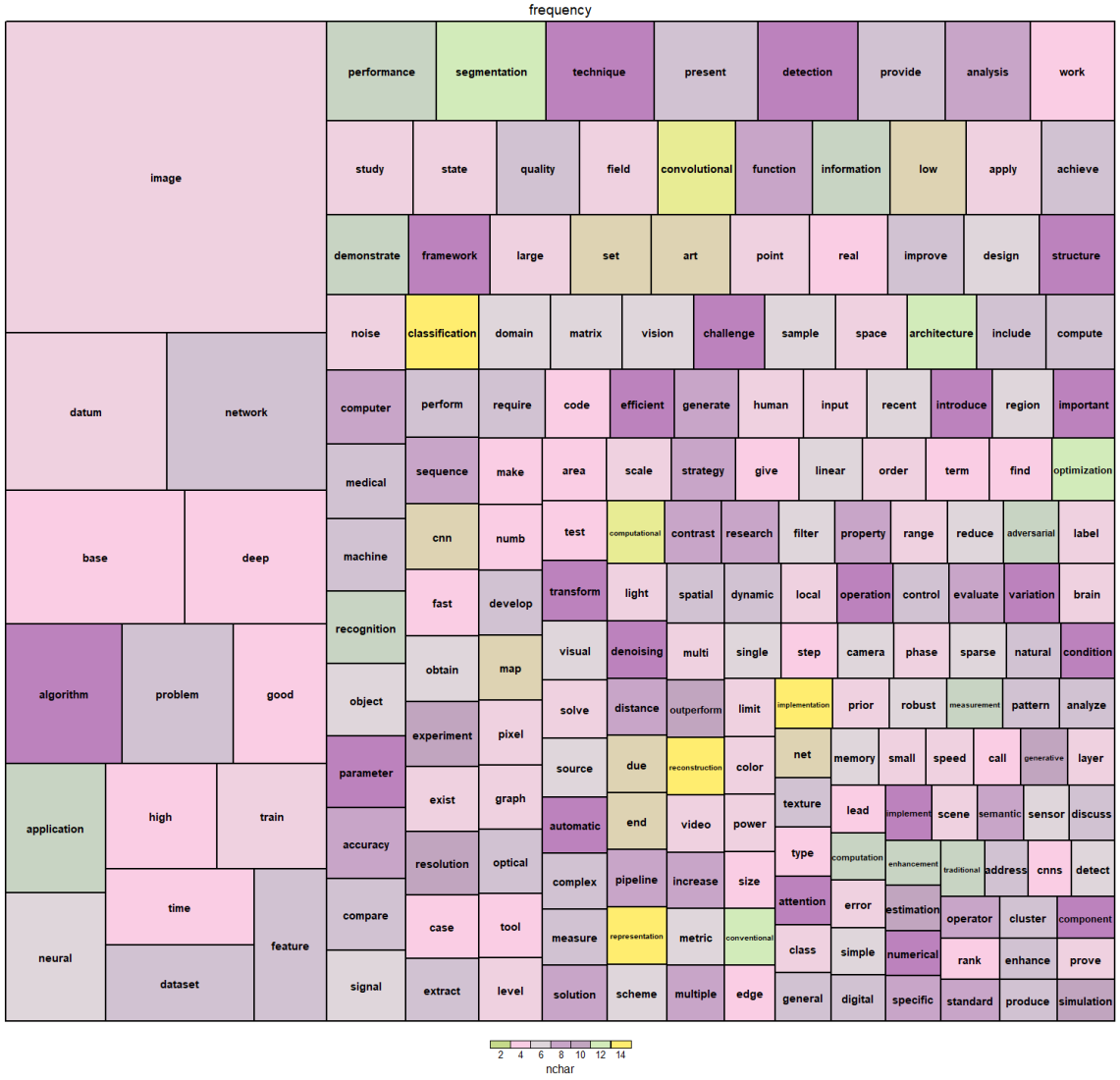
Text Summarization: Treemap

- Treemap for "nlp" dataset



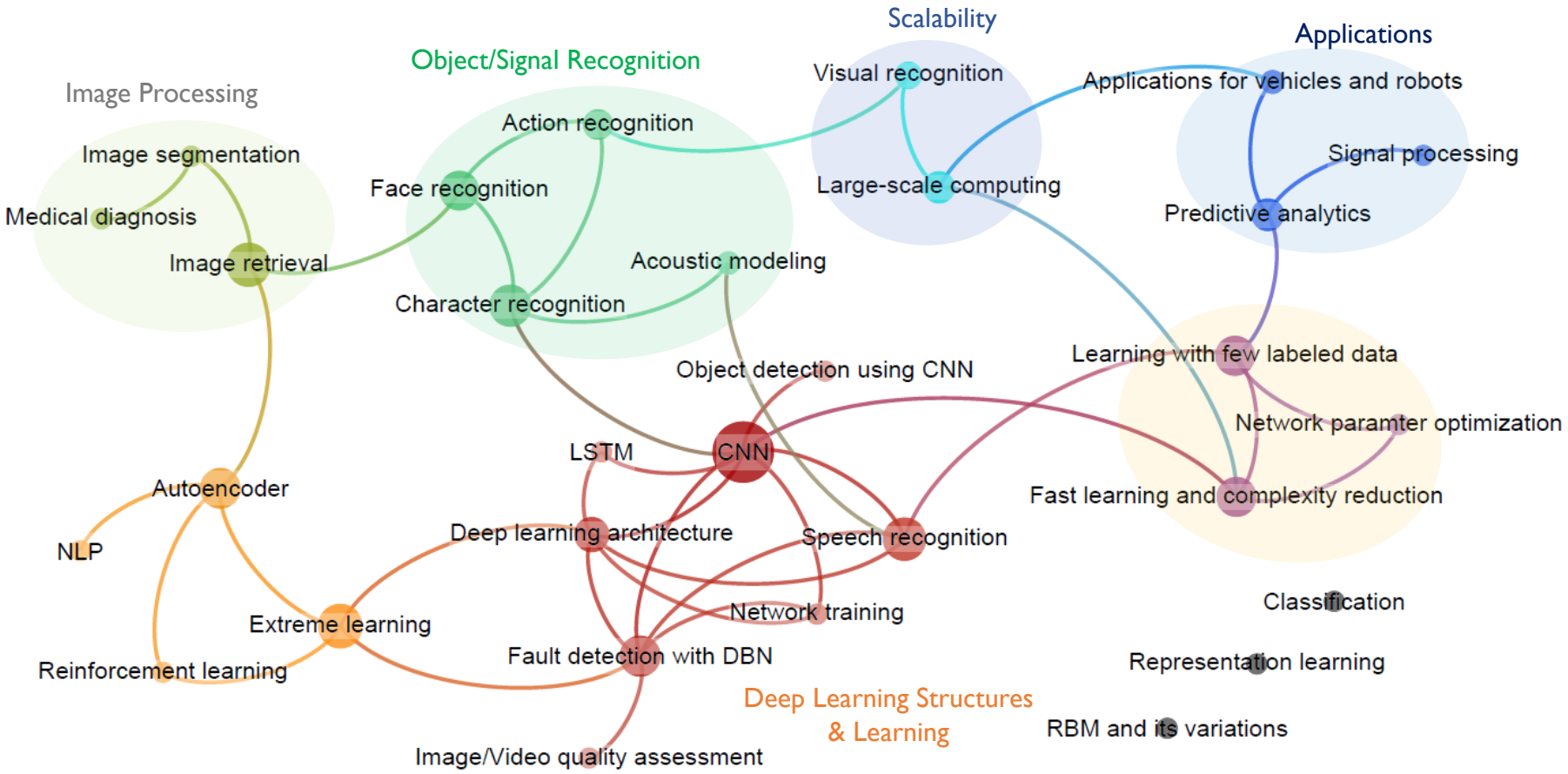
Text Summarization: Treemap

- Treemap for "image" dataset



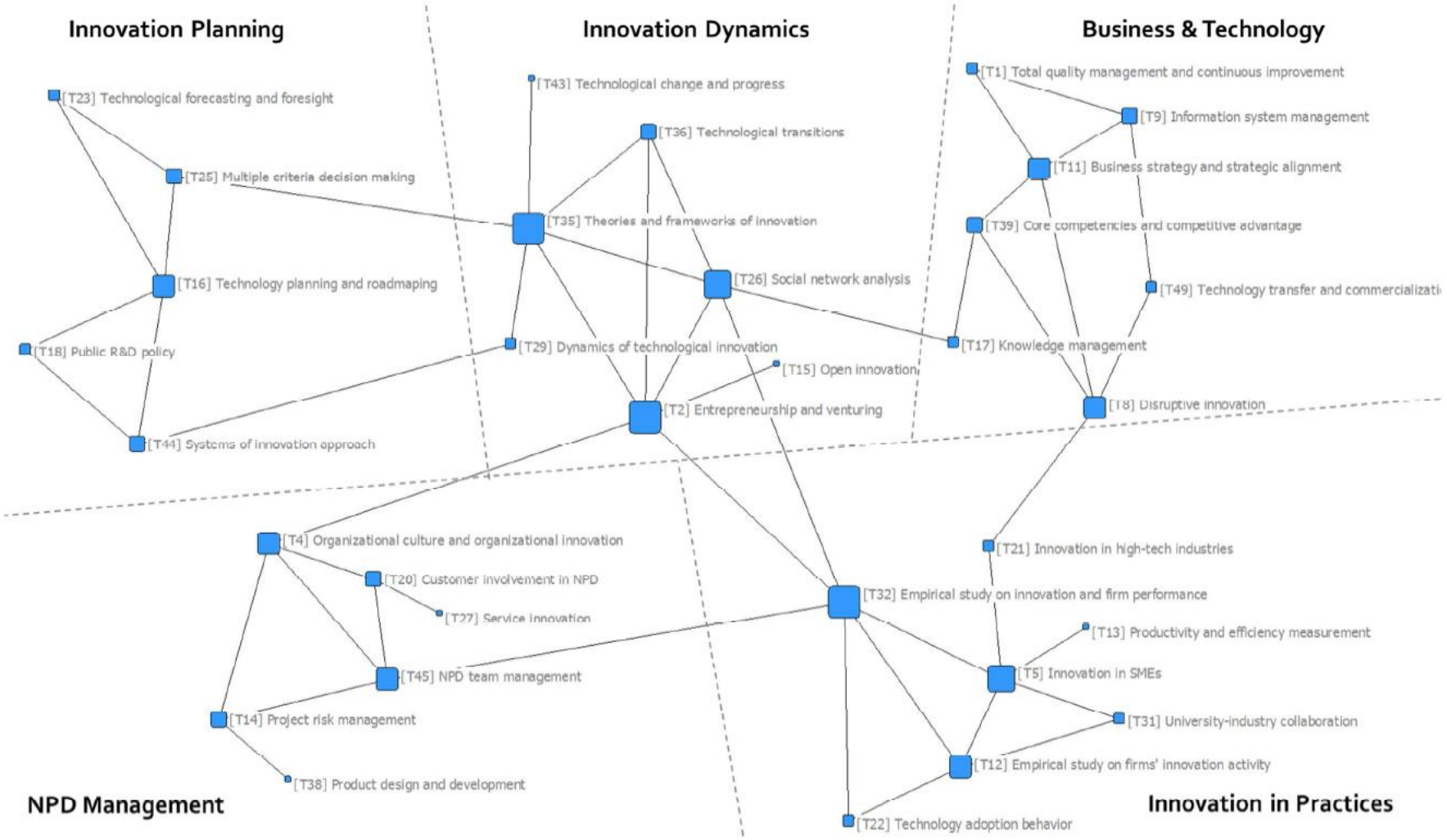
Text Summarization: Word Network

