

CSCI446/946 Big Data Analytics

Week 8 Advanced Analytical Theory and Methods: Text Analysis

School of Computing and Information Technology
University of Wollongong Australia

Advanced Analytical Theory and Methods: **Text Analysis**

- Overview of Text Analysis
- Collecting and Representing Text
- Term Frequency --- Inverse Document Frequency (TFIDF)
- Categorizing Documents by Topics
- Determining Sentiments
- Gaining Insights

All the figures, tables and codes are from the book “[Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data](#)” unless indicated otherwise.

Overview of Text Analysis

- Text analysis (text analytics)
 - Refers to the **representation**, **processing**, and **modelling** of textual data to derive useful insights
 - Suffers from the curse of **high dimensionality**
 - Most of the time the text is **not structured**
- **Corpus**
 - A large collection of texts (documents) used for various purposes in Natural Language Processing

Overview of Text Analysis

Corpus	Word Count	Domain	Website
Shakespeare	0.88 million	Written	http://shakespeare.mit.edu/
Brown Corpus	1 million	Written	http://icame.uib.no/brown/bcm.html
Penn Treebank	1 million	Newswire	http://www.cis.upenn.edu/~treebank/
Switchboard Phone Conversations	3 million	Spoken	http://catalog.ldc.upenn.edu/LDC97S62
British National Corpus	100 million	Written and spoken	http://www.natcorp.ox.ac.uk/
NA News Corpus	350 million	Newswire	http://catalog.ldc.upenn.edu/LDC95T21
European Parliament Proceedings Parallel Corpus	600 million	Legal	http://www.statmt.org/europarl/
Google N-Grams Corpus	1 trillion	Written	http://catalog.ldc.upenn.edu/LDC2006T13

Overview of Text Analysis

Data Source	Data Format	Data Structure Type
News articles	TXT, HTML, or Scanned PDF	Unstructured
Literature	TXT, DOC, HTML, or PDF	Unstructured
E-mail	TXT, MSG, or EML	Unstructured
Web pages	HTML	Semi-structured
Server logs	LOG or TXT	Semi-structured or Quasi-structured
Social network API firehoses	XML, JSON, or RSS	Semi-structured
Call center transcripts	TXT	Unstructured

Text Analysis Steps

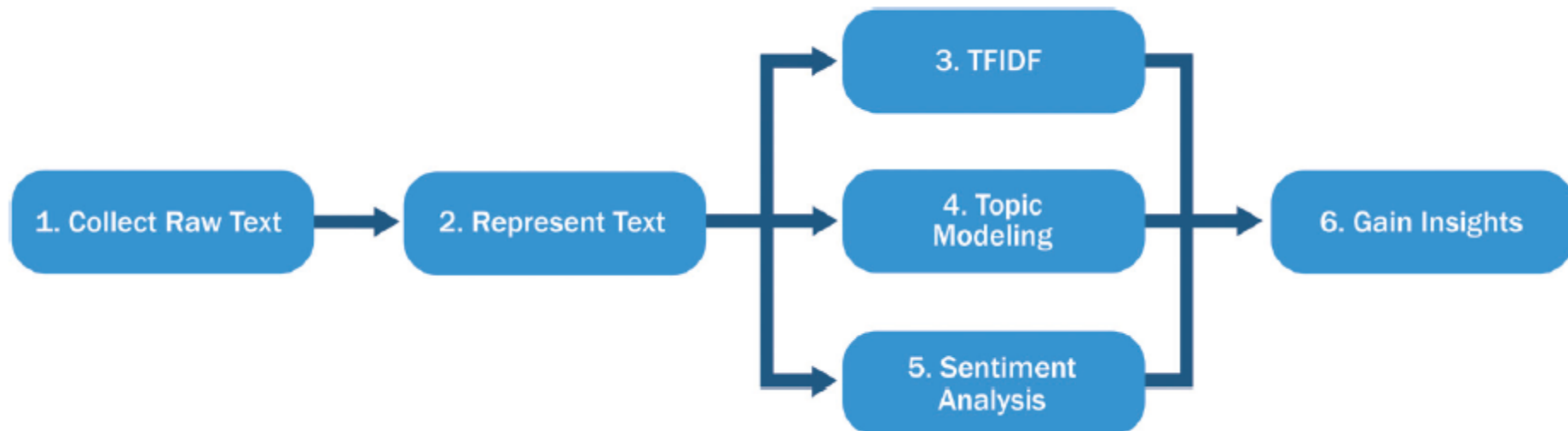
- Parsing
 - Takes unstructured text and imposes a structure for further analysis
- Search and retrieval
 - Identification of the documents in a corpus that contain search items (key terms)
- Text mining
 - Use the results of the prior steps to discover meaningful insights

Text Analysis Steps

- Text mining
 - Clustering and classification techniques can be adapted to text mining
 - K-means to cluster documents into groups
 - Naïve Bayes classifier for sentiment analysis
 - Utilises various methods and techniques
 - Statistical analysis
 - Information retrieval
 - Data mining and Natural Language Processing

A Text Analysis Example

- A company would like to **monitor what is being said** about its products in social media
 - Are people mentioning its products?
 - What is being said? Good or bad?



