ISM 6359 – Text Mining Exam/Reflection (100 pts)
DUE: see Canvas for due date

DIRECTIONS: Answer any **Two** (2) of the three possible questions below.

1. Part 1: The authors proposed an alternative to the CRISP-DM Framework for Text Mining, compare and contrast their framework to CRISP-DM. Part II: Sampling is a key process in data collection, discuss the many types of sampling and the pros and cons of each method. Be sure to answer both parts of this question. (50%)

Not chosen.

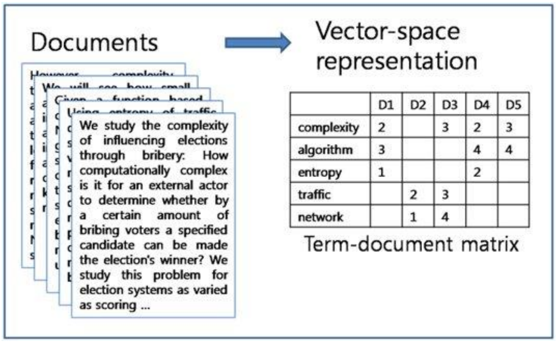
2. Much of your time building text analytical models consists of text preparation, sometimes called preprocessing. Part 1: List and explain many of the common text preparation functions that you might perform to your data sets in order to get the ready for text analytics. Part 2: Discuss the following terms/acronyms and tell me how they relate to Text Mining: DTM/TDM, TF, IDF, and TF-IDF. Be sure to answer both parts of this question. (50%)

Generally, after the planning stage, the data should have been collected, and the goal well defined. The next step requires the data to be "massaged" for analysis. Moreover, each record should have been catalogued with a unique identifier used to refer to that instance. In text mining, the instances are known as docs. Consequently, a doc has made up of many characters that form words or terms. As far as I am concerned, much of the text preparation have its roots in natural language processing. Each step taken removes unnecessary from the original text. Having said all of that, it is important to understand those steps that makes text analytics efficient.

- <u>Tokenization</u>: it is the art of transforming words into attributes. The operator breaks unstructured data and natural language text into chunks of information that can be considered as discrete elements. It immediately turns that numerical data suitable for a machine learning model. In addition to that, converted data may be used in a machine learning pipeline as features that trigger more complex decisions or behavior. Simply put, tokenization helps to split sentences, words, characters, or sub words.
- <u>N-Grams</u>: They are an alternative to single words in the tokenization process. A set of co-occurring words within a given window and when computing the n-grams it typically moves one word forward. For instance, if **X = Num of Words** in each sentence K, the number of n-grams would be X – (N – 1). In Text mining, the intuition is to use tokens such as bigrams "N = 2" in the model rather than just unigrams. The reason for that is to yield any significant improvement to the classification **-** that's not always the case.
- <u>Standardization or case</u>: this step levels the playing field by making the terms in each of the documents comparable. For example, the scientist does not want *tree, Tree, or TREE* to be considered separate items because one is in lower case, one in upper case, and the other the initial consonant capitalized. Standardize prevents such a possibility to happen by converting the terms in the text to lower case.
- <u>Stop Words Filter</u>: It is useful to drop frequently used filler words, which add no value to the analysis. Examples are *the, be, to, and, as, that, which, or, of.* Even though, they are commonly used in the English dictionary, they serve as a grammatical purpose which provides little information in terms of content.
- <u>Stemming & Lemmatization</u>: These are perhaps the last stages of preprocessing. Both involve breaking words down to their root word. Stemming is a process that stems or removes last few characters from a word, often leading to incorrect meanings and spelling. For example, stemming the word "Caring" displaying kindness and concern for others, would return "Flu". Generally, it is employed in case of large datasets where performance is an issue. Whereas Lemmatization, considers the context and converts the word to its meaningful base form. With the same example, "Caring", it would return "Care". In addition, lemmatization helps when the scientist encounters words with different meanings. For instance, the word "Ground" can mean an area of land or sea used for a specified purpose or the past tense of the verb grind. As far as I am concerned, lemmatization can be computationally expensive since it involves look-up tables and what not.

