Module-3:
Data Exploration & Visualization

3.1 Why is Exploratory Data Analysis (EDA) for text data required?

Exploratory Data Analysis or EDA is important for both structured and unstructured data As it gives us insights on the underlying features of the data and helps us find trends ,patterns so that we can apply the appropriate models as well as give our own hypothesis on the data. The EDA for structured data and unstructured data differ greatly. Common visualization tools like bar plots.histograms,correlation matrices are not suited for textual data.[1] Visually representing the contents of a text document is one of the most important steps in any NLP problem. It helps us get a heuristic idea of the events in the summary and on how we can create storylines from that data. However, we can use histograms and bar plots for problem statements that predict categorical data.For example: Sentiment Polarity Distribution using tools like Plotly.[2] Visualization of textual data comes after the text cleaning and preprocessing stage as described in the previous module.This blog mainly focuses on the the sentiments of the reviews on amazon products. In this blog, a Document Term Matrix was created to provide frequency of words in a corpus. It helps in analysing the frequency of words in each document.

Exploratory Data Analysis on Amazon Product Reviews - The first step is to plot the most common words which can be obtained from the Document Term Matrix and visualize it using WordClouds. WordCloud is a visualization tool that gives the frequencies of different tools in the document. It gives more importance to frequent bigger words when compared to the less frequent words. It was found out that LOVE,EASY,BUY and GREAT were the most common words in the document. This shows us that probably most people love purchasing their products on amazon. It also can give us a start in the error analysis in cases we get much varying predicted outputs. To find the polarities, they have used the TextBlob library that classifies the positive and the negative sentiments. This was visualized easily using barplots where it was concluded that Amazon had room for improvement in their Fire Kids Edition Tablet. Another way we can analyse textual data is by analysing the readability of the data that is how understandable the text data is. There are various methods to achieve the same which include Flesch Reading Ease, Dale Chall Readability Score and Gunning Fox index.

Flesch Reading Ease - Flesch Reading Ease is a technique used to understand how difficult a passage in English is easy to understand. Higher the score, easier the material is to understand and read. Flesch Reading Ease test is given by :

$$206.835 - 1.0.15(\text{total words}/\text{total sentences}) - 84.6(\text{total syllabus}/\text{total words})$$

Dale Chall Readability Score - It provides a comprehension difficulty that readers come upon while reading a text. It uses a list of 3000 words that groups of fourth grade American students could reliably understand. The Dale Chall Readability score is :

$$0.1579((\text{difficult words}/\text{words}) \times 100) + 0.0496(\text{words}/\text{sentences})$$

Gunning Fog Index - It’s a readability test that estimates the years of formal education required by a person to understand on the first reading . The formula for the test is :

0.4[(words/sentences) + 100(complex words/words)]

3.2 What are the basic statistics on the text data?

[3]Word Frequency Analysis - Word Frequency involves the occurrence of every n-gram model that can be uni gram or bi gram in a corpus that can be counted to classify it into a potential category.In the example given in this particular blog, they have used term frequency to categorize the words into categories of *trustworthy* and *untrustworthy*.
For example consider the corpus of words to be :

He is a cheater and never stops lying.
She is incompetent and has a history for lying
He is a lying racist.

The most common words in this text corpus are *is,a* and *lying* in which *lying* is the most significant word and *is,a* can be removed in the preprocessing stage. If the test corpus contains the word *lied* or *cheated*, by using lemmatization, we would be able to correctly classify the words to the untrustworthy category as lemmatization takes into consideration the base words.
[4]Collocation - Collocation helps in giving us insights in how often two words co-occur. Analysing bi-grams and trigrams can give us the hidden semantic patterns in the text which is more efficient than using uni grams. For example *‘water park’* can be a collocation in a given context and *‘water heater’* in another context can be very useful in analysing text data.
Concordance - Concordance is used to identify the context and instances of a word or a set of words. The example used in the blog uses *simple* as a target and analyses the preceding context as well as the following context of the target in each of the reviews. For example :

Preceding context : *Hate the new update*
Target: *simple*
Following Context : *as that*

Preceding context: *It’s quite good*
Target: *simple*
Following Context: *to use*