Collecting Raw Text

- For text analysis, data must be collected before anything can happen
- The team
 - Starts by actively monitoring various websites for user-generated contents
 - Will deal with semi-structured data
- Public APIs and Web scraper
- Be careful about the rights of the owner

Representing Text

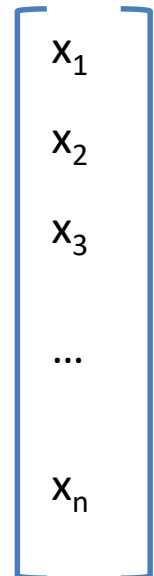
- Raw text needs to be transformed with **text normalization** techniques
- **Tokenization**
 - The task of **separating words** from the body of text
 - Tokenizing based on **spaces**
 - Tokenizing based on **punctuation marks & spaces**
 - Much more **difficult** than expected
 - **No one-size-fits-all** tokenization scheme

Representing Text

- Raw text needs to be transformed with **text normalization** techniques
- **Case folding**
 - Reduces all letters to **lowercase** (or uppercase)
 - If implemented incorrectly...
 - General Motors; WHO; US; ...
 - May need to create a lookup table of words not to be case folded

Representing Text

- **Bag-of-words** representation
 - A document becomes a **high-dimensional vector**, indicating the **presence/absence/frequency** of various words in this document



Representing Text

- Bag-of-words representation
 - A simple yet widely used to represent text
 - Represent a document as a set of terms (words), ignoring other information (such as order, context, inferences, and discourse)
 - “a dog bits a man” same as “a man bits a dog”
 - Although a naïve and over-simplified approach, it is still considered a good approach to start with

Representing Text

- Representation of a corpus
 - A corpus is a collection of documents, usually focused on specific domains
 - Some corpora include the information content of every word in its metadata
- Information content (IC)
 - A metric denotes the importance of a term in a corpus.
 - Terms with higher IC values are more important

Representing Text

- However, information content (IC)
 - Cannot satisfy the need to analyse the dynamically changed, unstructured data
- Two problems
 - Both traditional corpora and IC metadata do not change over time
 - Traditional corpora limits the entire knowledge used for a text analysis algorithm

Term Frequency – Inverse Document Frequency (TFIDF)

- So, we need a metric that adapts to the context and the nature of text (not like IC)
- TFIDF is based entirely on all the fetched documents
- TFIDF can be easily updated once the fetched documents change
- TFIDF is a measure widely used in text analysis

Term Frequency

- Given a term t and a document $d = \{t_1, t_2, \dots, t_n\}$
- Term frequency of t in d is defined as the number of times t appearing in d

$$TF_1(t, d) = \sum_{i=1}^n f(t, t_i) \quad t_i \in d; |d| = n$$

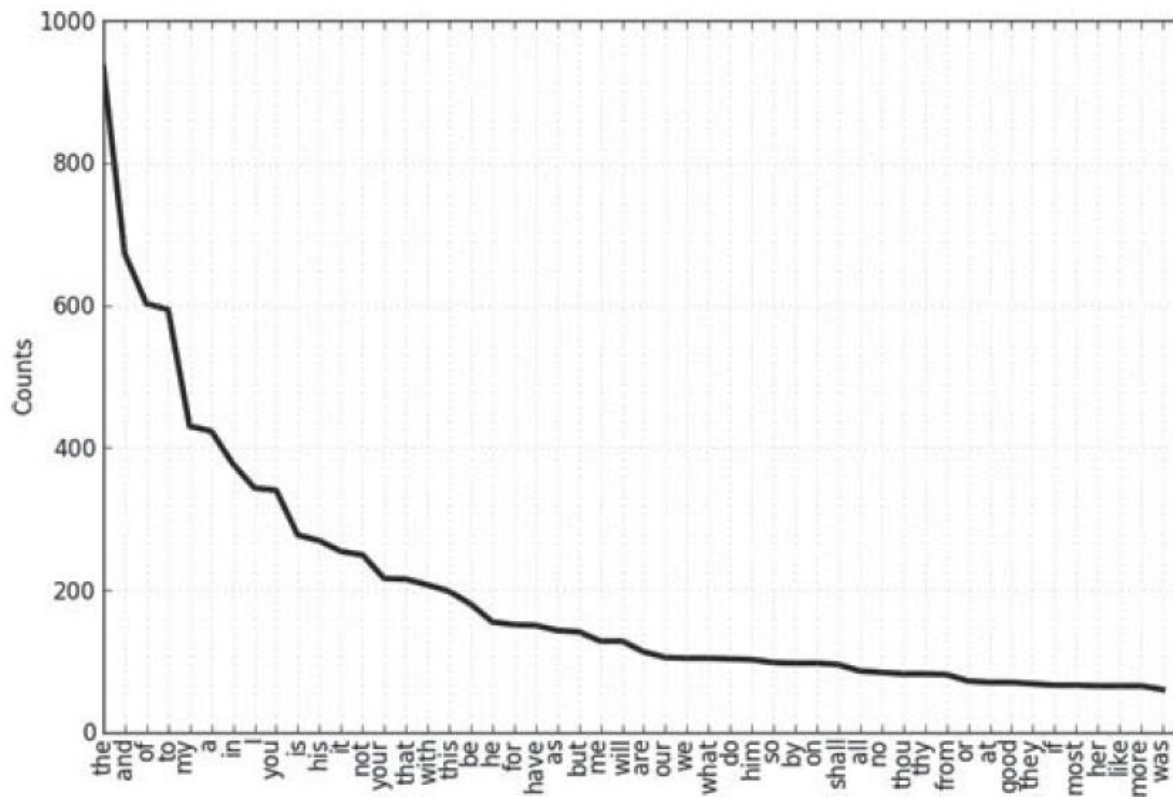
$$f(t, t') = \begin{cases} 1, & \text{if } t = t' \\ 0, & \text{otherwise} \end{cases}$$

$$TF_2(t, d) = \log[TF_1(t, d) + 1]$$

$$TF_3(t, d) = \frac{TF_1(t, d)}{n} \quad |d| = n$$

Term Frequency

- **Zipf's Law**: the i -th most common word occurs approximately $1/i$ as the most frequent term



Term Frequency

- Stop words
 - Not all the words from a given language need to be computed for term frequency
 - “the, a, of, and, to, ...” which are not likely to contribute to semantic understanding
- Terms that do not appear in a document or even a corpus need not to be computed
- Lemmatization and stemming

