

Associate Professor
HUANG, Ke-Wei



1. Overview
2. Pre-processing and Jargons (Textbook Chapter 4 for R codes)
3. Method 1: Dictionary Approach
4. Method 2: Text Classification
5. Method 3: Readability analysis

- Using numerical variables or categorical variables for “analytics” has been used in academics for even hundred years.
- But using textual information for analytics is very recent, mostly within 15-20 years and is still evolving because existing academic studies in social sciences mostly use non-Natural Language Processing Methods ⇔ do not fully use the meaning of text.
- We will go over several methods that can turn textual information into numeric information for analytics today.

Definition of Text Mining

- The discovery by computer of new, previously unknown information, by automatically extracting information from different **unstructured** textual documents.
 - Extracting structured textual information, such as categorical variables, does not count.
- Also referred to as text data mining, roughly equivalent to text analytics which refers more specifically to problems based in a business settings.

Why Text Mining is Important?

- Text is everywhere (on the Internet).
- Under-explored treasure of information.
- Many businesses have lengthy document that is impossible for human being to read everything.
- Many applications and sources of text: legal documents, blogs, medical records, customer complaints, customer feedback survey, corporate internal emails, repair records, product inquiries, police written reports, news articles, and more on the next few slides.

Real World Examples and Applications

- News classification
- Spam, Fishing, Scam emails identification
- Google! Search engine and several other products/services
- Summarizing documents
 - Nick D'Aloisio, 17, developed the app, called Summly, while revising for his mock GCSEs in 2011. Telegraph UK, 2013
- Spelling correction

Real World Examples and Applications

- FB Twitter Reddit (all social media and online forums) brand mention monitoring and opinion mining (sentiment mining)
- Online consumer reviews summarizing and sentiment analysis
- Stocks investment forum sentiment analysis
- Annual report analysis in accounting academic studies
- IBM Watson Jeopardy (understanding language + knowledge graph)
- Amazon recommendation system based on textual contents
- Many others...

Why Text Mining is Difficult?

- Text is often referred to as “unstructured” data.
 - Unstructured data for data mining is relatively new.
 - Other unstructured data includes: image, video, and voice (tone, not the meaning of speech). All are still at the nascent stage for data analytics.
- “A meaningful phrase” may consist of different number of words in both English and Chinese.
- Your textual data may have typos
- Domain matters
- Abbreviations causes synonyms
- An example, what is the sentiment?

*“The first part of this movie is far better than the second. The acting is poor and it gets out-of-control by the end, with the violence overdone and an **incredible** ending, but it’s still fun to watch.”*

← Positive or negative?

Methods Covered in these 2 Weeks

1. Dictionary Approach

- Relatively easy to code by R; accuracy not high except in few applications; can be applied to short sentences and very small number of documents.

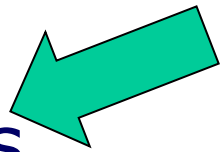
2. Text Classification

- More complicated than dictionary approach. Accuracy is better than dictionary approach and can be quite good in some cases. Weakness is supervised learning.

3. Readability: only about the quality of writing **(skipped if not enough time)**

4. Topic Modelling (LDA): Conceptually more complicated than text classification. Accuracy may be similar or worse. Unsupervised learning.

5. (very brief, basic) Natural Language Processing for sentiment analysis: (at least one semester of class for NLP)

1. Overview
2. Pre-processing and Jargons 
3. Method 1: Dictionary Approach
4. Method 2: Text Classification
5. Method 3: Readability analysis

Terminologies

- A **document** is one piece of text, no matter how large or small.
 - A document could be a single sentence or a 100-page report, or anything in between, such as a You-Tube comment or a blog posting.
 - For example, research on annual reports in accounting. The same method can be applied to different units of analysis (different definitions of documents) => the whole annual report of one firm in one year, each section, each paragraph, or each sentence.
- A collection of documents is called a **corpus**.
- A document is composed of individual **tokens** or **terms**. Terms can be one English word or can be more than one English words.

Bag of Words

- This approach is to treat every document as just a collection of individual words and also each word can appear multiple times.
- In other words, this approach ignores grammar, word order, sentence structure, and (usually) punctuation => we only consider the occurrence of words.
- The representation is straightforward and computationally inexpensive to generate, and tends to work well for many tasks.
- **Most text mining methods are bag of words approach.**

Bag of Words Representation

The Bag of Words Representation

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



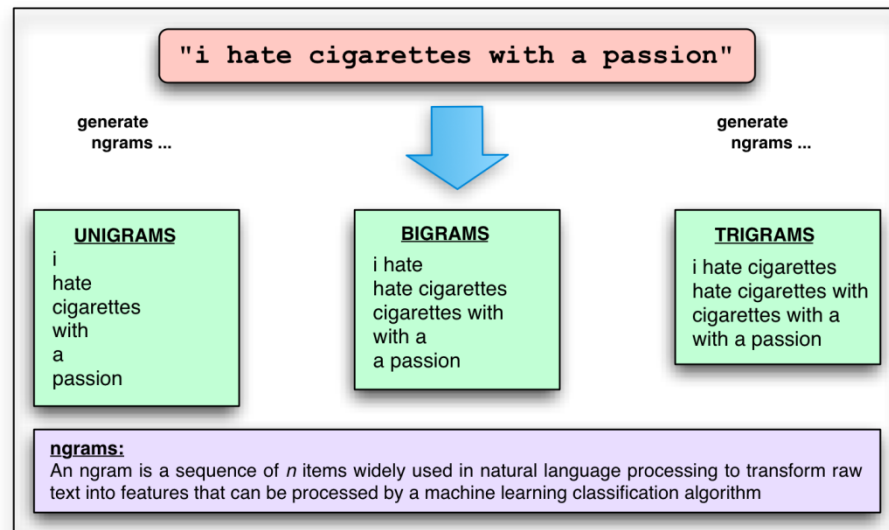
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Pre-Processing

