A BRIEF SURVEY OF TEXT MINING: CLASSIFICATION, CLUSTERING AND EXTRACTION TECHNIQUES

RESEARCH REVIEW BY WILLIAM STEIMEL

SOURCES

- A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques
  - https://arxiv.org/pdf/1707.02919.pdf
  - KDD Bigdas, August 2017, Halifax Canada
    - Mehdi Allahyari (University of Georgia- Computer Science Department)
    - Seyedamin Pouriyeh (University of Georgia- Computer Science Department)
    - Mehdi Assefi (University of Georgia- Computer Science Department)
    - Saied Safaei (University of Georgia- Computer Science Department)
    - Elizabeth D. Trippe (University of Georgia- Institute of Bioinformatics)
    - Juan B. Gutierrez (University of Georgia- Department of Mathematics)
    - Krys Kochut (University of Georgia- Computer Science Department)
- Feature Engineering for Machine Learning
  - Alice Zheng, Amanda Casari
    - Chapter 3: Text Data: Flattening, Filtering, and Chunking
    - Chapter 4: The Effects of Feature Scaling: From Bag-of-Words to Tf-Idf

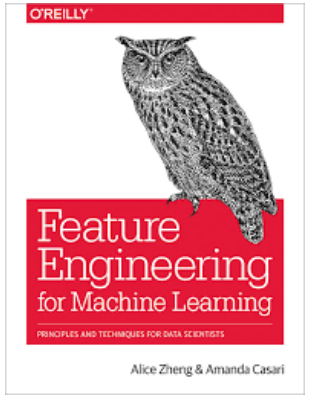


TABLE OF CONTENTS

- A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques
  - Abstract
  - 1. Introduction
  - 2. Text Representation and Encoding
- Feature Engineering for Machine Learning - Text Pre-Processing
  - Chapter 3: Text Data: Flattening, Filtering, and Chunking
  - Chapter 4: The Effects of Feature Scaling: From Bag-of-Words to Tf-Idf

ABSTRACT

- The amount of unstructured text data is increasing every day
  - Text mining is the task of extracting meaningful information from this unstructured text data.
- This paper discusses the basic text mining tasks:
  - Text pre-processing
  - ~~Classification~~
  - ~~Clustering~~
- This paper also includes topics related to biomedical and health care text mining which I have omitted as it is not related to my research.

1. INTRODUCTION

- Text mining as a field has gained a lot of popularity as of recently:
  - Text data is considered unstructured data
    - Social Network Data
    - Patient Records
    - Health Care Insurance Data
    - News Outlets (NHK, CNN, BBC etc.)
  - A report published by EMC expects text data to grow to 40 zetabytes by 2020:
    - This makes the development of text mining approaches and methods essential to effectively process this huge amount of data.
- Text mining approaches are related to traditional data mining and knowledge discovery methods

1.1 KNOWLEDGE DISCOVERY VS DATA MINING

- How are text mining approaches related to Knowledge Discovery and Data mining?
  - Overall Process vs Step in the process
- **Knowledge Discovery** – "is extracting implicit valid, new and potentially useful information from data, which is nontrivial " KDD PROCESS
    - 1. Understanding application of data and identifying the goal
    - 2. Data preparation/pre-processing
    - 3. Modeling
    - 4. Evaluation
    - 5. Deployment
- **Data Mining** – "is a the application of particular algorithms for extracting patterns from data."
    - Data Mining is considered a step or the modeling phase in the Knowledge Discovery Process
      - Data mining continues to evolve at the intersection of many fields like Artificial Intelligence, Machine Learning, Databases, statistics, etc..

1.2 TEXT MINING APPROACHES

- Text mining refers to the process of extracting high-quality information from text.