Term Frequency

- An issue with Term Frequency
 - The importance of a term is solely based on its presence within a particular document
 - What if this term frequently appears in EVERY document? Is it still important?
- We need to have a broader view of the world
 - Consider the importance of a term not only in a single document but also in a corpus

Inverse Document Frequency

- **Document Frequency** of a term
 - The number of documents in a corpus that contain a term
- Let a corpus $D = \{d_1, d_2, \dots, d_N\}$

$$DF(t) = \sum_{i=1}^N f'(t, d_i) \quad d_i \in D; |D| = N$$

$$f'(t, d') = \begin{cases} 1, & \text{if } t \in d' \\ 0, & \text{otherwise} \end{cases}$$

Inverse Document Frequency

- Inverse Document Frequency of a term

$$IDF_1(t) = \log \frac{N}{DF(t)} \quad IDF_2(t) = \log \frac{N}{DF(t) + 1}$$

- Stop words have higher TF and higher DF
- The IDF of a rare term would be high
- The IDF of a frequent term would be low
- IDF solely depends on the DF

Term Frequency – Inverse Document Frequency (TFIDF)

- A measure that **considers**
 - The prevalence of a term within a document (**TF**)
 - The scarcity of the term over the corpus (**IDF**)

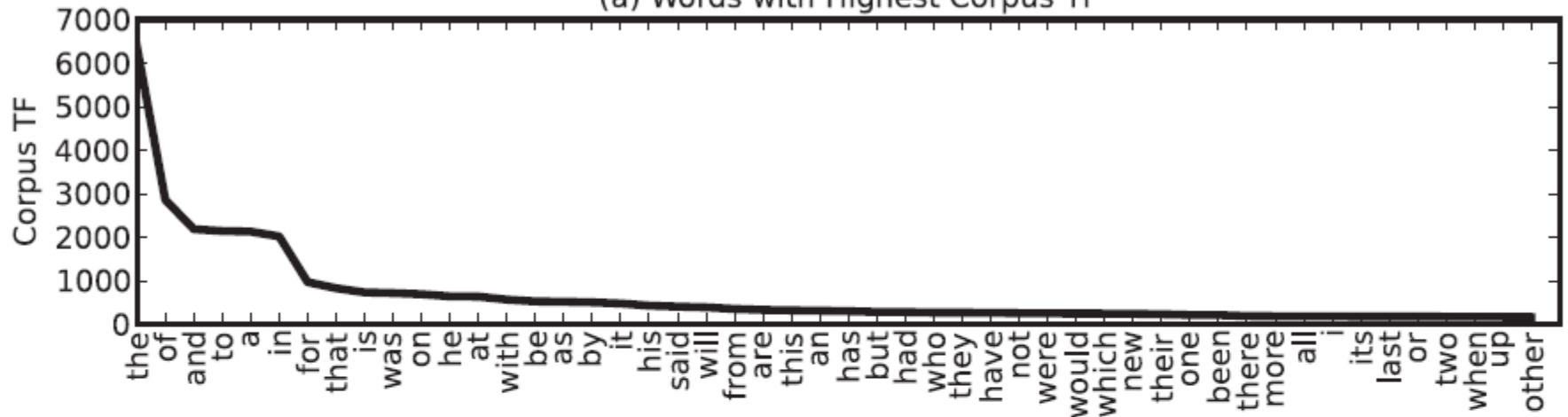
- The **TFIDF** of a term *t* in a document *d* is

$$TFIDF(t, d) = TF(t, d) \times IDF(t)$$

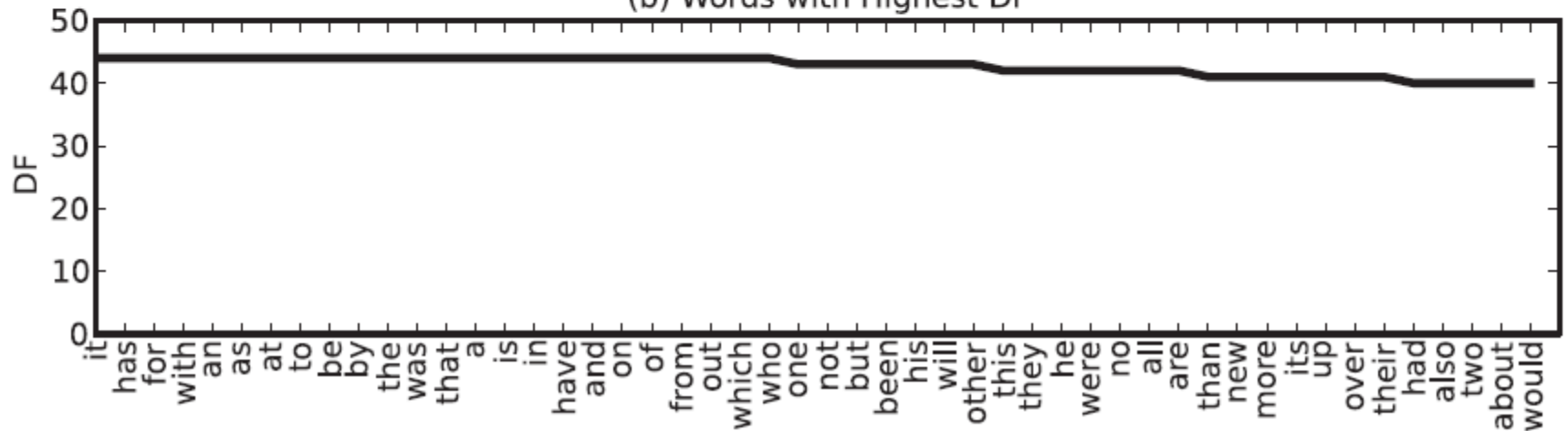
- TFIDF scores a term **higher** if it appears **more often** in a document but **less** in a corpus

Term Frequency – Inverse Document Frequency (TFIDF)