Before discussing the terms, it is important to introduce the term Vector Space Model. It is a generalization of the Bag of Words model. In the VSP, each document from the corpus is represented as a multidimensional vector. Consequently, each unique term from the corpus represents one dimension of the vector space. A term can be a single word or a sequence of words. The number of unique terms in the corpus, determines the dimension of the vector space.

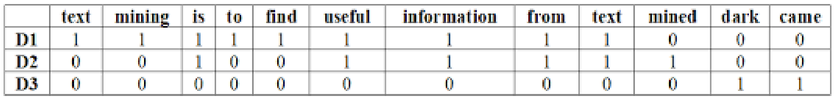
In a VSM, the corpus is represented in the form of the Term Document Matrix or TDM. It represents documents vectors in matrix form in which the rows correspond to the terms in the document, columns correspond to the documents in the corpus, and cells correspond to the weights of the terms.



We study the complexity of influencing elections through bribery: How computationally complex is it for an external actor to determine whether by a certain amount of bribing voters a specified candidate can be made the election's winner? We study this problem for election systems as varied as scoring ...

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

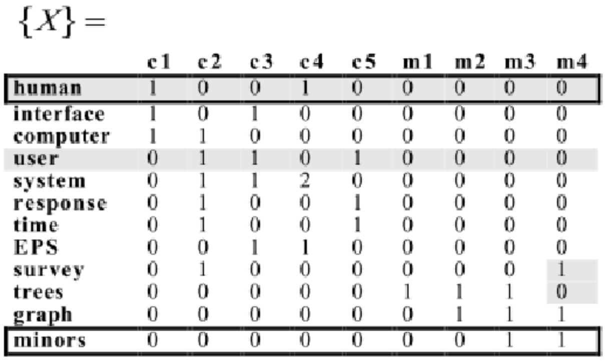
In contrast, DTM or Document Term Matrix, is obtained by taking the transpose of TDM. Rows correspond to the documents in the corpus, columns correspond to the terms in the document, and the cells correspond to the weights of the terms.



	text	mining	is	to	find	useful	information	from	text	mined	dark	came
D1	1	1	1	1	1	1	1	1	1	0	0	0
D2	0	0	1	0	0	1	1	1	1	1	0	0
D3	0	0	0	0	0	0	0	0	0	0	1	1

Having said all of that, there are various approaches for determining the terms' weights. Two operate locally such as "TF" term frequency, "IDF" inverse document frequency, whereas, "TF-IDF" term frequency - inverse document frequency, operates both locally and globally.

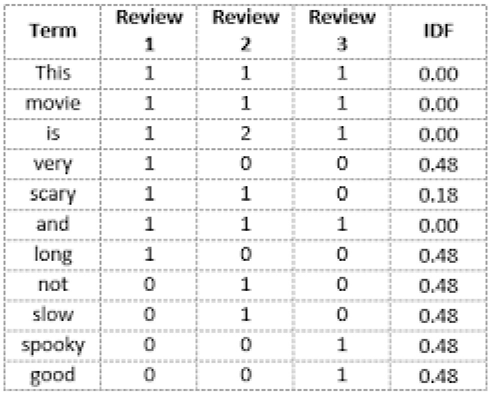
In the case of the term frequency, the weights represent the frequency of the term in a specific document. The underlying assumption is that the higher the term frequency in a document, the more important it is for that document.



$\{X\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

In the case of inverse document frequency, the underlying idea is to assign higher weights to unusual terms, for example, to terms that are not so common in the corpus. IDF is computed at the corpus level, and thus describes corpus as a whole, not individual documents.