Method	Definition
Information Retrieval (IR)	Activity of finding information resources from a collection of unstructured data that satisfies the information need.
Natural Language Processing (NLP)	Sub-field of Computer science, AI, and linguistics – aims at understanding natural language using computers.
Information Extraction from Text (IE)	automatically extracting information or facts from unstructured or semi-unstructured documents • Name Entity Recognition (NER)
Text Summarization	Summarizing text to get a concise overview of a large document or collection of documents • Extractive summarization, abstractive summarization
Unsupervised Learning Methods	Finding hidden structure in unlabeled data • Clustering, Topic Modeling
Supervised Learning Methods	Machine Learning methods to learn a function/classifier from training data in order to perform predictions on unseen data. • Many Methods – Nearest Neighbors Classifiers, Decision Trees, rule-based classifiers, probabilistic classifiers
Probabilistic Methods for Text Mining	Probabilistic Techniques for Text Mining • Probabilistic Latent Semantic Analysis (pLSA) , Latent Dirichlet Allocation (LDA)
Text Streams and Social Media Mining	Mining of Text Data from Social Media Streams • Mining of Data from text feeds like twitter/google
Opinion Mining and Sentiment Analysis	The mining of data related to product reviews/user opinions • Frequently used with e-commerce and online shopping data

TEXT REPRESENTATION AND ENCODING

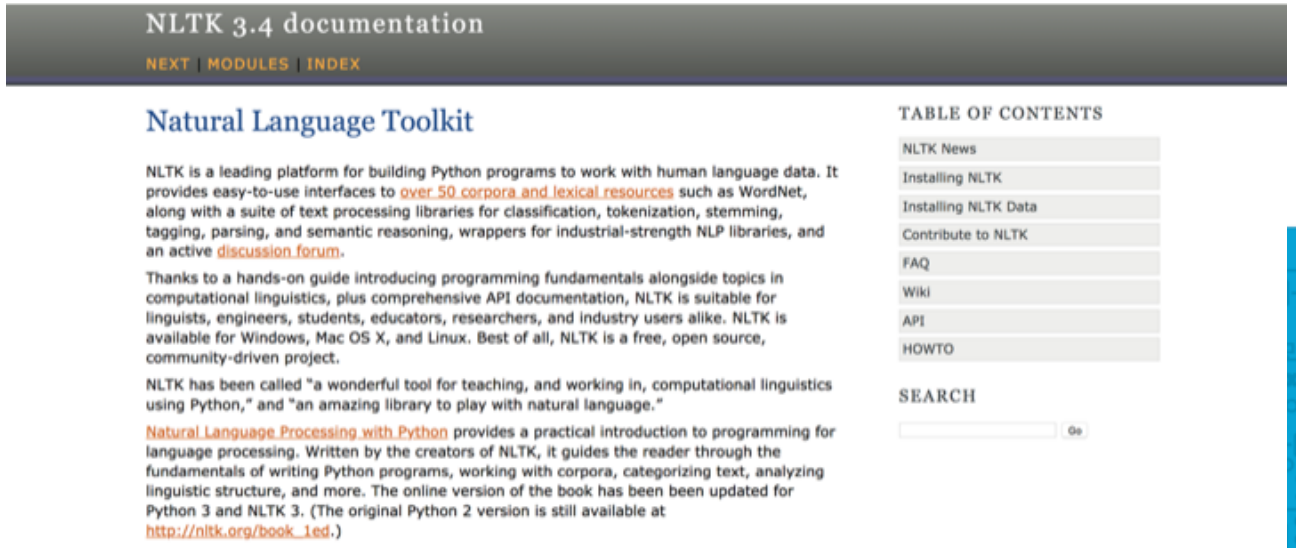
- Preprocessing is a key step for success before application of text mining algorithms much like in any other data related problem
- Includes Tasks like:
  - Tokenization
  - Filtering
  - Lemmatization
  - Stemming

2.1 TEXT PRE-PROCESSING

- **Tokenization –** "Tokenization is the task of breaking a character sequence up into pieces (words/phrases) called tokens."
  - Words, n-grams
  - Removal of punctuation/symbols is also included
- **Filtering –** Cleaning up of text data through removal of words:
  - StopWords – Most Common words in the English language
  - Words containing little information as well as very rare words.
- **Lemmatization –** Lemmatization considers the morphological analysis of words and groups together various forms of the same word so that they can be analyzed together.
- **Stemming –** Stemming aims to find the root of a word
  - Stemming algorithms are language dependent
  - Stemmer is the most widely used stemming method in English