(a) Words with Highest Corpus TF

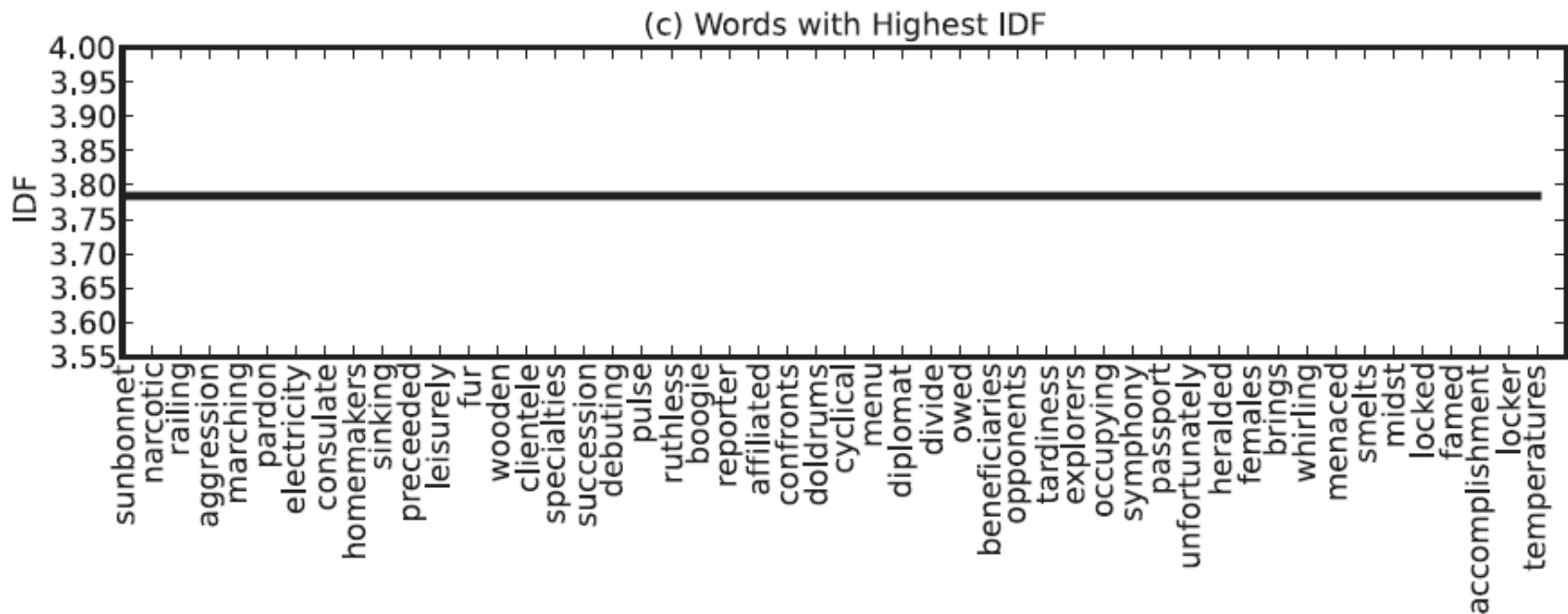


(b) Words with Highest DF



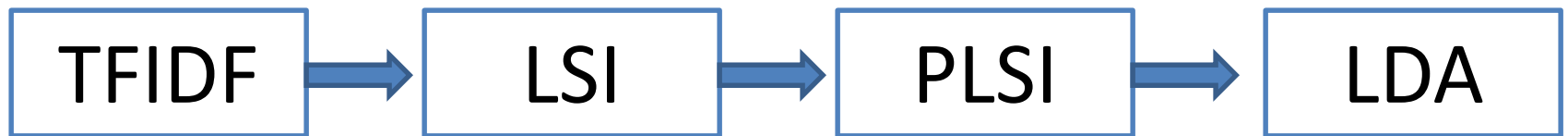
Term Frequency – Inverse Document Frequency (TFIDF)

- TFIDF can be simply and efficiently computed
- A document d is represented as a **high-dimensional vector** of $\text{TFIDF}(t, d)$ value for each term t



Categorizing Documents by Topics

- **TFIDF** approach
 - Provides **relatively small** amount of **reduction** in description length
 - reveals **little** inter-document or intra-document statistical structure



PLSI: probabilistic latent semantic indexing

LDA: latent dirichlet allocation

Categorizing Documents by Topics

- **TFIDF** approach
 - Provides **relatively small** amount of **reduction** in description length
 - reveals **little** inter-document or intra-document statistical structure
- A **topic**: a cluster of words that frequently occur together and share the same theme
 - Each word has a **weight** inside this topic

Categorizing Documents by Topics

- A document typically consists of **multiple** themes running through the text in different proportions

“This paper presents NeuroChess, a program which learns to play chess from the final outcome of games. NeuroChess learns chess board evaluation functions, represented by artificial neural networks. It integrates inductive neural network learning, temporal differencing, and a variant of explanation-based learning. Performance results illustrate some of the strengths and weaknesses of this approach.”

Categorizing Documents by Topics

problem	0.05
technique	0.04
game	0.02
play	0.01
...	

neural	0.06
learning	0.05
networks	0.05
system	0.04
...	

policy	0.02
reinforcement	0.02
state	0.01
model	0.01
...	

report	0.05
technical	0.03
paper	0.02
university	0.02
...	