1. **Normalization**. Every term is in lowercase. Almost all methods are case-insensitive. All special characters or even numbers, punctuations are removed.
 - BUT: 4TB and 1Q13 may be removed too.
2. **Stemming**:
 - Suffixes removed, so that verbs like *announces*, *announced* and *announcing* are all reduced to the term **announc** <= not a typo, it is really like this.
 - Stemming transforms noun plurals to the singular forms
3. **Stopwords** have been removed.
 - A stopword is a very common word in English. For example: *the*, *and*, *of*, and *on* are considered stopwords in English so they are typically removed.
 - BUT: movies "*The Road*" and "*On The Road*" are different.

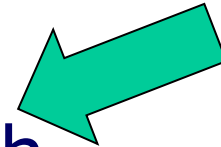
4. Tokenization is not discussed in the textbook but is important in practice and if considered,

- It will be implemented after normalization (case insensitive) and before stemming.
- You need to decide **n-gram tokenization** ⇔ how many words in one term will you consider. For example: "mutual fund" is very different from "mutual" & "fund". "annual report" loses its meaning when you consider it separately.



Algorithms will decide keeping only meaningful ngrams, not using all of it.

1. Overview
2. Pre-processing and Jargons
3. Method 1: Dictionary Approach
4. Method 2: Text Classification
5. Method 3: Readability analysis



Dictionary Approach

- The idea is straightforward. Given a dictionary of “Positive words” (=P) and a dictionary of “Negative words” (=N), you can count the number of positive or negative words of an input text.
- Then you can use a formula to represent the sentiment of that text. For example, $P/(P+N)$. (If $P+N=0$, then 0). OR, $P/\text{Length}(\text{document})$.
- Harvard’s General Inquirer has many dictionaries for you to use
<http://www.wjh.harvard.edu/~inquirer/homecat.htm>
- There are many other dictionaries.
- Dictionaries are important for this approach and you may need to customize for your own purpose.

Dictionary Approach

- Although this approach does not consider grammar (especially negate) at all, it could provide useful information on average if the label maps to a set of keywords (your dictionary).
 - This method does not fit all applications.
- Because of its simplicity, widely used in academic papers and in practice.
- **Also, you can use other formulas to represent sentiment or other conceptual variables.**
- Bear in mind that context matters, for example, “liability” is not that negative in an annual report. It is a neutral noun in annual reports. In plain English, it is indeed negative.

Example

- Tetlock, Paul C., Maytal Saar-Tsechansky, and Sofus Macskassy. "More than words: Quantifying language to measure firms' fundamentals." *The Journal of Finance* 63.3 (**2008**): 1437-1467.

Three main findings are:

- (1) The fraction of negative words in firm-specific news stories forecasts low firm earnings;
- (2) Firms' stock prices briefly underreact to the information embedded in negative words;
- (3) The earnings and return predictability from negative words is largest for the stories that focus on fundamentals.

Data:

- The fraction of negative words in *DJNS* and *WSJ* stories about **S&P 500 firms** from 1980 through 2004. In total, 260K from *DJNS* and over 90K from *WSJ*.
1. News articles are collected from Factiva database.
 2. Stock prices are available at CRSP dataset.
 3. Analysts forecasted earnings available at I/B/E/S dataset.
 4. Accounting information available from Compustat.
 5. All are “publicly available” to universities.