LIBRARIES

NLTK



spaCy

2.2 VECTOR SPACE MODEL

- The most common way to represent documents is to convert them into numeric vectors
  - Vector Space Model (VSM)
  - VSM is broadly used in text mining and enables efficient analysis of large collections of documents

Lets Define the terms of this paper

- Given a collection of documents $\mathfrak{D} = \{d_1, d_2, \dots, d_D\}$
- Let $\mathcal{V} = \{w_1, w_2, \dots, w_v\}$
  - $\mathcal{V}$ is called vocabulary – The set of distinct words/terms in the collection.
- The frequency of term $w \in \mathcal{V}$ in $w \in \mathfrak{D}$ is shown by $f_d(w)$ and the number of documents having the word w is represented by $f_{\mathfrak{D}}(w)$
- The Term vector for document d is denoted by:
  - $\overrightarrow{t_d} = (f_d(w_1), f_d(w_2), \dots, f_d(w_v))$

2.2 VECTOR SPACE MODEL

- In Vector Space Models, each word is represented by a variable representing weight(importance) of the word in the document.
  - Boolean
    - This model assigns a weight $w_{ij} > 0$ to each term $w_i \in d_j$ and $d_j, w_{ij} = 0$ for any term that does not appear.
  - Term-Frequency-Inverse Document Frequency (TF-IDF)
    - The Most Popular term weighting method
      - Let q be this term weighting scheme, the weight of each word $w \in d$ is computed as below:

$$q(w) = f_d(w) * log \frac{|\mathcal{D}|}{f_{\mathcal{D}}(w)} \qquad (1)$$

    - Where $|\mathfrak{D}|$ is the number of documents in collection $\mathfrak{D}$
      - This normalization decreases the weight of terms occurring more frequently in the document collection.
    - Based on this term weighting scheme each document is represented by a vector of term weights $\omega(d) = (\omega(d, w_1), \omega(d, w_2), \ldots \omega(d, w_v))$
    - We can also compute similarity between documents $d_1 \; and \; d_2$ as seen below and one of the most common used measures is cosine similarity.

$$S(d_1, d_2) = cos(\theta) = \frac{d_1 \cdot d_2}{\sqrt{\sum_{i=1}^{v} w_{1i}^2} \cdot \sqrt{\sum_{i=1}^{v} w_{2i}^2}} \qquad (2)$$

FEATURE ENGINEERING FOR MACHINE LEARNING

BOOK REVIEW BY WILLIAM STEIMEL

CHAPTER 3: TEXT DATA: FLATTENING, FILTERING, AND CHUNKING

- How do we analyze the following paragraph of text with an algorithm?
  - "Emma knocked on the door. No answer. She knocked again and waited. There was a large maple tree next to the house. Emma looked up the tree and saw a giant raven perched at the treetop. Under the afternoon sun, the raven gleamed magnificently. Its beak was hard and pointed, its claws sharp and strong. It looked regal and imposing. It reigned the tree it stood on. The raven was looking straight at Emma with its beady black eyes. Emma felt slightly intimidated. She took a step back from the door and tentatively said, "Hello?" "
- We can see that it is a story between Emma and a giant raven but how could we represent this type of information to an algorithm?
- This first chapter discusses text extraction features, then covers how to filter and clean those features.

BAG-OF-X: TURNING NATURAL TEXT INTO FLAT VECTORS

- Bag-of-words is a simple vector representation and is generally just a list of word count statistics
  - A text document is converted into a vector of counts of each word
  - The original text is a sequence of words but bag-of-words are just counts without sequence (flat vector)