- Network of Deep Learning Topics Until Feb. 2016



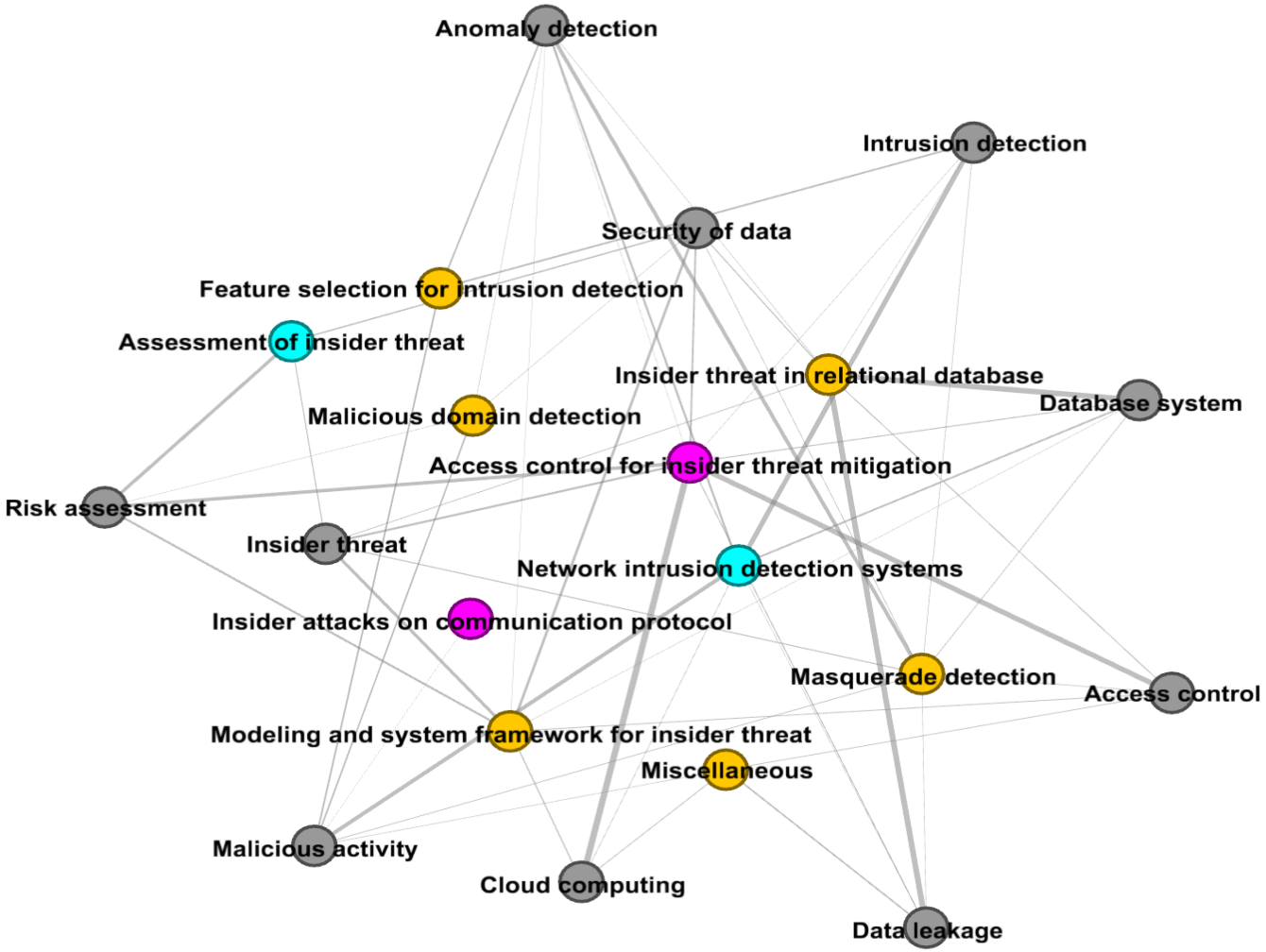
Text Summarization: Word Network

- Network of research topics for "technology and innovation management"



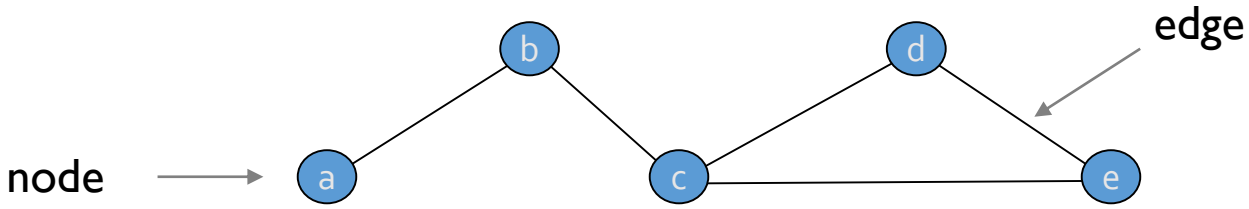
Text Summarization: Word Network

- Network of research topics for "Insider threats"