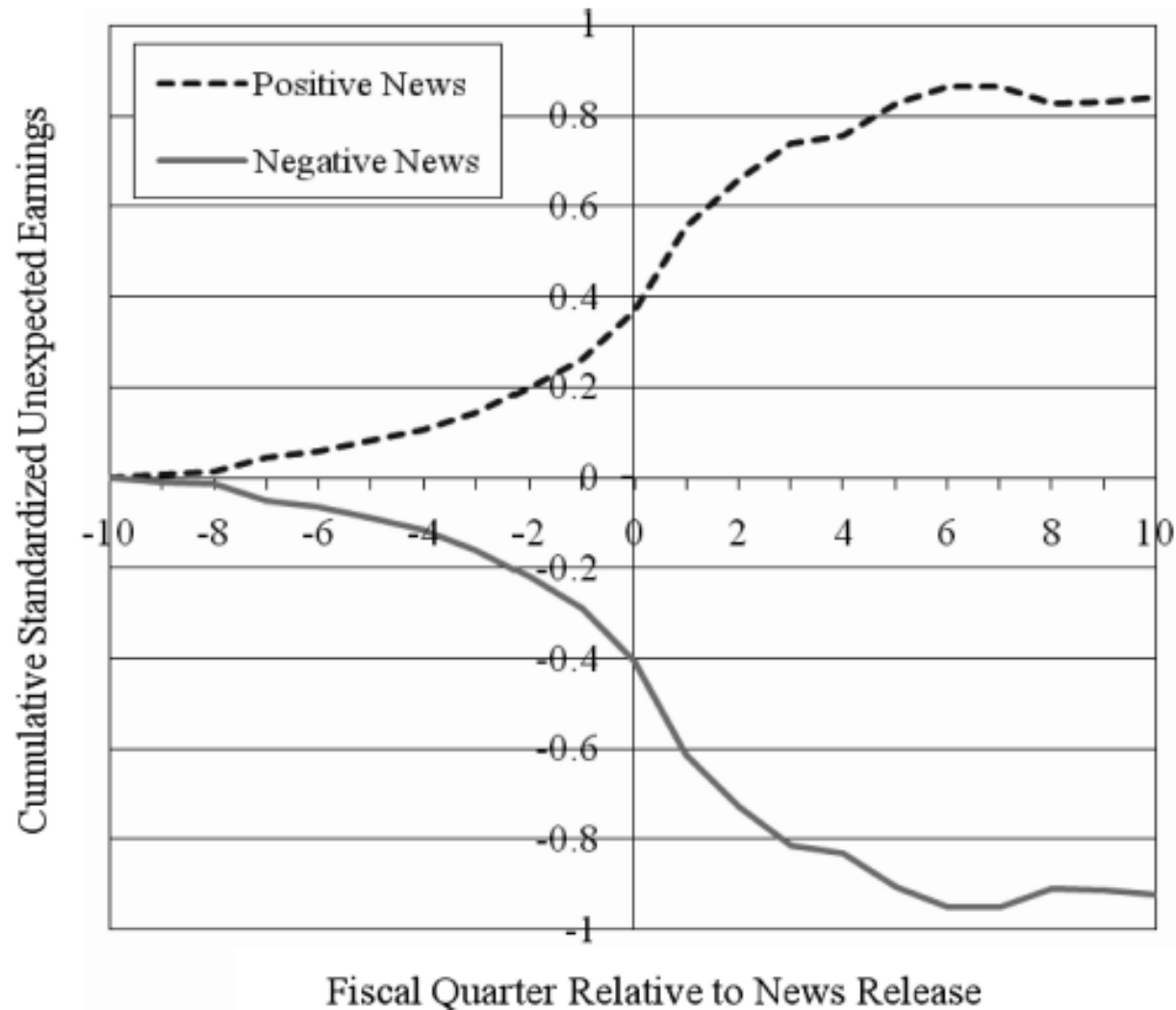
Sentiment Formula in this Paper

$$Neg = \frac{\text{No. of negative words}}{\text{No. of total words}} \quad (1)$$

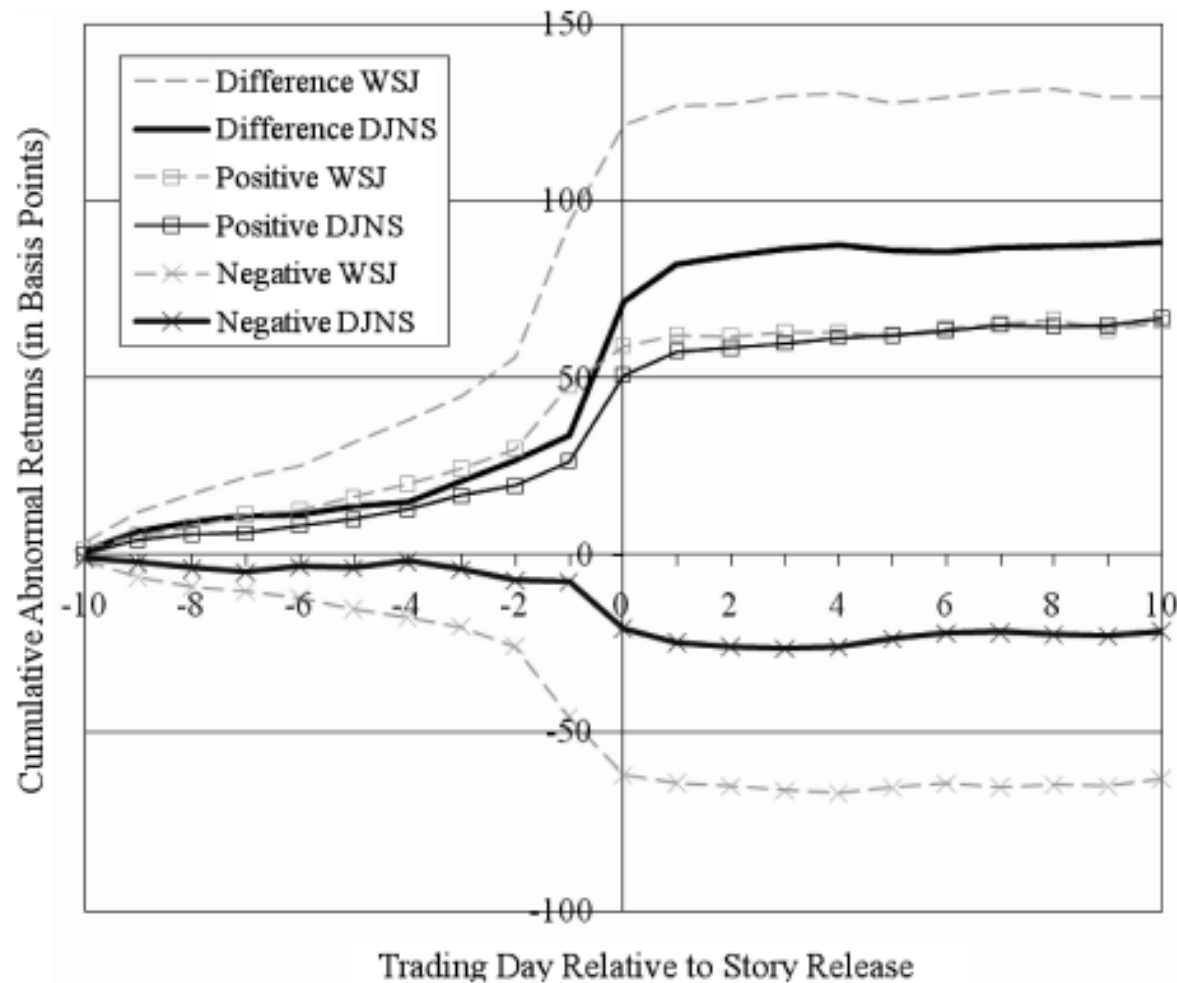
$$neg = \frac{Neg - \mu_{Neg}}{\sigma_{Neg}}, \quad (2)$$