Term	Review 1	Review 2	Review 3	IDF
This	1	1	1	0.00
movie	1	1	1	0.00
is	1	2	1	0.00
very	1	0	0	0.48
scary	1	1	0	0.18
and	1	1	1	0.00
long	1	0	0	0.48
not	0	1	0	0.48
slow	0	1	0	0.48
spooky	0	0	1	0.48
good	0	0	1	0.48

Lastly, in the case of TF-IDF, the intuition stems from the ability to value those terms that are not so common in the corpus (relatively high IDF), but still have some reasonable level of frequency (relatively high TF). This is the most common used metric for computing term weights in a vector space model.

3. We’ve now done Classification and Clustering of both Data and Text in class. From the remaining three foci of text analysis (Sentiment Analysis, Topic Modeling, and Latent Semantic Analysis), pick any TWO of them and describe what they do and how they do it, write your answer as if speaking to a computer savvy person who has never done Text Mining. Remember to reflect on TWO of the three. (50%)

Sentiment analysis is the process of detecting positive or negative sentiment in text. It is often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers behavior. It tends to focus on the polarity of a text (positive, negative, neutral) but it also goes beyond polarity to detect specific feelings, emotions, urgency, and intentions. Depending on the business’s needs, it seems possible to define and tailor categories to meet the sentiment analysis needs, concurrently. There are several analyses.

- Graded sentiment analysis, if polarity precision matters. It could be used to interpret stars rating in a review, for instance.
- Emotion detection analysis, allowing to go beyond polarity to detect emotions, like happiness, frustration, anger, or sadness. Unfortunately, there is a down side to it, people express emotions in different ways.
- Aspect-based sentiment analysis, good to categorize which aspects or features people are mentioning in a positive, neutral, or negative way.
- Multilingual sentiment analysis.

Human beings express their thoughts and feelings more openly than ever before. Automatically, analyzing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can tailor products and services to meet their customers’ needs. Following are some benefits of it – even though I think the list might be endless.

It helps to sort data at scale, there’s just too much business data to process manually. Sentiment analysis may help an enterprise to process huge amounts of unstructured data in an efficient and cost-effective way.

Sentiment analysis can identify critical issues in real-time. For example, is an angry customer about to churn? This model can help the company immediately identify these situations and take prompt actions.

Tagging text by sentiment is highly subjective, influenced by personal experiences, thoughts, and beliefs. A company can apply the same criteria to all its data to improve accuracy and collect better insights.

Topic modeling is a machine learning technique that automatically analyzes text data to determine cluster words for a set of documents. It is unsupervised because it does not require labels and training data. Given the previous assumptions, it is a quick and easy way to start analyzing the data. Unfortunately, it does not guarantee accurate results. To obviate, businesses invest time on a supervised technique called topic classification model. It requires to know the topics of a set of texts before analyzing them. Thus, data is tagged manually so that a topic classifier can learn and later make predictions by itself.

Topic modeling involves counting words and grouping similar word patterns to infer topics within unstructured data. By detecting patterns such as word frequency and distance between words, a topic model clusters feedback that is similar, or a word/expression that appear most often, a scientist can quickly deduce what each set of texts are talking about.

Topic modeling and topic classification are two worlds apart, although being the most commonly used topic analysis techniques. The former, requires less manual input than the latter, because it does not need to be trained by humans with labels. However, it requires high-quality data. Whereas the latter, deliver neatly packaged results with topic labels. Of course, it requires training and patience, but if labeling has been done accurately, the scientist will be rewarded with a model that can accurately classify unseen texts according to its topic.

Topic modeling divides a corpus of documents in two. 1) a list of the topics covered in the corpus, and 2) several sets of documents from the corpus, grouped by the topics they cover. It seems that each document comprises a statistical mixture of topics, for example, a statistical distribution of topics that can be obtained by adding up all of the distributions for all the topics covered.

Try to spend 30-45 minutes on each of your TWO questions. It is suggestion that you spend the first ½ of your time (per question) just typing from memory, and then use the last ½ of the allotted time using your available resources (book, Internet, lecture notes, etc. – but no other humans – no classmates, peers, spouses, bosses, mentors, etc.). Please do NOT copy content from the book, Wikipedia, or other websites, use your own words.