Text Summarization: Word Network

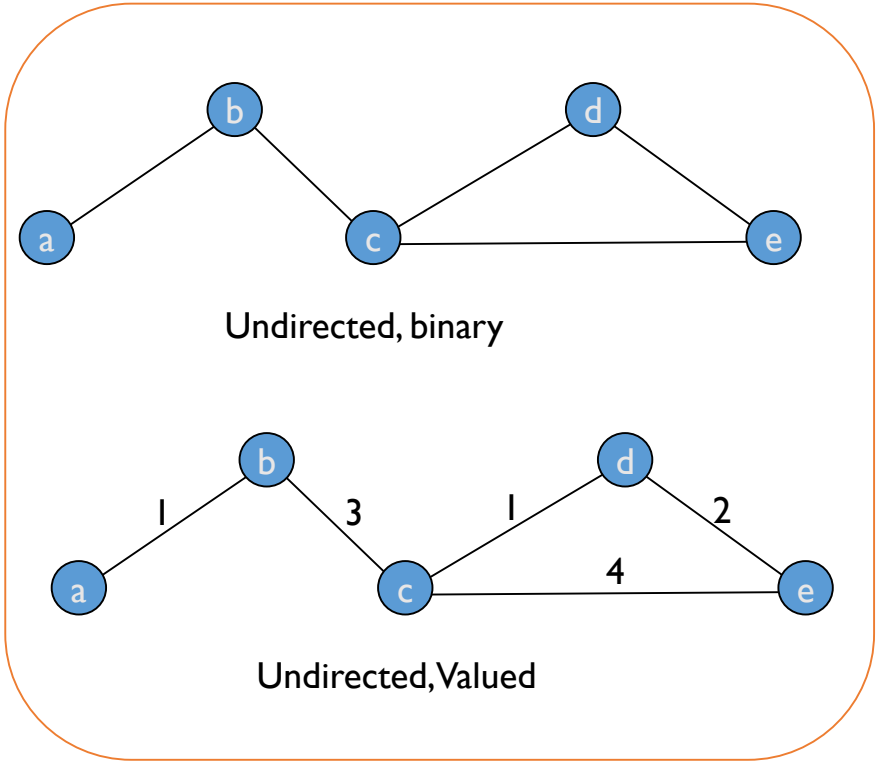
- What is a Network?
  - ✓ A network is a combined set of nodes connected by edges
  - ✓ Network ≡ Graph

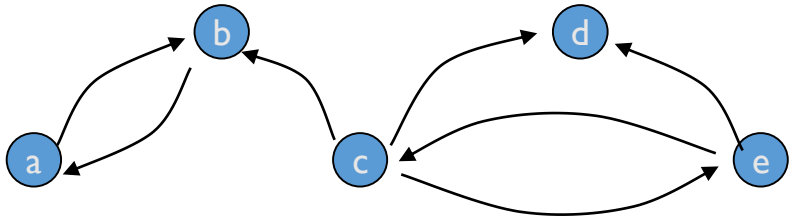


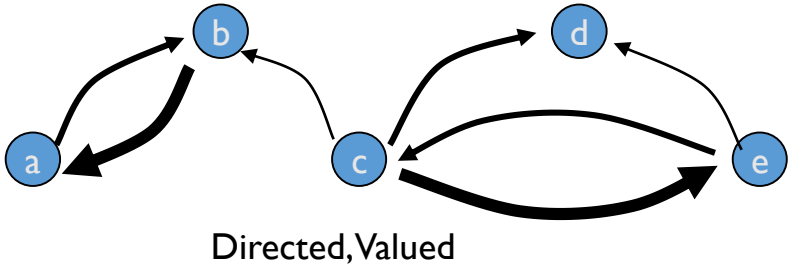
Points	Lines	Domain
Vertices	Edges, arcs	Math
Nodes	Links	Computer Science
Sites	Bonds	Physics
Actors	Ties, relations	Sociology
Keyword	Co-occurrence	Text Mining

Text Summarization: Word Network

- Type of Connections
  - ✓ A relation can be binary or valued, directed or undirected



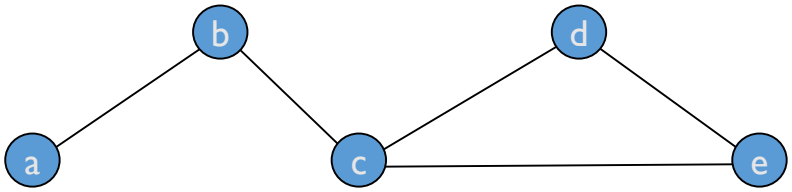




"Undirected networks are commonly used for Text Mining"

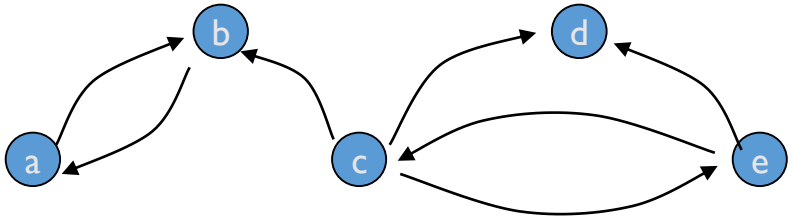
Text Summarization: Word Network

- Basic Data Structure
  - ✓ From pictures to matrices



Undirected, binary

	a	b	c	d	e
a		1			
b	1		1		
c		1		1	1
d			1		1
e			1	1	

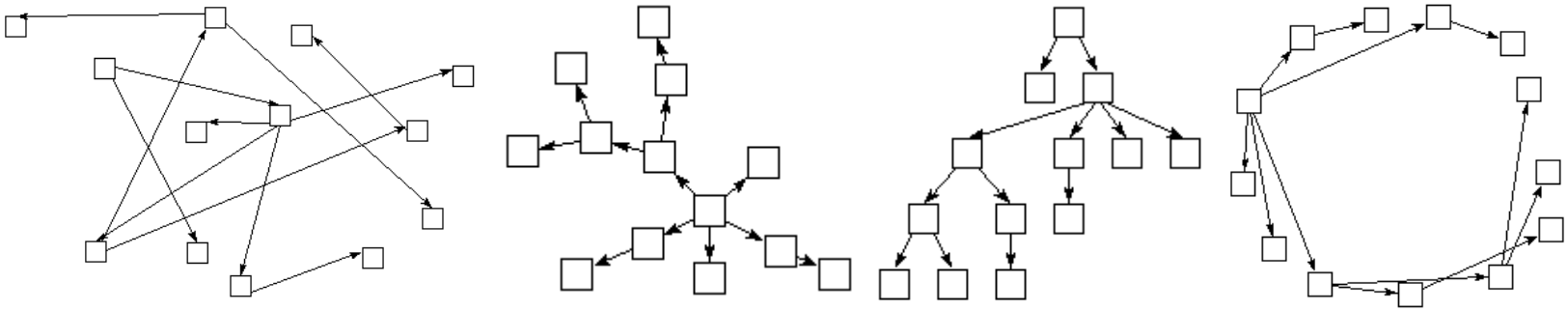


Directed, binary

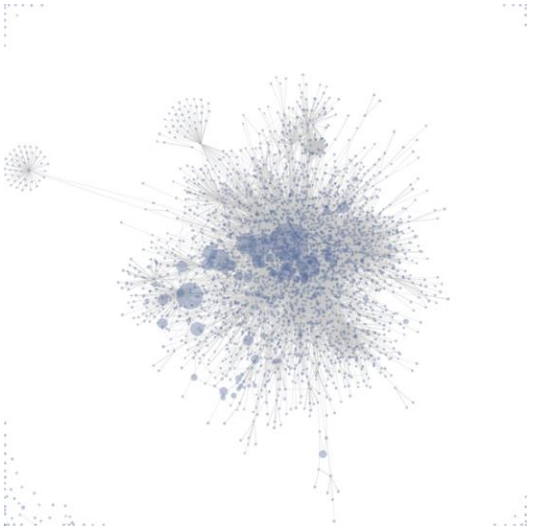
	a	b	c	d	e
a		1			
b	1				
c		1		1	1
d					
e			1	1	

Text Summarization: Word Network

- Graphical Representation
  - ✓ No standard way to draw a sociogram
    - Each of these are equal