- The authors standardize the fraction of negative words in each composite news story by subtracting the prior year's mean and dividing by the prior year's standard deviation of the fraction of negative words.
- The authors claim that they find very similar results using combined measures of positive (P) and negative (N) words, such as $(P - N)/(P + N)$ and $\log((1 + P)/(1 + N))$.

Earnings and News Sentiment



Stock Abnormal Return & News Sentiment




Pros and Cons of Dictionary Approach

Pros

1. Simple! Everyone can understand and interpret results.
2. Fast.
3. No training.
4. Works even if you have only one observation.

Cons.

1. May not be that accurate for some applications.
2. You may not have a dictionary that fits your goal.
3. Do not capture grammars and especially “negate”.
4. Not sure which formula is the best.

1. Overview
2. Pre-processing and Jargons
3. Method 1: Dictionary Approach
4. Method 2: Text Classification and TF-IDF 
5. Method 3: Readability analysis

3. Text Classification

- The first step is to convert a textual document into a list of numerical variables (numerical features) and that numerical features capture the key properties of the document.
- The most intuitive way to achieve this goal is we have a long list of “**word vector**”. For example, say all English words and there are 10000 English words.
- Each document can be represented by a vector with 10000 elements and the value of each element is the **term-frequency** of each word.
 - TF=>The number of occurrence of that word as a % of a document (divided by the total number of words of that document).

Text Classification

- For 10,000 words, let's say you already exclude stop words and apply stemming.
- Given this numeric vector with 10000 elements, we consider each term frequency as a feature for traditional supervised learning.
- You can add some more features from non-textual information.
- Next, we can apply any traditional data mining classification algorithm to predict the category/label of a document as a starting point.
- CS researchers have found ways to improve performance based on this rough preliminary idea described above.

Term-Frequency (TF)

- Term Frequency means using the word count (frequency) in the document instead of just a zero or one.
- The underlying assumption is the importance of a term in a document should increase with the number of times that term occurs.

Table 10-1. Three simple documents.

d1 jazz music has a swing rhythm

d2 swing is hard to explain

d3 swing rhythm is a natural rhythm

Table 10-2. Term count representation.

	a	explain	hard	has	is	jazz	music	natural	rhythm	swing	to
d1	1	0	0	1	0	1	1	0	1	1	0
d2	0	1	1	0	1	0	0	0	0	1	1
d3	1	0	0	0	1	0	0	1	2	1	0

Term-Frequency (TF)

There are several formulas for TF

1. The most intuitive one is exactly the frequency of that word. This is also used in R's tm package.
2. The most commonly used is the scaled logarithm
TF: $1 + \log_{10}(\text{frequency})$, 0 if frequency is 0
3. 0-1 is acceptable and used in some special cases. (1 means that word appears, 0 else)
4. There are also other special-purpose TF measures, such as Augmented Frequency

Inverse-Document-Frequency (IDF)

- TF is not good enough \Leftrightarrow We may also care, when deciding the weight of a term, how common it is in the entire corpus we're mining.
- A rare word (with low TF) could be very informative and it is quite common for sentiment analysis or text classification in which when a rare word occurs, it is a strong signal of the class label.
- A word may have high TF in ALL documents and thus is still not that important even when we observe high TF in one document.

Equation 10-1. Inverse Document Frequency (IDF) of a term

$$\text{IDF}(t) = 1 + \log \left(\frac{\text{Total number of documents}}{\text{Number of documents containing } t} \right)$$

From the 2nd textbook, 

Inverse-Document-Frequency (IDF)