Categorizing Documents by Topics

- A **topic** is formally defined as a **distribution** over a fixed vocabulary of words
 - Different topics have different distributions over the same vocabulary
- A topic can be viewed as **a cluster of words** with related meanings
 - **A word** from the vocabulary can reside in **multiple topics** with different weights

Categorizing Documents by Topics

- **Topic modelling** provides **short descriptions** for documents
- Helps to organize, search, understand, and summarize text
- **Topic models** are **statistical** models that
 - Examine words from a set of documents
 - Determine the themes over the text
 - Discover how themes are associated

Categorizing Documents by Topics

- The process of **topic modelling** can be simplified to the following
 - **Uncover** the **hidden topical patterns** within a corpus
 - **Annotate** documents according to these topics
 - **Use** annotations to organize, search, and summarize texts

Latent Dirichlet Allocation

- The simplest topic model is **Latent Dirichlet Allocation (LDA)**
 - A **generative** probabilistic model of a corpus
- **Generative** probabilistic model
 - Model observations drawn from a probability density function
- In LDA, **documents** are treated as the **result** of a **generative** process (with hidden variables)

Latent Dirichlet Allocation

- LDA assumes
 - There is a **fixed vocabulary** of words
 - The number of the **latent topics** is predefined
 - Each latent topic follows a **Dirichlet distribution** over the vocabulary
 - Each **document** is represented as a random **mixture of latent topics**

Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.¹

LDA assumes the following generative process for each document \mathbf{w} in a corpus D :

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n :
 - (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
 - (b) Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n .

author = {Blei, David M. and Ng, Andrew Y. and Jordan, Michael I.}, title = {Latent Dirichlet Allocation},
journal = {J. Mach. Learn. Res.}, year = {2003},

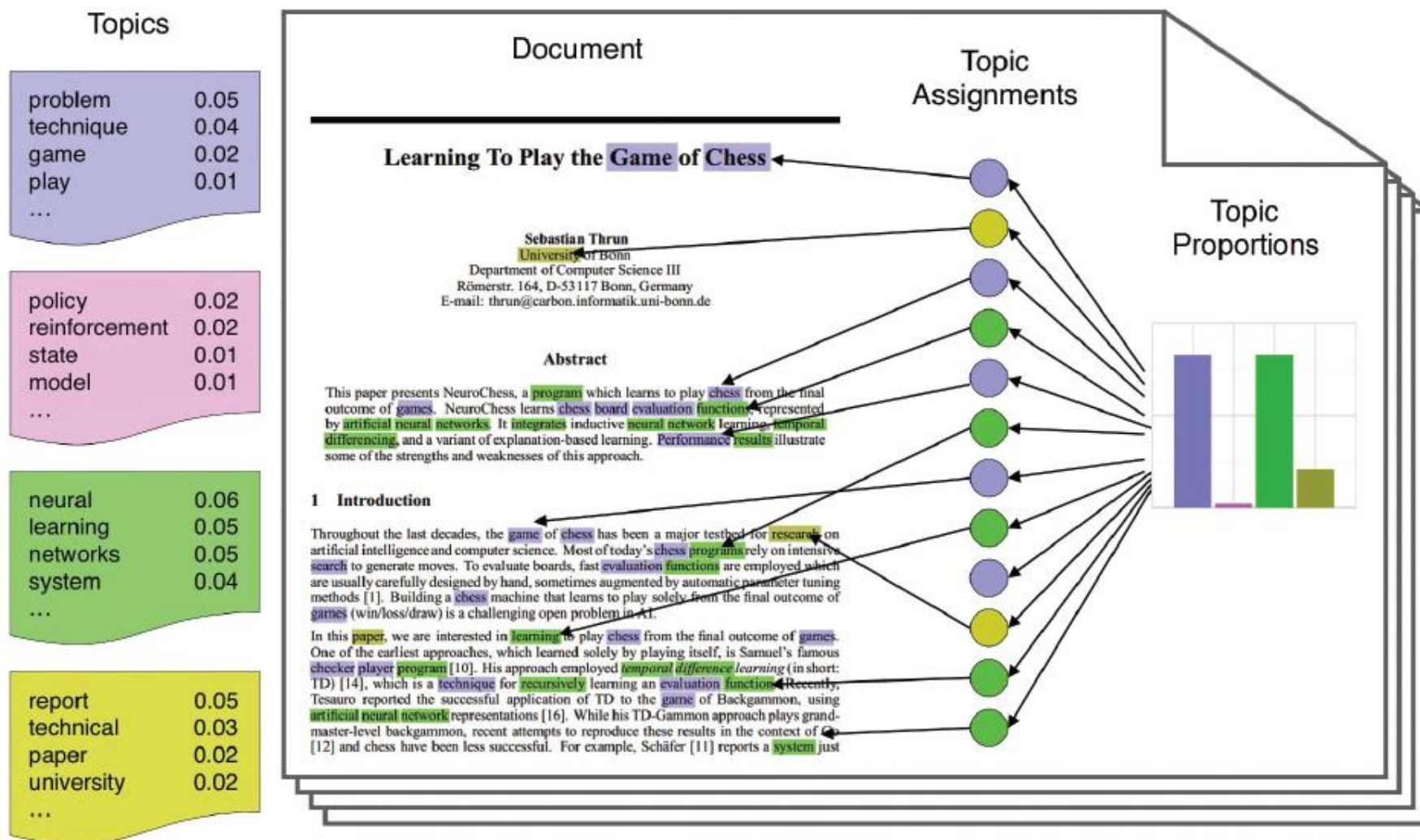
Latent Dirichlet Allocation

- Recipe: how to cook a document via LDA
 - Choose the length N of the document
 - Choose a distribution over the topics
 - For each of the N words of this document
 - Choose a topic based on the above distribution
 - Choose a word from the corresponding topic

- In reality, only the documents are available
- LDA aims to infer the underlying topics, topic proportions, and topic assignment for each document



Latent Dirichlet Allocation



Latent Dirichlet Allocation

- R comes with an [lda package](#) that has built-in functions and [cora](#) datasets (2,410 documents)

```
require("ggplot2")
require("reshape2")
require("lda")

# load documents and vocabulary
data(cora.documents)
data(cora.vocab)

theme_set(theme_bw())

# Number of topic clusters to display
K <- 10

# Number of documents to display
N <- 9
```