  - ✓ Force-directed layout (Fruchterman and Reingold, 1991)
    - Distribute the vertices evenly in the frame
    - Minimize edge crossings
    - Make edge lengths uniform
    - Reflect inherent symmetry
    - Conform to the frame



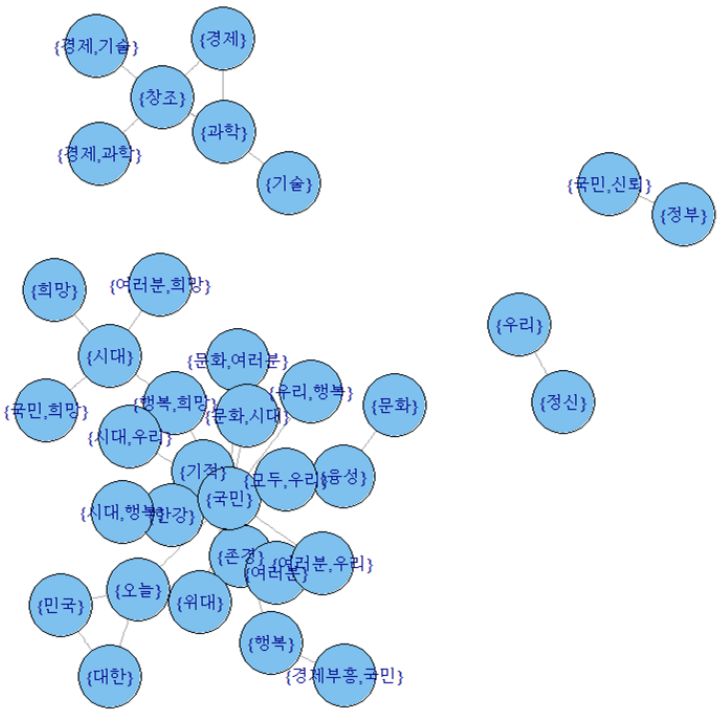
Text Summarization: Word Network

- Measuring Network: Network Level
  - ✓ Size
    - Number of nodes
  - ✓ Density
    - Number of ties that are present
  - ✓ Out-degree (directed network)
    - Sum of connections from an actor to other
  - ✓ In-degree (undirected network)
    - Sum of connections to an actor

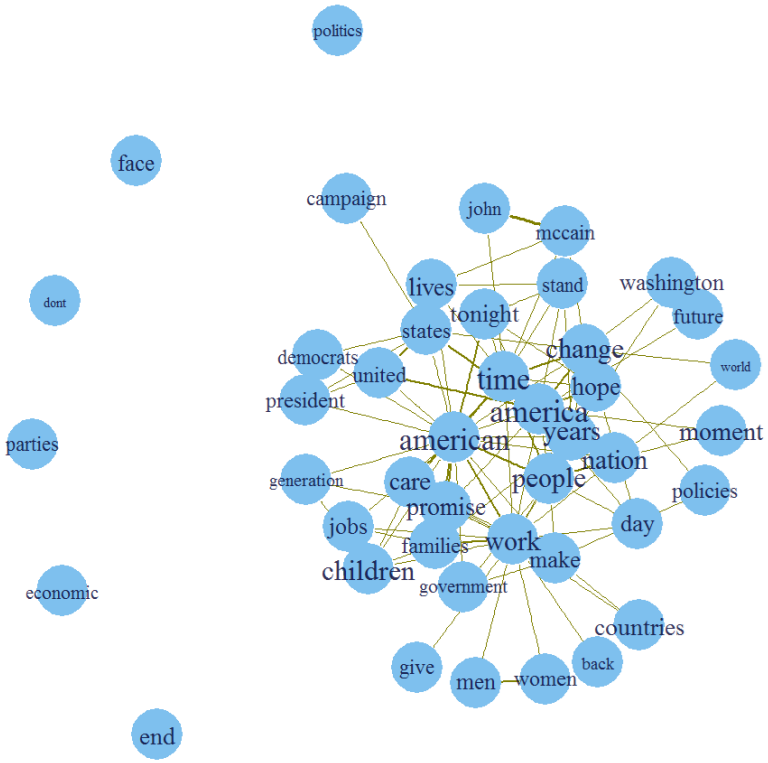
Text Summarization: Word Network

- Community Structure
  - ✓ Nodes of a network can be grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally.

Four non-overlapping communities

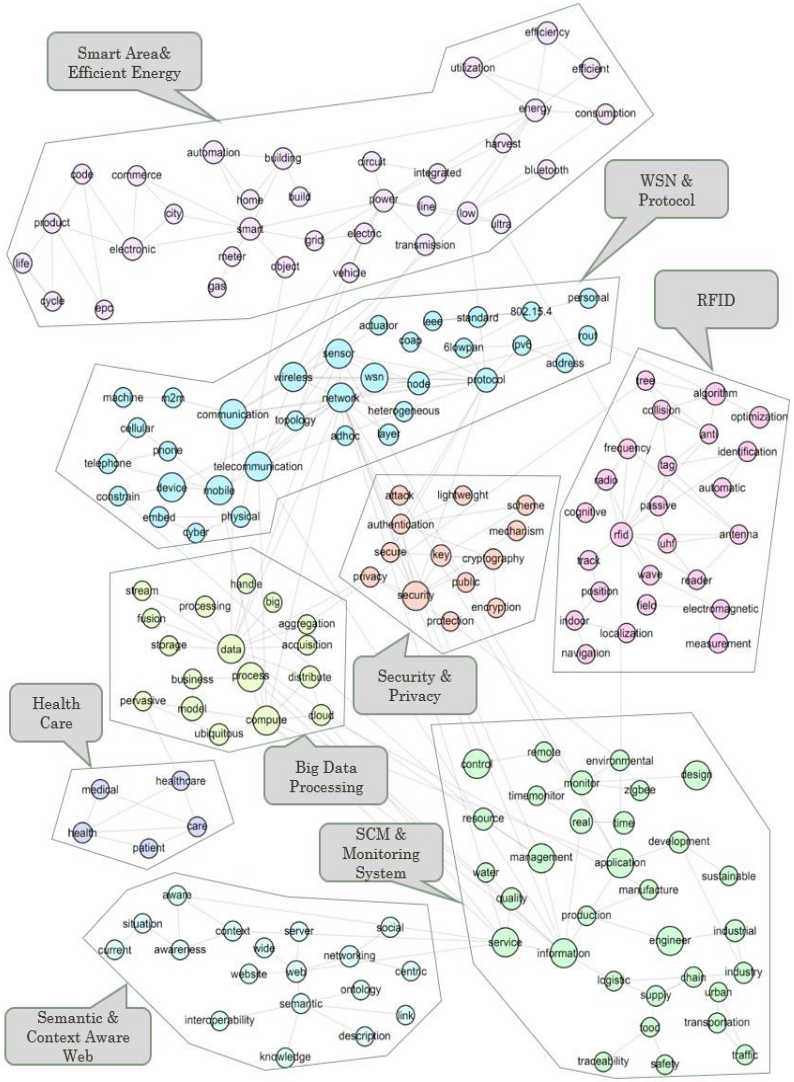


Difficult to identify the community structure



Text Summarization: Word Network

- Keyword network of Research Papers for "Internet of Things"



Text Summarization: Word Network

- Conduct some processing to draw word networks
  - ✓ Convert the frequency of each TDM to binary representation
  - ✓ Remove loops, delete edges whose edge weights are smaller than 50, etc.

```
# Construct a word network for the nlp papers
# Change it to a Boolean matrix
nlpTDM_matrix[nlpTDM_matrix >= 1] <- 1
nlp_adj_mat <- nlpTDM_matrix %*% t(nlpTDM_matrix)

# inspect terms numbered 1 to 10
nlp_adj_mat[1:10,1:10]

# Build a graph from the above matrix
nlp_graph <- graph.adjacency(nlp_adj_mat, weighted=T, mode = "undirected")
# remove loops
nlp_graph <- simplify(nlp_graph)
# set labels and degrees of vertices
V(nlp_graph)$label <- V(nlp_graph)$name
nlp_graph <- delete.edges(nlp_graph, which(E(nlp_graph)$weight <= 50))
nlp_graph <- delete.vertices(nlp_graph, which(degree(nlp_graph) == 0))
V(nlp_graph)$degree <- degree(nlp_graph)
```