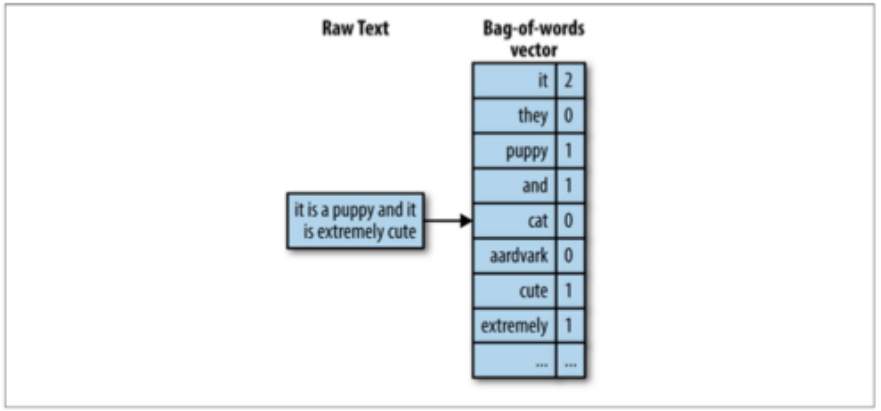


Figure 3-1. Turning raw text into a bag-of-words representation

BAG-OF-N-GRAMS

- Bag of N-Grams is an extension of bag-of-words
  - A word is a 1-gram (unigram)
  - After tokenization, the counting mechanism can combine individual tokens (1-grams) or count overlapping sequences (n-grams)
  - For example, the sentence "Emma knocked on the door" generates (4 different) *n*-grams (bi-grams) "Emma knocked," "knocked on," "on the," and "the door."
- Bag of n-grams creates a much sparse feature space and is more expensive to compute, store, and model as seen on the right.

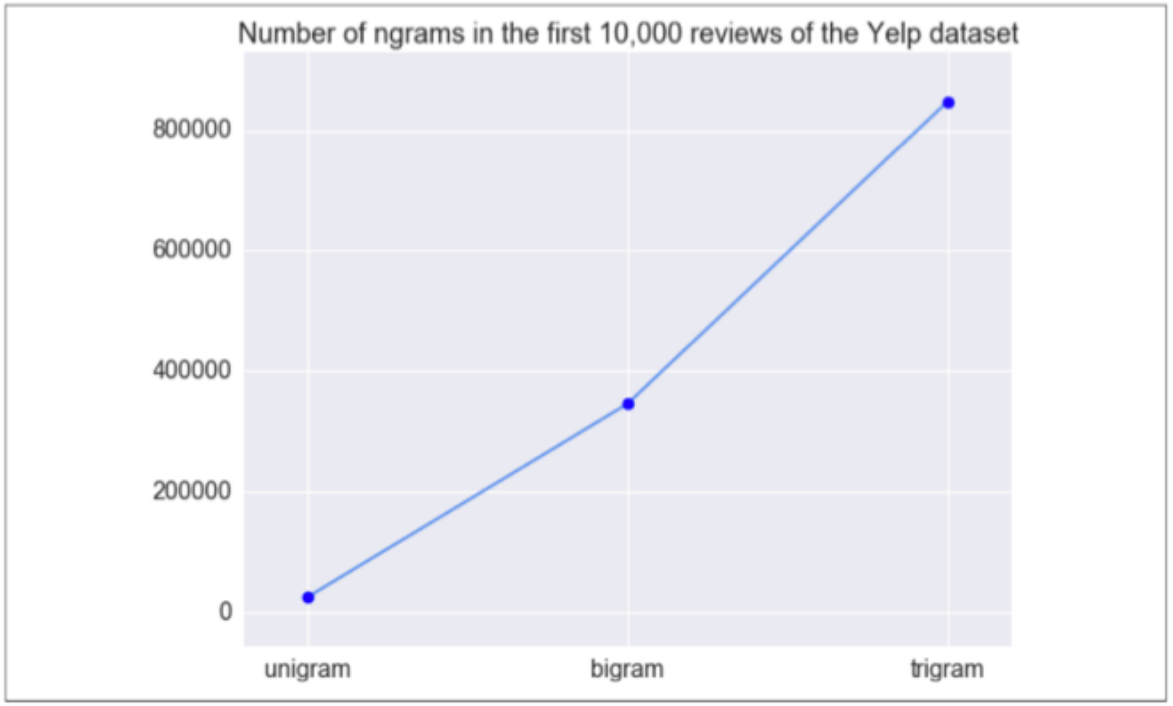


Figure 3-6. Number of unique n-grams in the first 10,000 reviews of the Yelp dataset