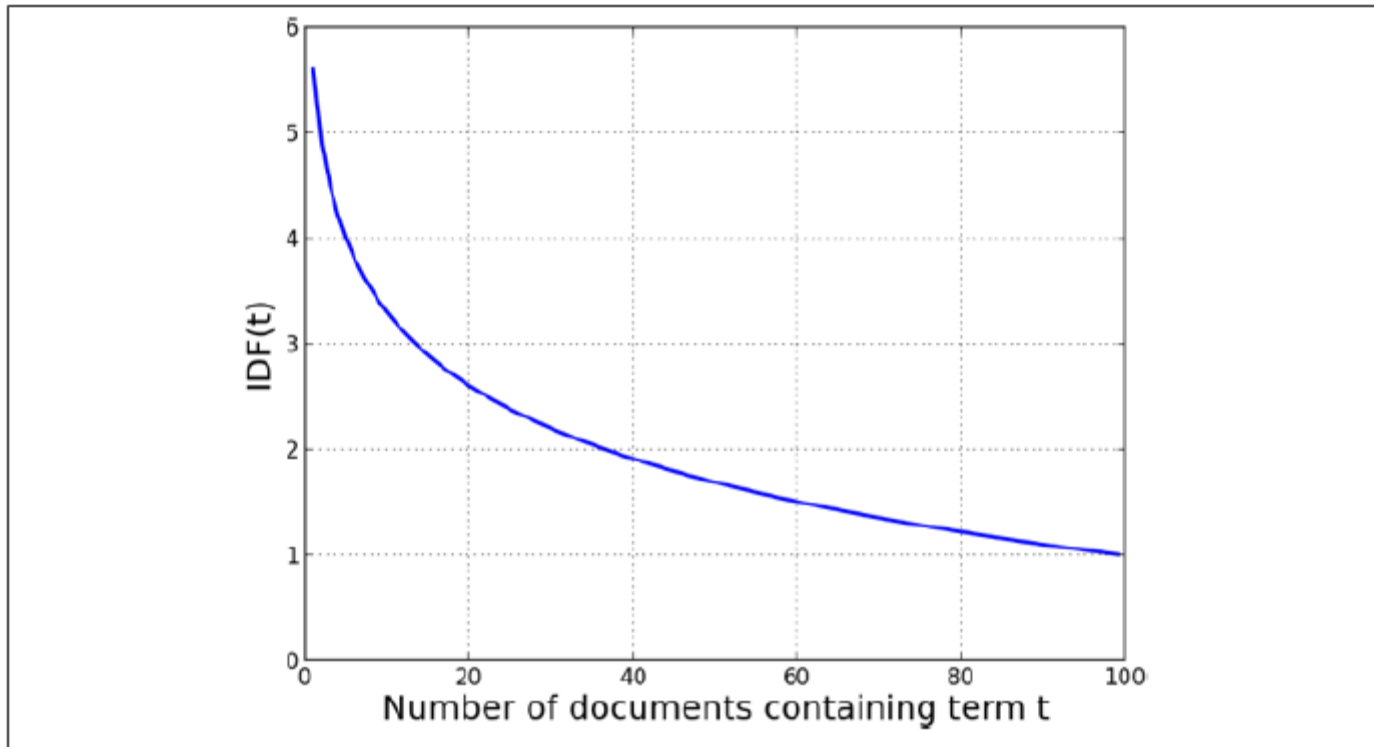


Figure 10-1. IDF of a term t within a corpus of 100 documents.

When all documents contain that keyword, IDF is 1. When only 1 document contains that keyword, IDF is large ($1 + \log N$)

Inverse-Document-Frequency (IDF)

Variants of IDF weight

weighting scheme	IDF weight ($n_t = \{d \in D : t \in d\} $)
unary	1
inverse document frequency	$\log \frac{N}{n_t}$ → Implemented in R's tm package
inverse document frequency smooth	$\log \left(\frac{N}{1 + n_t} \right)$
inverse document frequency max	$\log \left(\frac{\max_{\{t' \in d\}} n_{t'}}{1 + n_t} \right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

TF-IDF as the Values in Classification Algorithms

- The “norm” of text mining is to use TF-IDF as the weight for each term in the feature vector that represents a document.
- TF-IDF is given by $\text{TFIDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$
- One common version of TF-IDF is

$$w_{t,d} = (1 + \log tf_{t,d}) \times \log_{10} \left(\frac{N}{df_t} \right)$$

Total number of documents

Number of documents that contain that keyword

- The TF-IDF implemented in R's tm package can be found in <https://www.rdocumentation.org/packages/tm/versions/0.7-3/topics/weightTfIdf>

Illustrative Example of Output Vector

← Keywords=attributes

id	men	entered	bank	charlotte	missiles	masks	aryan	guns	witnesses	reported	silver	suv	august
seg1.txt	0.239441	0	0.153457	0.195243	0	0.237029	0	0.195243	0.237029	0.140004	0.195243	0.237029	0
seg13.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg14.txt	0	0.192197	0	0	0	0	0	0	0	0	0	0	0.172681
seg15.txt	0	0	0	0	0	0	0	0	0	0	0	0	0.149652
seg16.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg17.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg18.txt	0	0.158432	0	0	0	0	0	0	0	0	0	0	0
seg19.txt	0	0	0	0.197255	0	0	0	0	0	0.141447	0	0	0.155038
seg2.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg20.txt	0	0.234323	0	0	0	0	0	0	0	0	0	0	0
seg21.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg22.txt	0	0	0	0	0.139629	0	0.127389	0	0	0	0	0	0
seg23.txt	0	0	0	0	0	0	0	0	0	0.180656	0	0	0
seg24.txt	0	0	0	0	0	0	0.117966	0	0	0.117966	0	0	0
seg25.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg26.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg27.txt	0	0	0.235418	0	0	0	0.214781	0	0	0	0	0	0
seg28.txt	0	0	0	0	0.151753	0	0	0	0	0	0	0	0
seg29.txt	0	0	0	0	0	0	0.129852	0	0	0	0	0	0.142329
seg3.txt	0	0	0	0	0.18432	0	0	0	0	0	0	0	0
seg30.txt	0.078262	0	0	0	0	0	0	0	0	0	0	0	0
seg31.txt	0	0	0.213409	0	0	0	0.194701	0	0	0	0	0	0
seg32.txt	0	0	0	0	0	0	0	0	0	0	0	0	0