Latent Dirichlet Allocation

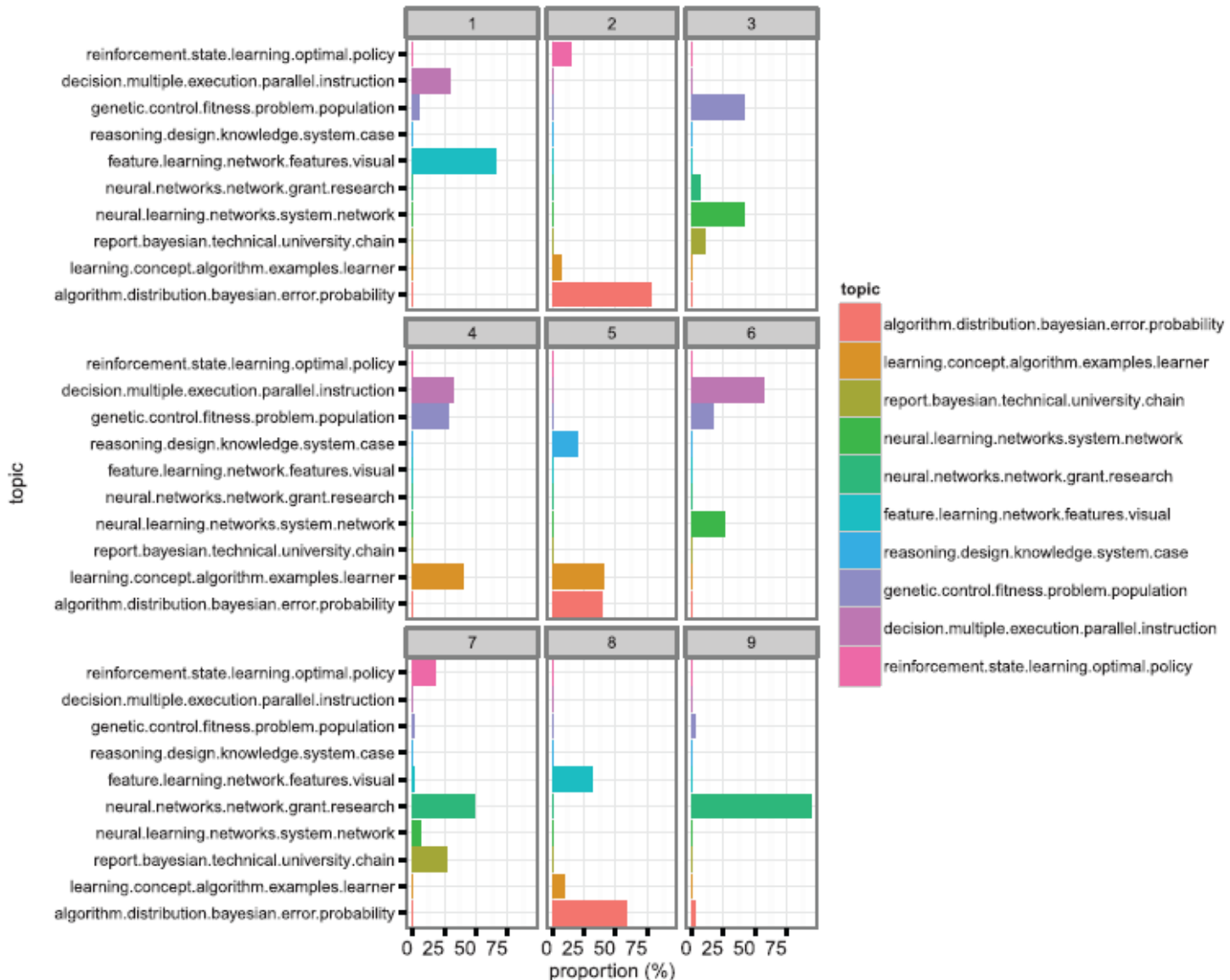
- R comes with an **lda package** that has built-in functions and **cora** datasets (2,410 documents)

```
result <- lda.collapsed.gibbs.sampler(cora.documents,  
                                     K, ## Num clusters  
                                     cora.vocab,  
                                     25, ## Num iterations  
                                     0.1,  
                                     0.1,  
                                     compute.log.likelihood=TRUE)
```

```
# Get the top words in the cluster  
top.words <- top.topic.words(result$topics, 5, by.score=TRUE)
```