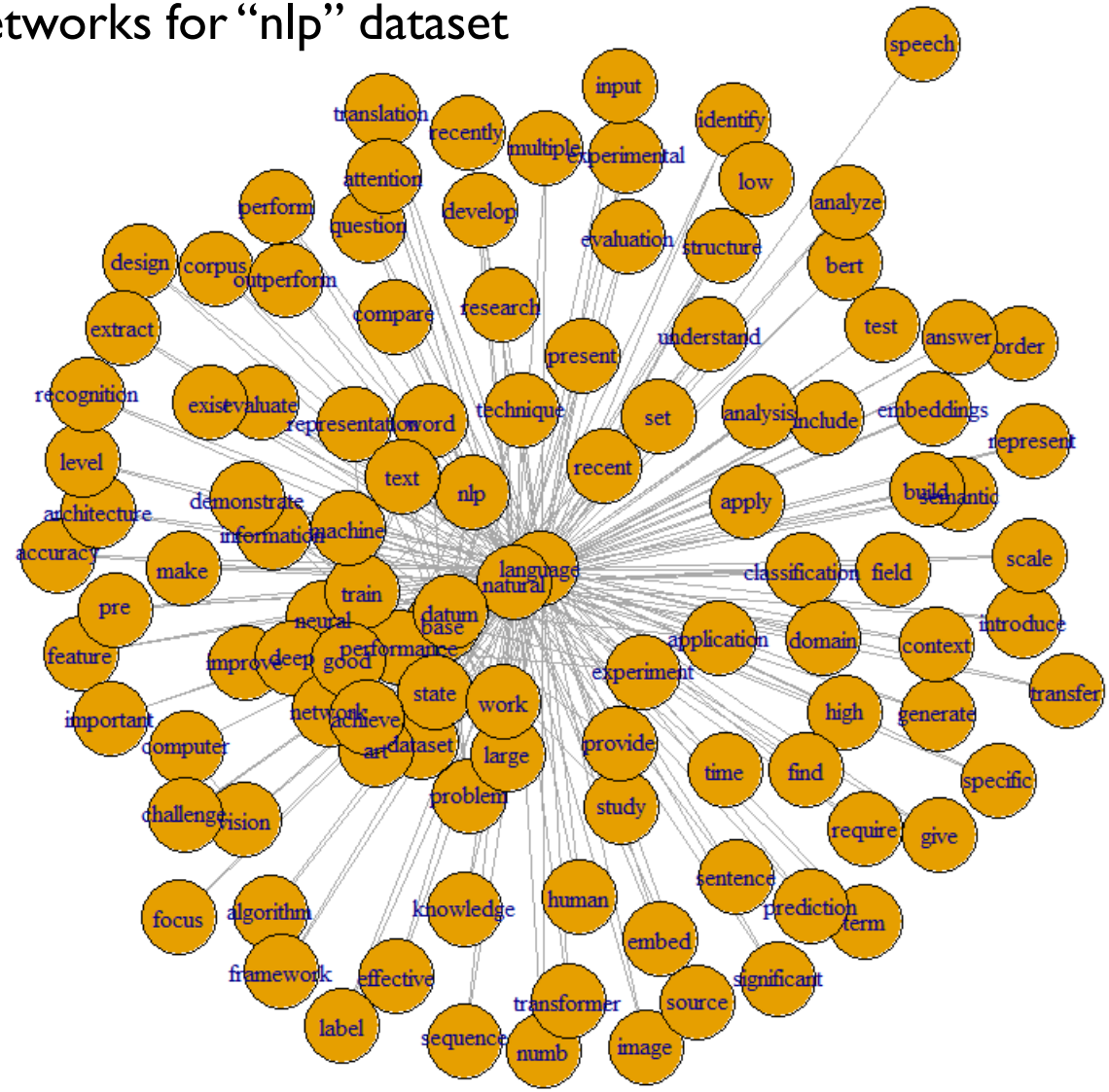
Text Summarization: Word Network

- Plot some networks for "nlp" dataset

```
# set seed to make the layout reproducible
set.seed(3952)
plot(nlp_graph, layout=layout.fruchterman.reingold)
plot(nlp_graph, layout=layout.kamada.kawai, vertex.size = 10,
         vertex.color = 8, vertex.label.cex = 1)
```