← Your documents

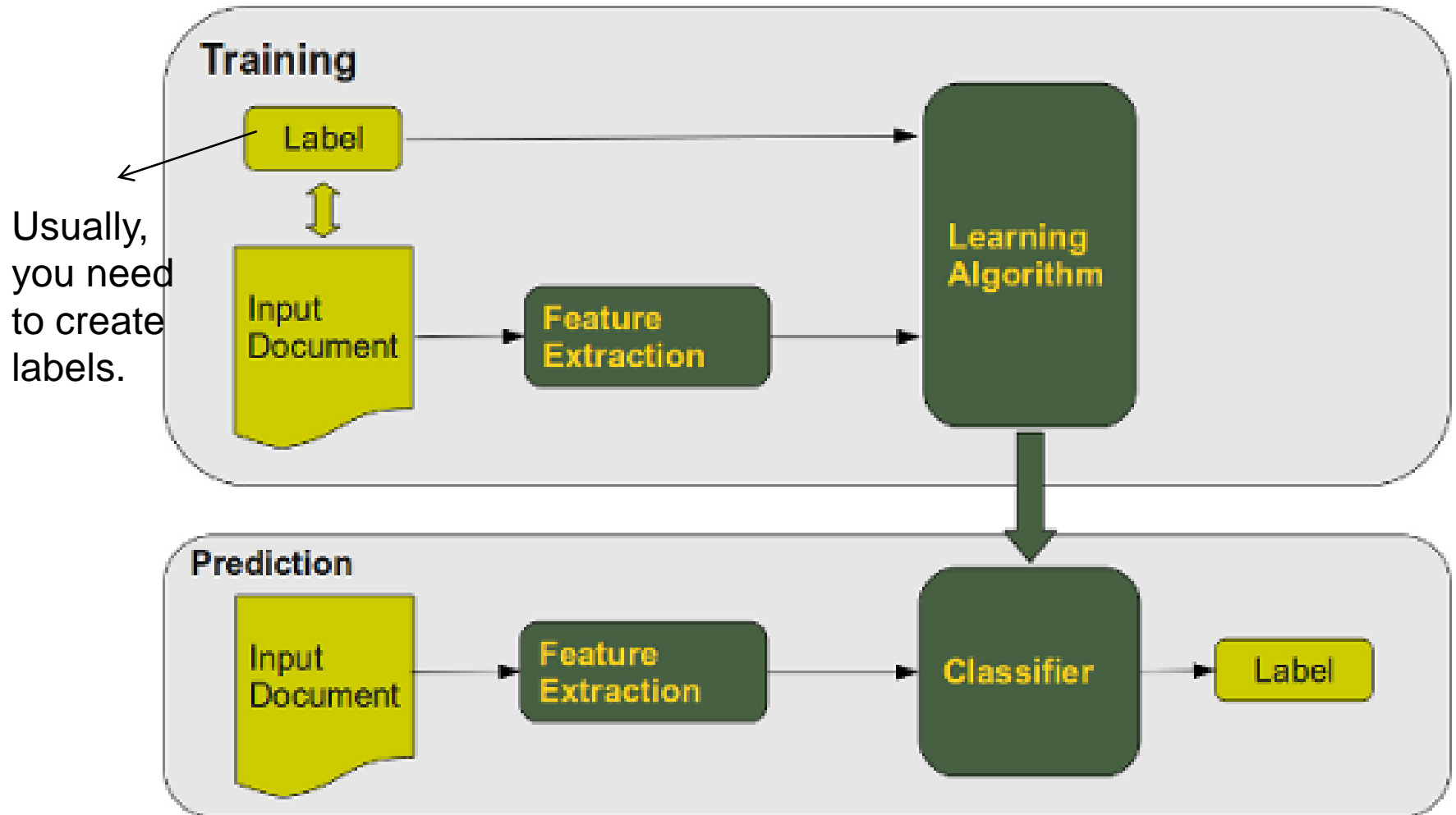
↑ TF-IDF as values

↑ Sparse Matrix (many 0s)
is common

Text Classification Procedure

- Step 1: apply 4 pre-processing steps to decide the keywords as the attributes used for classification. Those keywords are the elements of the word vector.
- Step 2: choose one formula of TF-IDF to decide the values of word vector (just let the software do it).
 - If the word vector is not too long, you should check **manually** and customize it. Eliminate useless words and increase the weights of keywords if needed.
- Step 3: Now all documents are represented by numeric vectors, you can apply any existing methods (decision tree, neural network, SVM, XGBoost) for predicting values or predicting label classes.
 - You may need to manually label documents if label are not available in the data.
- Step 4: Select the best algorithm and parameters tuning by 10-fold cross-validation as usual.