Details can be found at <https://cran.r-project.org/web/packages/lda/lda.pdf>

Latent Dirichlet Allocation



Determining Sentiments

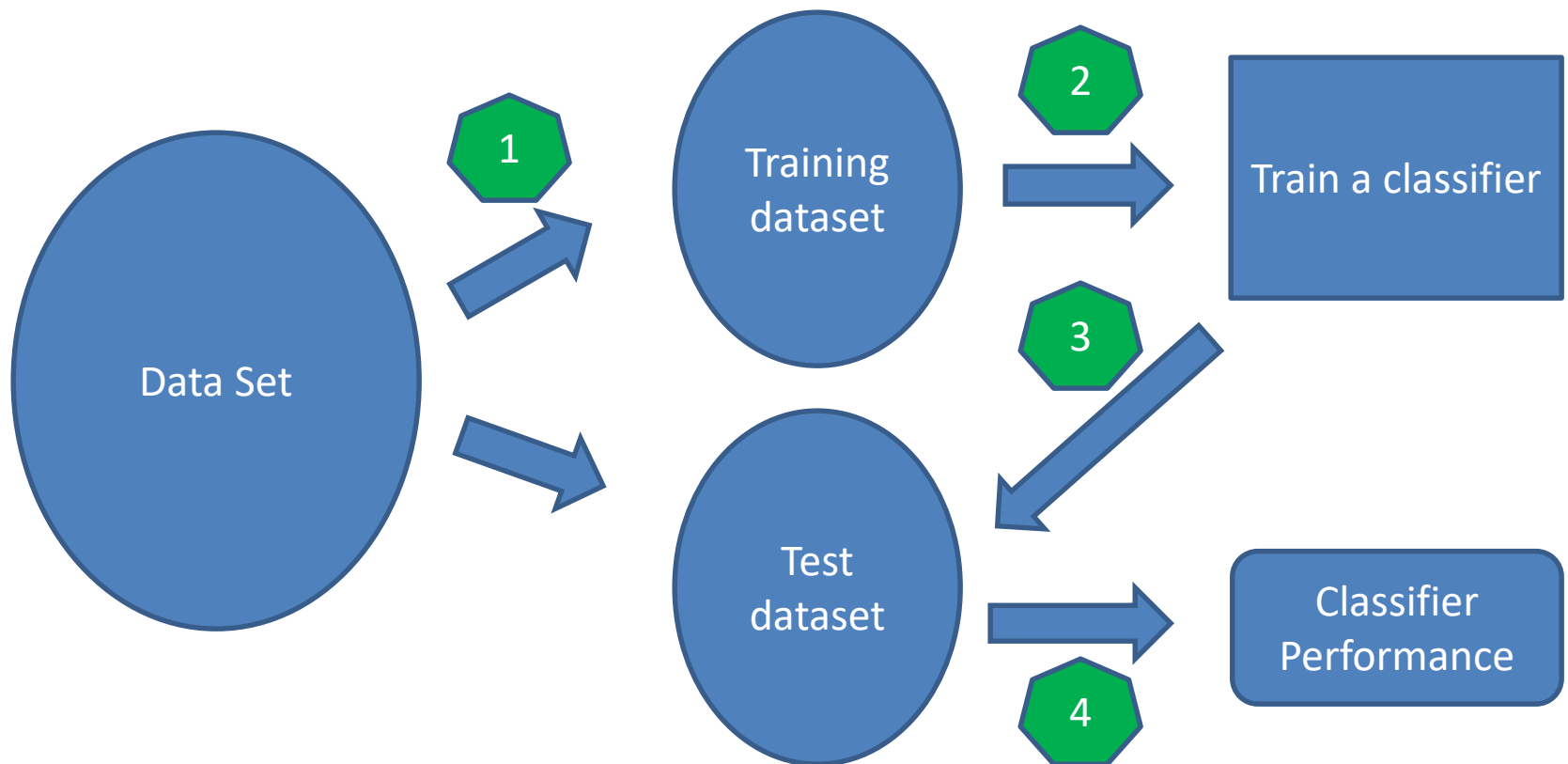
- **Sentiment** analysis
 - Uses statistics and NLP to **mine opinions** to identify and exact **subjective information** from texts
- Applications
 - Detect the polarity of product or movie reviews
- Analysis level
 - Document, sentence, phrase, and short-text

Determining Sentiments

- **Classification** methods are often used to extract corpus statistics for sentiment analysis
 - Naïve Bayes classifier, Maximum Entropy, Support Vector Machines
- **Movie review** corpus
 - Consists of 2000 movie reviews
 - Manually tagged into 1000 positive and 1000 negative reviews

Determining Sentiments




- How to perform classification on a data set for sentiment analysis?



Determining Sentiments

- An example
 - Code written in Python using the Natural Language Processing Toolkit (NLTK) library
 - Split the 2000 reviews into 1600 reviews as training set and 400 reviews as testing set.
 - Naïve Bayes classifier learns from the training set
 - The classifier achieves an accuracy of 73.5%
 - Show most informative features for pos/neg

Determining Sentiments

- Classifiers determine sentiments **solely** based on the datasets on which they are **trained**
 - Word meaning **varies** with the domain
 - Cannot be **directly** apply to **another** domain
- **Absolute** sentiment level is **not informative**
- How to **label** a lager number of reviews?
 - Use emoticons   
 - Use Amazon Mechanical Turk (MTurk)

Gaining Insights

- How data scientists use text analysis techniques to **gain insights** into their tasks?
- **Word cloud** (tag cloud)



Gaining Insights

- TFIDF can be used to highlight the **informative words** in text

★★★★★ **minor bugs** September 17, 2013

this was for my sister who loves it. she says it has minor bugs but nothing she cant deal with. she is overall satisfied with it

★★★★★ **mint condition !!!** September 13, 2013

great price , not a scratch or bump on the bphone ! it came a lot speedier than expected so thats always a plus ! its just wonderful , only had it for a couple of days and could n't ask for anything more ! ! ! !