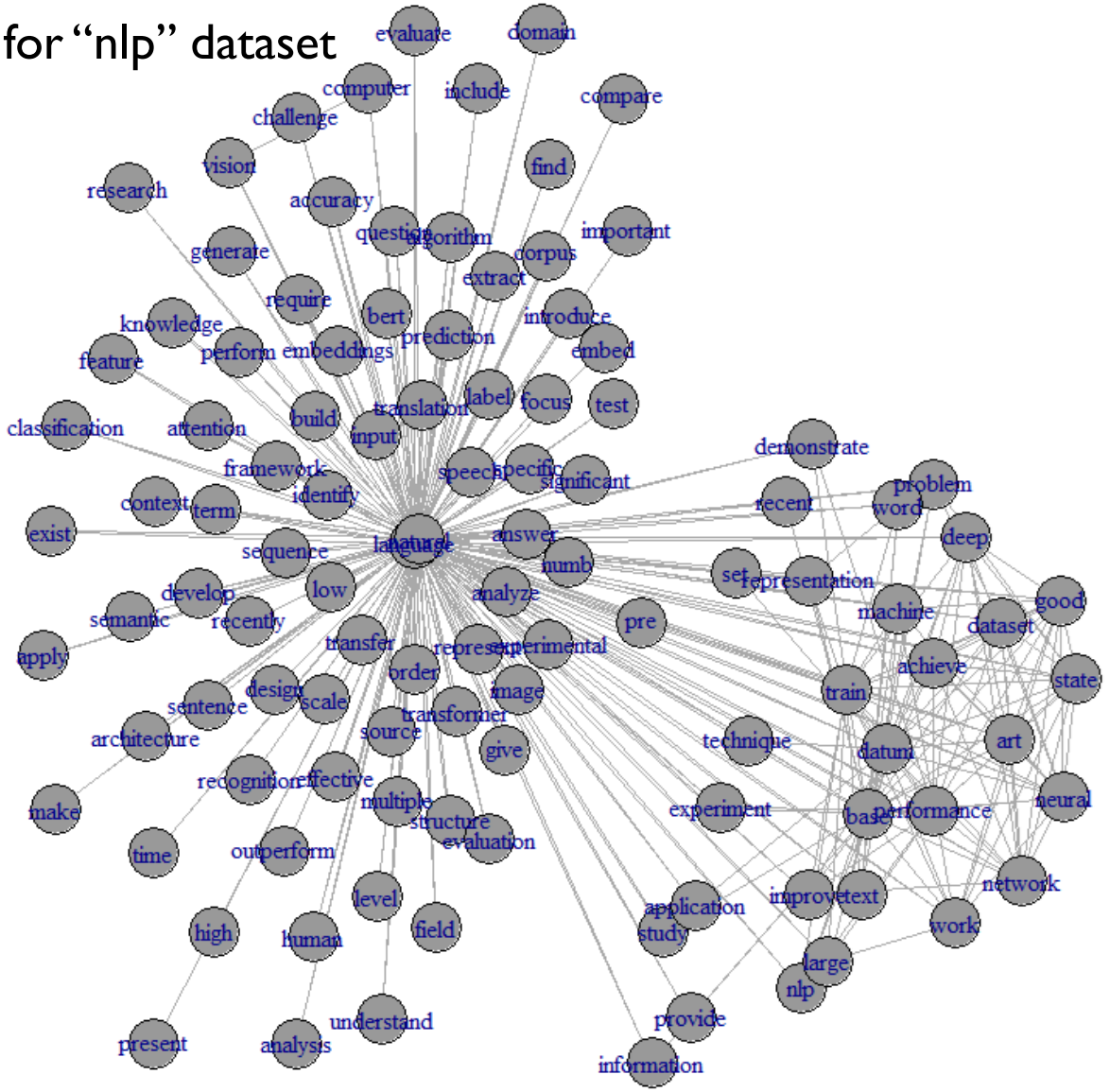
Text Summarization: Word Network

- Plot some networks for "nlp" dataset



Text Summarization: Word Network

- Plot some networks for "nlp" dataset



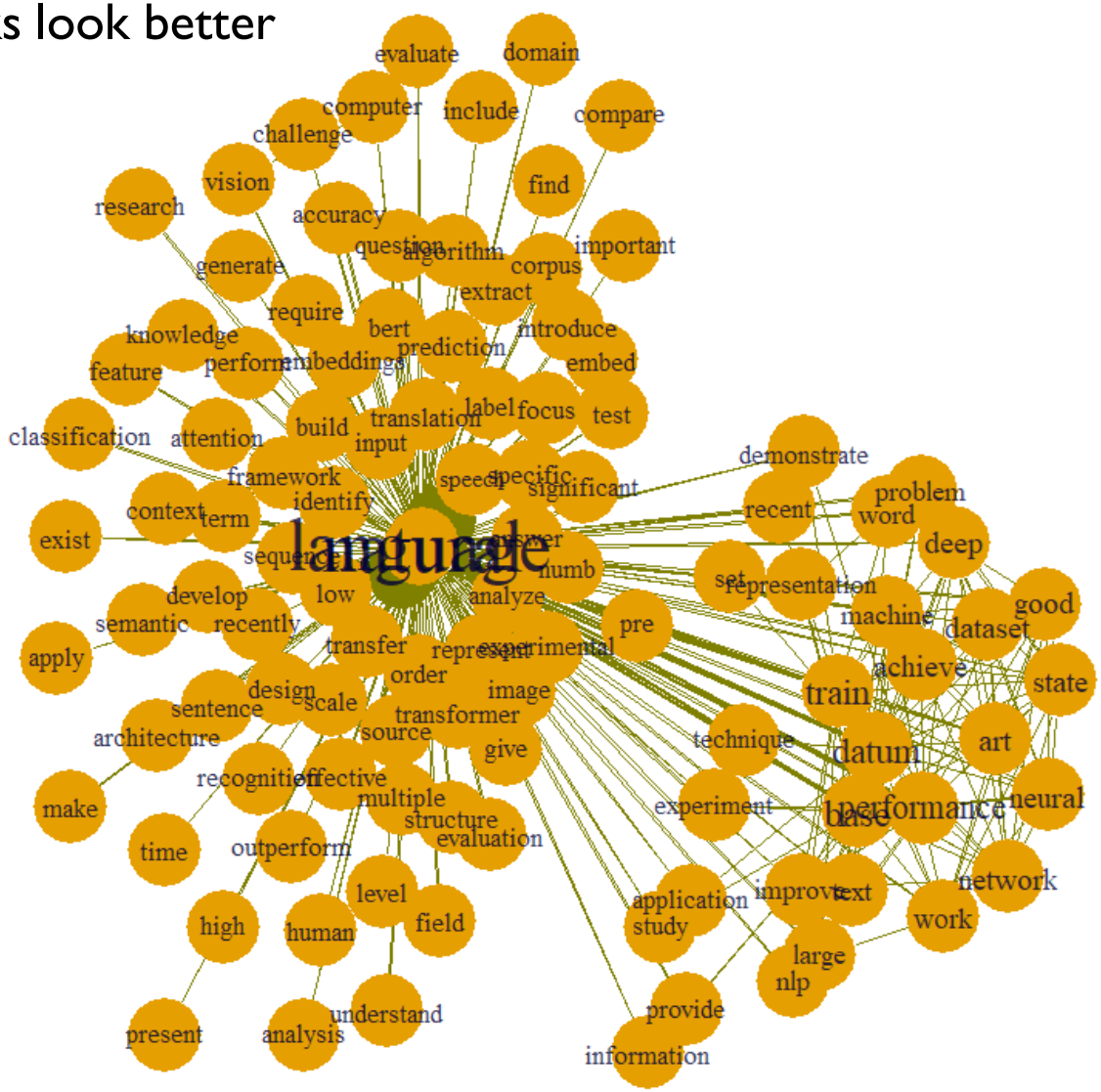
Text Summarization: Word Network

- Make networks look better

```
# Make the network look better
V(nlp_graph)$label.cex <- 2*V(nlp_graph)$degree/max(V(nlp_graph)$degree)+1
V(nlp_graph)$label.color <- rgb(0, 0, 0.2, 0.8)
V(nlp_graph)$frame.color <- NA
egam1 <- 300*(exp(E(nlp_graph)$weight/100))/sum(exp(E(nlp_graph)$weight/100))
E(nlp_graph)$color <- rgb(0.5, 0.5, 0)
E(nlp_graph)$width <- egam1

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

Text Summarization: Word Network

- Make networks look better



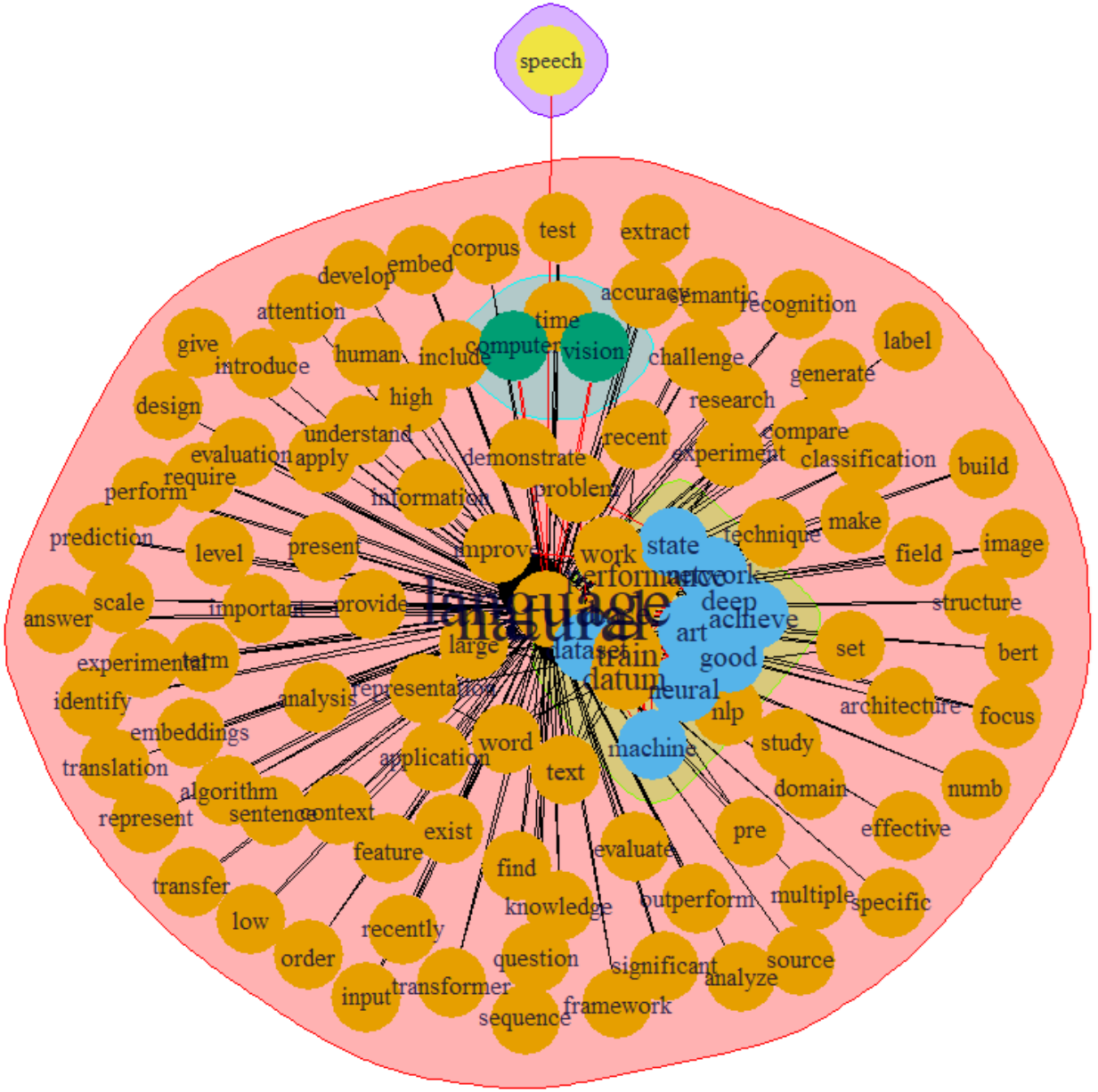
Text Summarization: Word Network

- Find community