Text Classification and Prediction

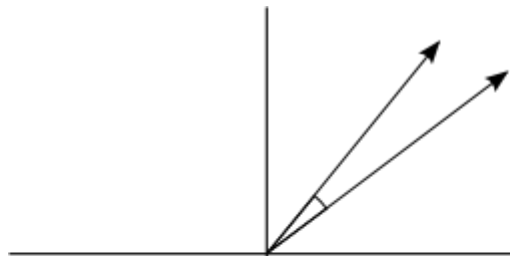
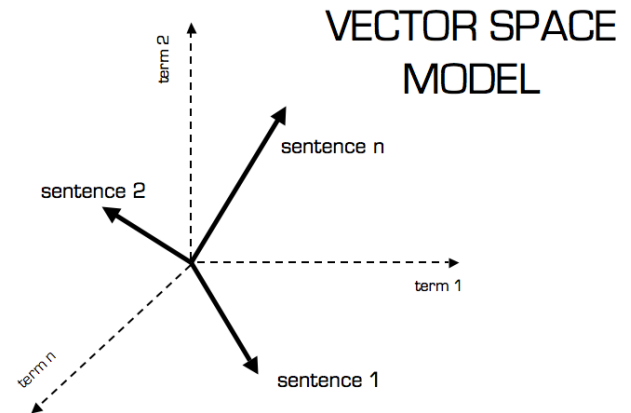
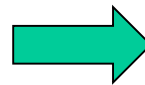


Vector Space Model and Cosine Similarity

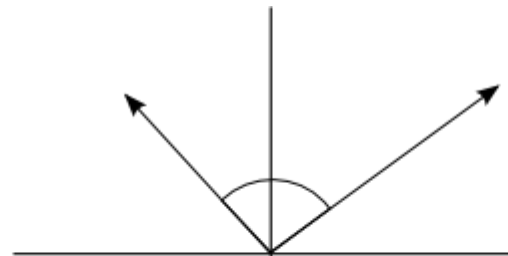
- However, results may not be good if we simply apply text classification on the attributes based on the TF-IDF of keywords in some applications.
- Some methods (e.g., KNN) rely on the similarity score between two documents for classification or clustering and we need a good metric to measure the similarity.
- Euclidean distance is the most intuitive choice.
- Research shows that **"Cosine Similarity" generally works well**, better than Euclidean distance.
- This is because when keywords proportionally increases, (E.g. you repeat a similar sentence twice.) Euclidean distance is large but cosine similarity is high in these cases.
- There are many other distance metrics that are available in software packages for text mining.

Cosine Similarity

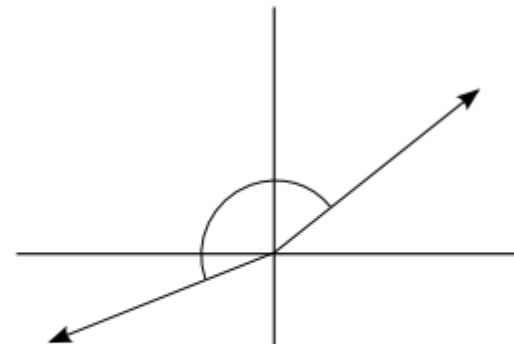
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$



Similar scores
Score Vectors in same direction
Angle between them is near 0 deg.
Cosine of angle is near 1 i.e. 100%