FILTER FOR CLEANER FEATURES

- Through text filtering we can reduce data dimensions and n-gram techniques can become more accessible.
  - **StopWords-** Common Words without much value that do not help us infer the meaning of the sentence.
  - **Frequent Words-** frequency statistics are great for filtering out corpus-specific stopwords or general-purpose stopwords.
    - Looking at the Yelp reviews dataset we can see that many of the most frequent words are stopwords
  - **Rare Words-** Rare words can be easily trimmed based on word count statistics.
    - Obscure Words, Rare Words
    - To a statistical model words appearing only once or twice are more like noise than useful information.
      - Rare words incur a large computation and storage cost for not much gain.
      - Rare words can also be binned into their own "Garbage" bin
  - **Stemming**- Different variations of the word are often counted as separate words
    - Flower – flowers
    - Swimmer- swimming
- Stemming is an NLP task that attempts to convert a word into its basic linguistic word stem form.

Stopwords

```
In [4]: from nltk.corpus import stopwords
        stop = stopwords.words('english')
        print stop
[u'i', u'me', u'my', u'myself', u'we', u'our', u'ours', u'ourselves', u'you', u'your', u'yours', u'yourself', u'yourselves', u'he', u'him', u'his', u'himself', u'she', u'her', u'hers', u'herself', u'it', u'its', u'itself', u'they', u'them', u'their', u'theirs', u'themselves', u'what', u'which', u'who', u'whom', u'this', u'that', u'these', u'those', u'am', u'is', u'are', u'was', u'were', u'be', u'been', u'being', u'have', u'has', u'had', u'having', u'do', u'does', u'did', u'doing', u'a', u'an', u'the', u'and', u'but', u'if', u'or', u'because', u'as', u'until', u'while', u'of', u'at', u'by', u'for', u'with', u'about', u'against', u'between', u'into', u'through', u'during', u'before', u'after', u'above', u'below', u'to', u'from', u'up', u'down', u'in', u'out', u'on', u'off', u'over', u'under', u'again', u'further', u'then', u'once', u'here', u'there', u'when', u'where', u'why', u'how', u'all', u'any', u'both', u'each', u'few', u'more', u'most', u'other', u'some', u'such', u'no', u'nor', u'not', u'only', u'own', u'same', u'so', u'than', u'too', u'very', u's', u't', u'can', u'will', u'just', u'don', u'should', u'now']
```

Frequent Words

Table 3-1. Most frequent words in the Yelp reviews dataset

Rank	Word	Document frequency	Rank	Word	Document frequency
1	the	1416058	21	t	684049
2	and	1381324	22	not	649824
3	a	1263126	23	s	626764
4	i	1230214	24	had	620284
5	to	1196238	25	so	608061
6	it	1027835	26	place	601918
7	of	1025638	27	good	598393
8	for	993430	28	at	596317
9	is	988547	29	are	585548
10	in	961518	30	food	562332
11	was	929703	31	be	543588
12	this	844824	32	we	537133
13	but	822313	33	great	520634
14	my	786595	34	were	516685
15	that	777045	35	there	510897
16	with	775044	36	here	481542
17	on	735419	37	all	478490
18	they	720994	38	if	475175
19	you	701015	39	very	460796
20	have	692749	40	out	460452

Rare Words

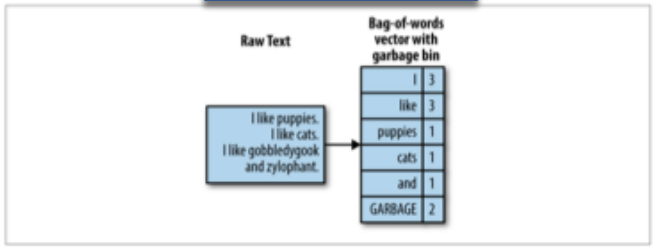


Figure 3-7. Bag-of-words feature vector with a garbage bin

Stemming

```
>>> import nltk
>>> stemmer = nltk.stem.porter.PorterStemmer()
>>> stemmer.stem('flowers')
u'flower'
>>> stemmer.stem('zeroes')
u'zero'
>>> stemmer.stem('stemmer')
u'stem'
>>> stemmer.stem('sixties')
u'sixti'
>>> stemmer.stem('sixty')
u'sixty'
>>> stemmer.stem('goes')
u'goe'
>>> stemmer.stem('go')
u'go'
```