```
# Find community
nlp_community <- walktrap.community(nlp_graph, steps = 3)
modularity(nlp_community)
membership(nlp_community)
plot(nlp_community, nlp_graph)
```

Text Summarization: Word Network

- Find community



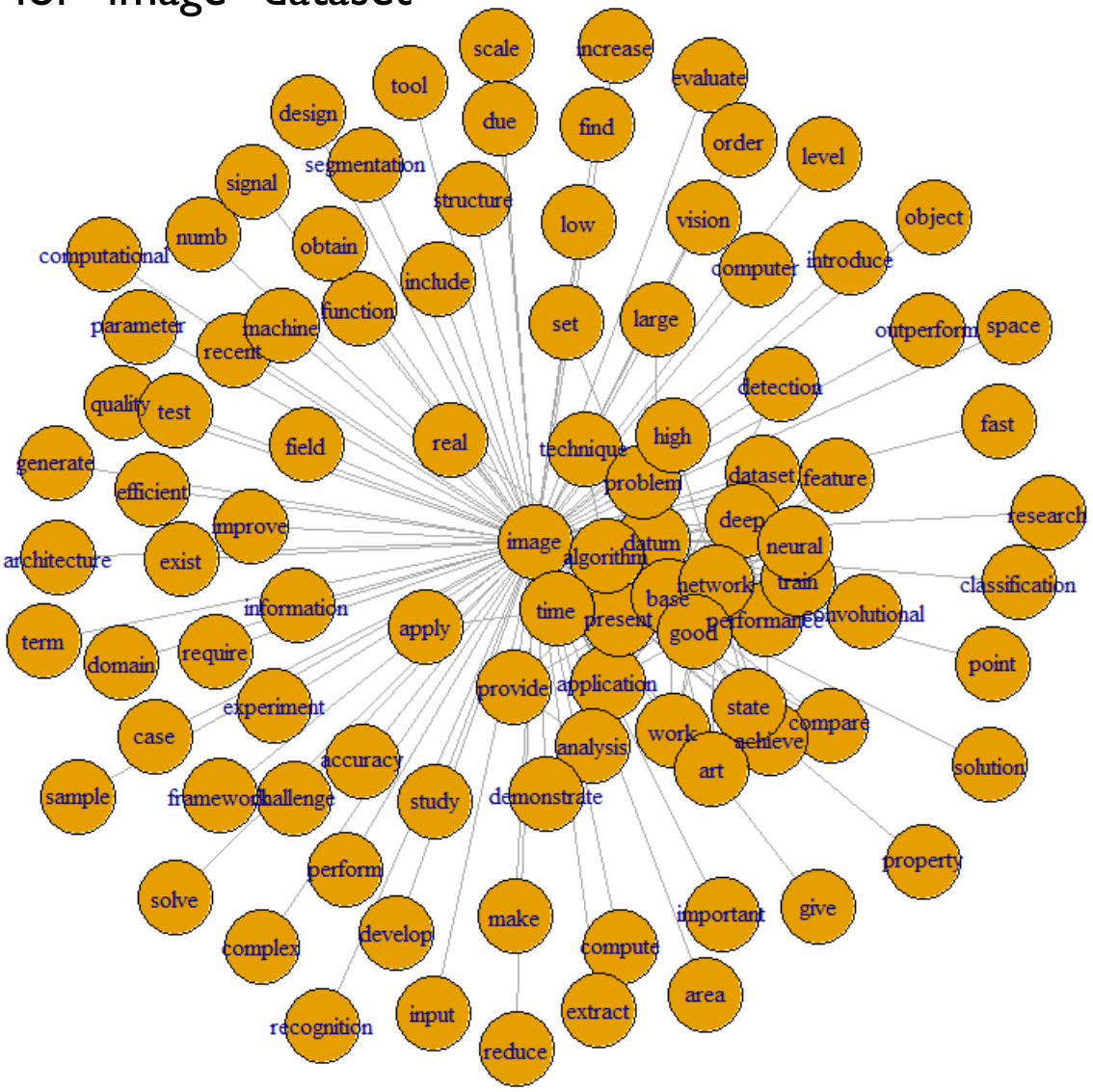
Text Summarization: Word Network

- Plot some networks for "image" dataset

```
# set seed to make the layout reproducible
set.seed(3952)
plot(image_graph, layout=layout.fruchterman.reingold)
plot(image_graph, layout=layout.kamada.kawai, vertex.size = 10,
         vertex.color = 8, vertex.label.cex = 1)
```

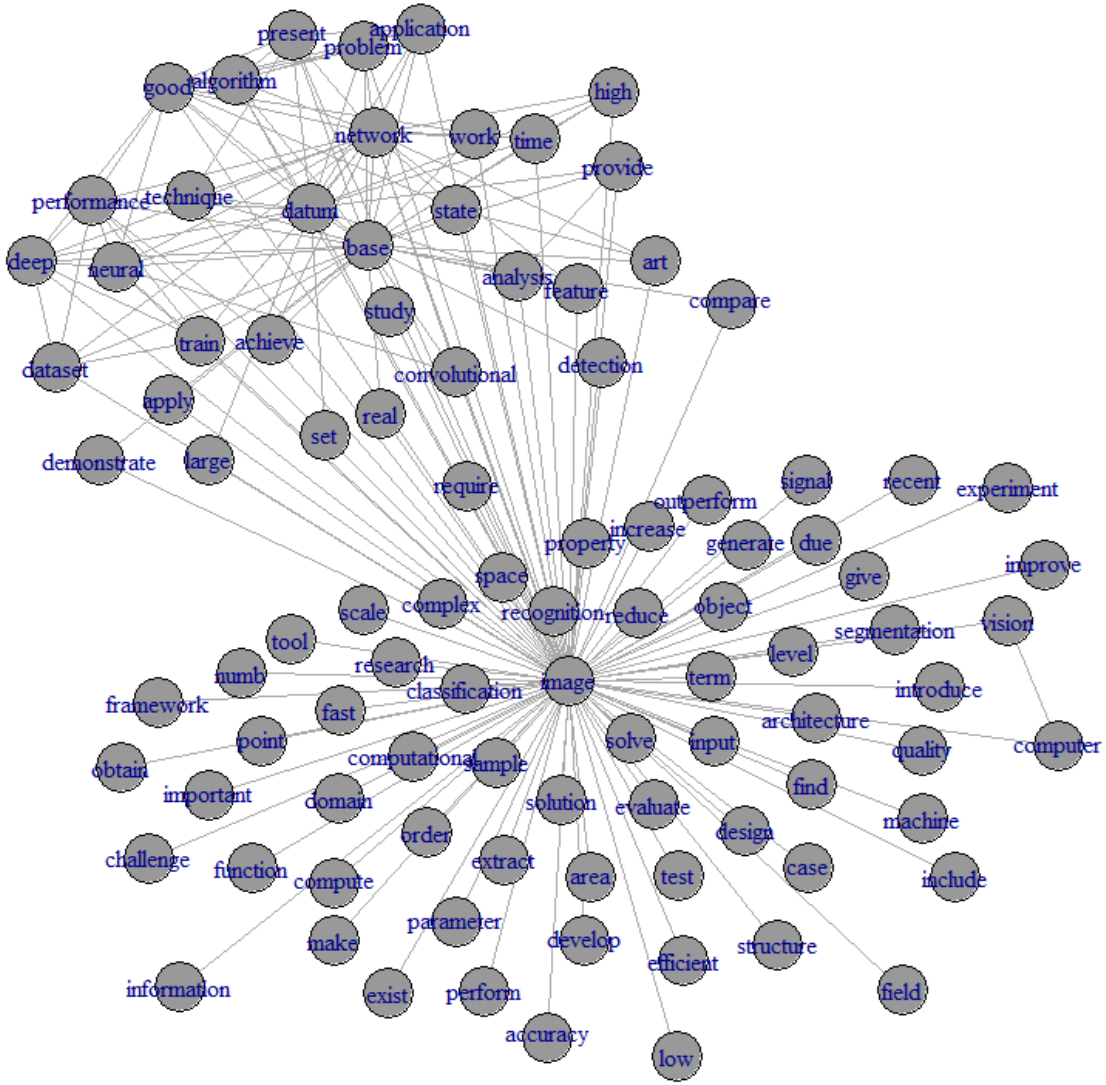
Text Summarization: Word Network

- Plot some networks for "image" dataset



Text Summarization: Word Network

- Plot some networks for "image" dataset



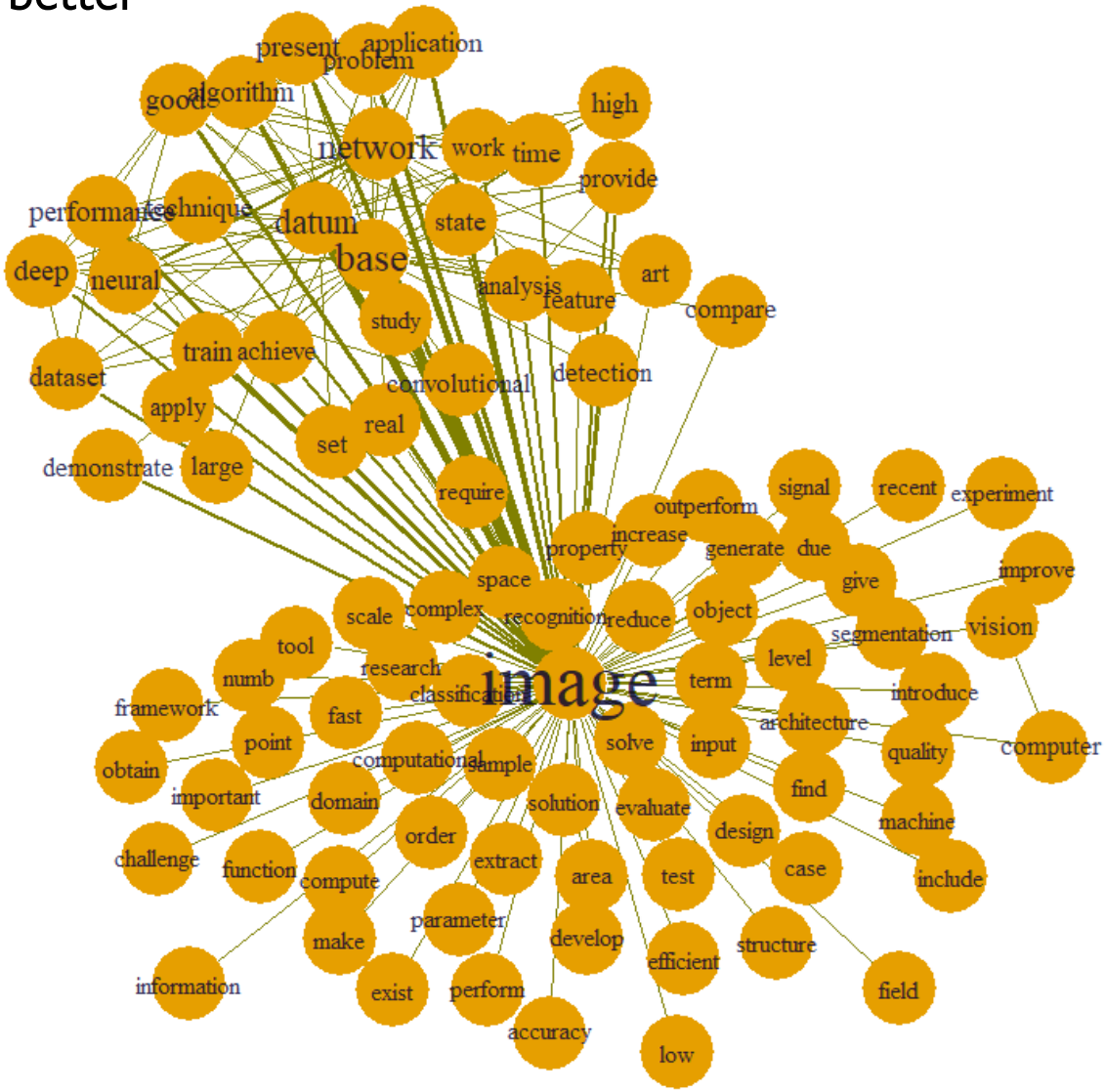
Text Summarization: Word Network

- Make networks look better