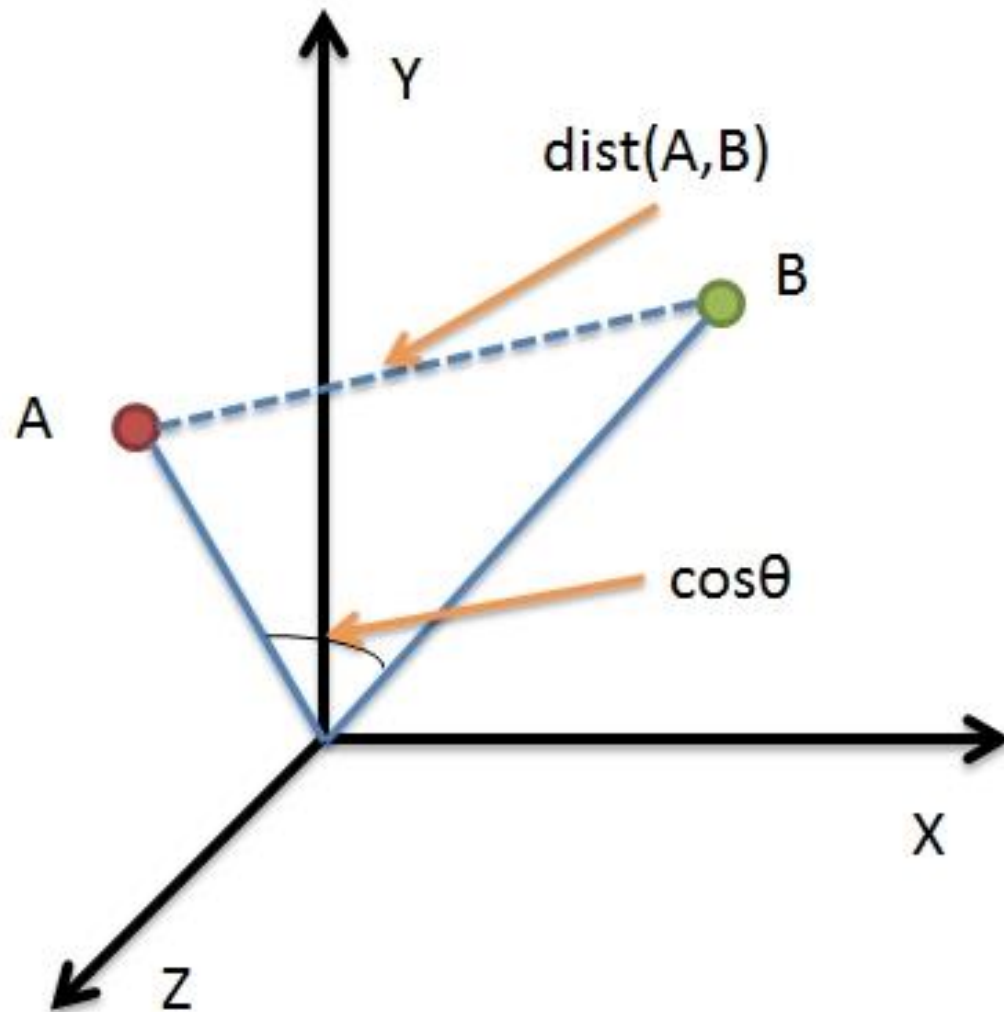


Unrelated scores
Score Vectors are nearly orthogonal
Angle between them is near 90 deg.
Cosine of angle is near 0 i.e. 0%



Opposite scores
Score Vectors in opposite direction
Angle between them is near 180 deg.
Cosine of angle is near -1 i.e. -100%

Cosine Similarity and Euclidean Distance



Examples of Text Classification Applications

- There are surge of interests in applying text classification techniques in accounting and finance research after 2005. See footnotes for the references.
- 1. Analyzing annual report of public firms and accounting disclosures.
- 2. Financial Forecasting for stock prices or Forex
 - 1. Easier to predict volume and volatility.
 - 2. Most CS publications over-claim the predictability of stock or Forex returns which are not credible to finance professors ⇔ simple methods wont work while advanced methods may work but challenging.
- 3. CRM applications
- 4. Financial Fraud Detection

Example: Dr. Huang's paper

Background and Motivation

- SEC regulation change in 2005: "Risk Factors" in the annual reports (SEC Form 10K).
- Risk factors are reported as a list of bullet in Section 1a "Risk Factors"
 - *Activision: "we rely on independent third parties to develop some of our software products."*
 - *SalesForce.com: "Because we recognize revenue from subscriptions..., downturns or upturns in sales may not be immediately reflected in our operating results."*
 - *Adobe: "We may incur substantial costs enforcing or acquiring intellectual property rights and defending against third-party claims"*

Research Design

- Stage 1: Applying text-classification algorithms to classify risk factors reported in the annual reports of all software companies. (NAICS 511210)
 - Training set: R.A. manually labeled the 10K of all software companies in 2008 into 20 types.
 - Training set performance => around 80-85%
- Stage 2: Panel-firm-level regression analysis
 - Dependent variable: various firm performance measures.
 - Independent variables: dummy variable that indicates Type X risk factor reported Year Y

Pros and Cons of Text Classification

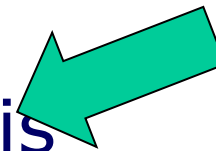
Pros

1. **Could be** more accurate than dictionary.
2. You can manually create any label! The applicability is much wider than the dictionary approach.

Cons.

1. Time consuming to read and code documents.
2. You may need two people to read the same document.
3. This is still “bag of words” approach that do not consider grammar and may not work well if label does not correlate with a set of word occurrences.

1. Overview
2. Pre-processing and Jargons
3. Method 1: Dictionary Approach
4. Method 2: Text Classification
5. Method 3: Readability analysis



5. Readability

- Wiki: *Readability is the ease with which a written text can be understood by a reader.*
- The readability of a particular text depends both on its content (for example, the complexity of its vocabulary and syntax) and on its typography (for example, its font size, line height, and line length).
- In practice, readability measures are proxies of (1) complexity of a text input, (2) quality of a text input.

The popular readability formula

- **1. Gunning fog index**

= $0.4 * [(\text{average sentence length}) + (\text{percentage of Hard Words})]$

- **2. The Flesch formulas**

Reading Ease score

= $206.835 - (1.015 \times \text{ASL}) - (84.6 \times \text{ASW})$

where:

ASL = average sentence length (number of words divided by number of sentences)

ASW = average word length in syllables (number of syllables divided by number of words)

- **3. LIX:**
$$\text{LIX} = \frac{A}{B} + \frac{C \cdot 100}{A}$$

where A is the number of words, B is the number of periods (defined by period, colon or capital first letter), and C is the number of long words (more than 6 letters)

Applications

- Quite a number of software can calculate this list of variables for you and the benefit of this method is: it is easy to obtain those metrics.
1. It has been shown in the literature that the worse readability of annual report is associated with poorer earnings prospect.
 2. Readability of marketing messages in advertisement or promotion email may affect the response rate.
 3. Readability metrics could be control variables or additional attributes for any of your text mining task for exploration.

Accounting Example

Li, Feng. "Annual report readability, current earnings, and earnings persistence." *Journal of Accounting and economics* 45.2 (**2008**): 221-247.

- This paper examines the relation between annual report readability and firm performance and earnings persistence.
- The readability of public company annual reports is measured by Fog index and the length of the document.
- Main Findings: (1) the annual reports of firms with lower earnings are harder to read (i.e., they have a higher Fog index and are longer); and (2) firms with annual reports that are easier to read have more persistent positive earnings.