CHAPTER 4: THE EFFECTS OF FEATURE SCALING: FROM BAG-OF-WORDS TO TF-IDF

- Bag-of-Words is simple and effective but ideally we'd want a representation that highlights meaningful words.
- Tf-idf is a simple modification on the bag-of-words approach – term frequency inverse document frequency
- Instead of looking at counts like bag of words it looks at a normalized count where each word is divided by the number of documents the word appears in.
  - If a word appears in many documents its Tf-idf is closer to 0
  - If it does not appear in many documents it is a much higher value near 1.
- Tf-idf makes rare words more important and reduces the effect of more common words in vector space.

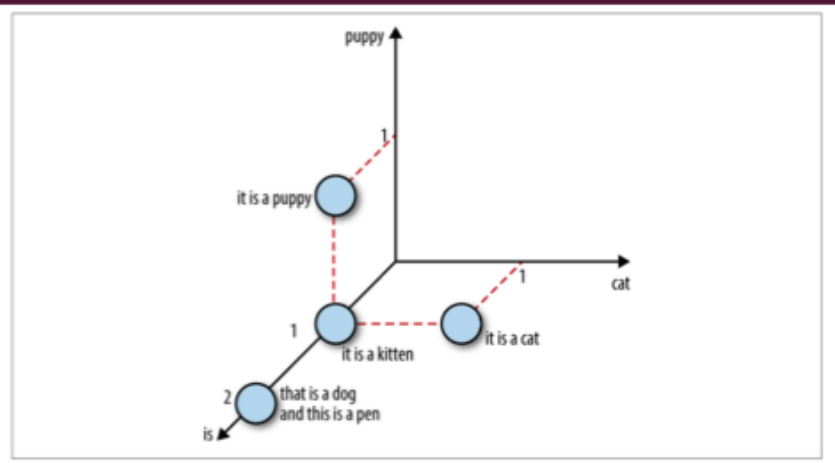


Figure 4-1. Four sentences about dogs and cats

After Tf-idf
(is becomes nearly eliminated as a feature)

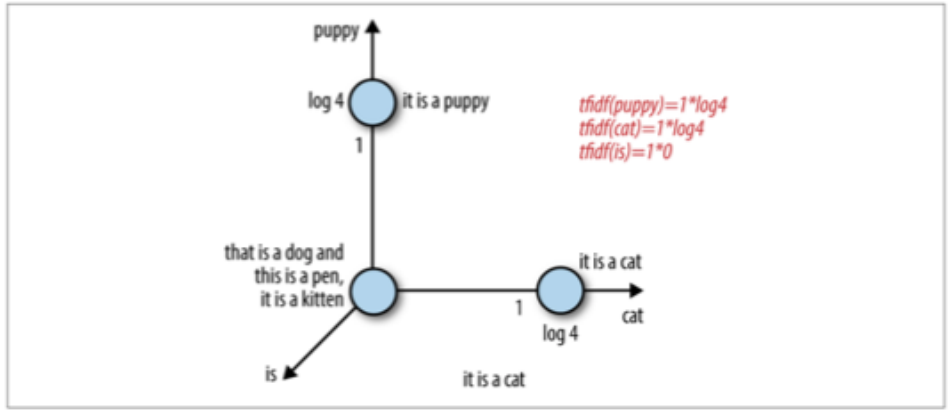


Figure 4-2. Tf-idf representation of the sentences in Figure 4-1

CONCLUSION

- These two Chapter discussed simple text featurization techniques:
  - Bag of Words – Simple, easy to compute, and useful for classification but simplistic.
  - Bag of N-grams – Bag-of-n-grams is an extension to Bag of Words that more accurately captures word relationships but leads to more sparse data representations.
  - Tf-Idf – A method of text scaling to highlight more meaningful words
- These methods turn a sequence of text tokens into a vector that represents text and can be utilized in machine learning modeling.