```
# Make the network look better
V(image_graph)$label.cex <-
2*V(image_graph)$degree/max(V(image_graph)$degree)+1
V(image_graph)$label.color <- rgb(0, 0, 0.2, 0.8)
V(image_graph)$frame.color <- NA
egam1 <-
300*(exp(E(image_graph)$weight/100))/sum(exp(E(image_graph)$weight/100))
E(image_graph)$color <- rgb(0.5, 0.5, 0)
E(image_graph)$width <- egam1

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

Text Summarization: Word Network

- Make networks look better



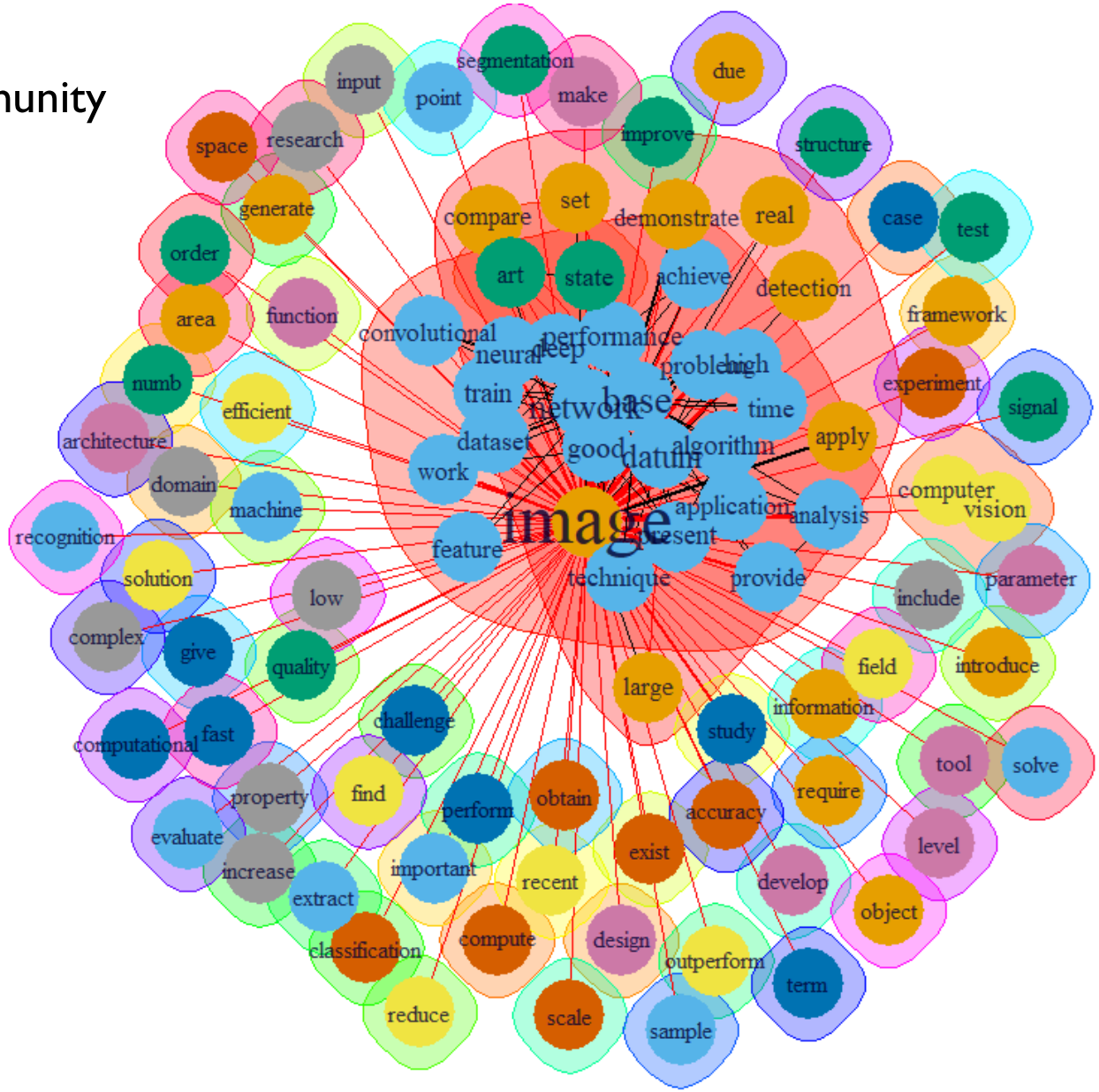
Text Summarization: Word Network

- Find community

```
# Find community
image_community <- walktrap.community(image_graph, steps = 3)
modularity(image_community)
membership(image_community)
plot(image_community, image_graph)
```

Text Summarization: Word Network

- Find community




ANY
questions?