★ **buttons did not work** September 08, 2013

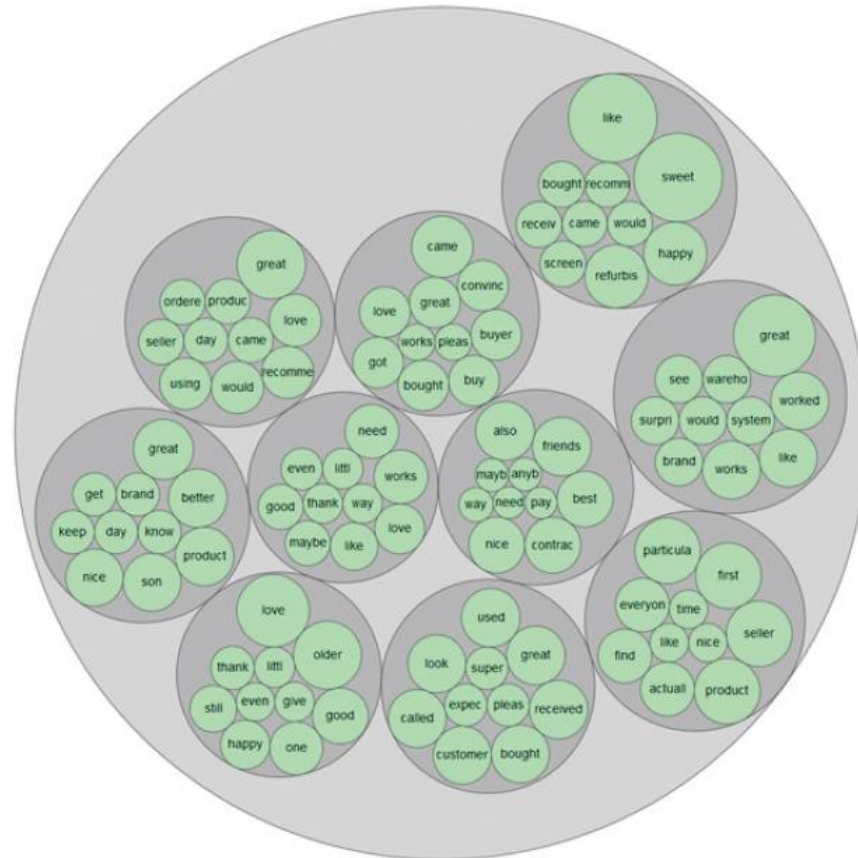
when i went to have my contacts transferred it was found that the two buttons need to switch did not work consistantly

★★★ **it 's a bphone.** August 12, 2013

i hate acme and acme products. base both on principle and on functionality (or lack thereof) . that being said i guess this phone is great for old people that are n't tech savvy. i bought this for my aunt .

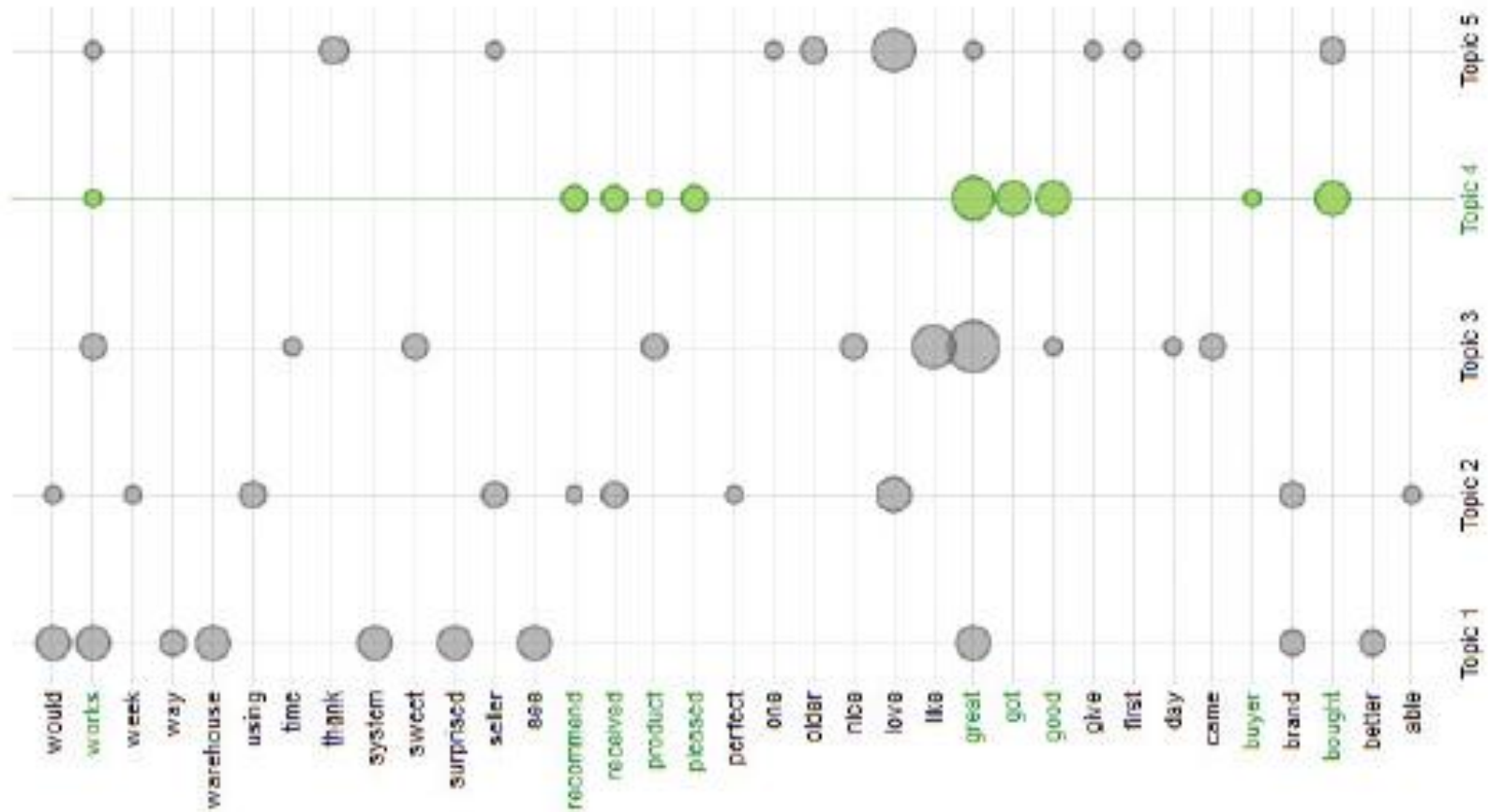
Gaining Insights

- Circular graph of topics obtained from LDA
 - The disc size represents the **weight** of a word



Gaining Insights

- Another way to visualize topics



Summary

- Discusses several subtasks of text analysis
- Talks about a typical text analysis process
 - Collecting raw text
 - Representing text
 - Using TFIDF to describe each word in each doc
 - Topic modelling
 - Sentiment analysis
 - Gaining greater insights