Simple is used in two complete different ways expressing negative as well as positive emotion. This method gives us insights on how users use the target word and can also be used to eliminate human ambiguity to some extent. From this example we can make out two different contexts and analyse more complex phrases.
[5] Handling unknown words (OOV or Out-Of-Word Vocabulary Model) - The limitations of word embeddings is that they are learnt by Natural Language Models and therefore words must have been part of the training dataset. If we encounter an OOV word, we sequence the words in the sentence and try to predict the meaning of the word in the sentence by comparing it with other sentences. We aim at producing embeddings for the OOV words

depending upon the context of the OOV word. We use Bi-Directional RNN LSTM. We predict the most probable word embeddings in the place of the OOV word and then taking the weighted average of their mapped word embeddings.The step by step process for this includes the preprocessing of a large corpus of text with tokenization ,encoding the words as unique integers and then the Embedding Prediction Model to predict the OOV words. The Forward as well as the Backward sequences of text are prepared as we are required to predict the embedding with respect to the former and the latter words of the OOV text. These sequences form the inputs and are required to have the same length. Next, we define the layers of the Neural Network required. The first layer is the embedding layer to get the vector representation of each sequence. The second layer is the Bi Directional layer that can be tuned to fit the training corpus. The last layer is the output layer with the softmax activation that outputs the probability distribution of the vocabulary based on the sequence we input the Neural Network. We take in a sample input text and locate the OOV word in the vocabulary. The embedding for the OOV word is generated and updated. In the end, this embedding is added to the pretrained models such as GLOVE,Word2Vec etc.
[6]Feature Engineering - The most important part of any text classification is feature engineering where we transform raw text data into embedded vectors or features that can be passed into the model. We may even want to consider checking if the corpus belongs to the same language or not.All of these steps can be included in the preprocessing of the textual data before a classification model is created.
Length Analysis - It is important for us to visualize the count of words and the lengths of the sentences in the corpus for sequence models and it gives us insights on the data. There are many ways to measure length in textual data. Word Count gives the count of the number of words or tokens in the corpus, character count gives the number of characters of each token, sentence count counts the number of sentences, average word count gives the sum of the number of words divided by their word count, average sentence count gives the number of sentences divided by the sentence count. Using these variables, we can find the distribution of those variables with respect to the target. In the example given this blog, we divide the dataset into 3 categories that is Entertainment,Politics,Tech. We compare the histograms and densities of the samples. If the distributions are different, then the variable is predictive because it can be easily classified into the 3 categories. We check these plots for each of the variables that is character count,word count etc with the target (y) and arrive at some insights as to what can be a predicting factor for the samples. Since the distribution plots turned out to be similar, we can conclude that character count is not really a good factor to analyse the category of the data.
The next analysis done in this blog is the sentiment analysis and whether the sentiment can be an effective way of classifying the data point into the three categories. TextBlob and Vader can be used as pretrained models to extract the sentiments of a text. Next we find if there is any pattern between the categories and the sentiments. It was found out that the Politics news is more skewed towards the negative tail while the Tech news is skewed more towards the positive tail.
Named Entity Recognition(NER) - NER is the process of tagging an entity in the dataset into

categories such as person, organizations, locations, buildings ,objects etc. SpaCy is an open source tool that is used to recognize such entities. To make use of this feature for visualizations, we run this tool on every observation in the dataset. In this dataset, we take each observation to be a news headline. For each text , we make a dictionary with the entity,tag and number of times it occurs. For example :

Will Smith ***Person*** *Joins Diplo* ***Person*** *And Nicky Jam* ***Person*** *For The 2018 World Cup's* ***Event*** *Official Song*

{('Will Smith','PERSON'):1,
('Diplo','PERSON'):1,
('Nicky Jam','PERSON'):1,
('The 2018 World Cup's','EVENT'):1}

Next we create a column for each category and find the number of entities in each tag.

PERSON :3
EVENT:1

We can visualize the three categories that are ENTERTAINMENT,POLITICS and TECH with respect to the most frequent tags found. For example in 'Entertainment' , we find that the 'Person' tag is most frequently found when compared to the other tags. We may run into problems while considering names like *'Will Smith'* with stopwords like 'will'. We use SpaCy to recognize a person's name, we can use **name detection** to change the string.

3.2 What are the steps involved in EDA of text data?

Word Frequency
A single word can tell us a lot about which category the sentence should belong to. We can visualize these words using n-grams. An n gram is a sequence of words given from a sample text. We can visualize the most frequent words in each category using bar plots. We can create new features based on how frequently the n grams appear in a category. In the Bag Of Words approach we use all words as features. In this example, we find that 'box office' is most commonly used in Entertainment, 'President' in politics and 'apple' in tech. We use these as features to represent the text corpus using CountVectorizer that returns a matrix of each of these token counts. We can visualize these using Word Clouds. The size and color of the font signifies the frequency of each tag.
Word Vectors - Neural Network Architectures used for feature learning and vectorization of text corpus has been replacing traditional N gram models. The real numbered vectors that are a result of word transformation are called Word Embeddings. These are estimated by calculating the probability distributions of what words may appear before or after the concerned word. Using these vectors, it would be fairly simple to estimate the similar words using the Euclidean distance between the vectors. A popular package used to obtain word embeddings is the gensim model. Gensim is an unsupervised model for obtaining vector representations of vocabulary size 300. We can figure out the most similar word from this model and plot these vector representations by reducing their dimensions to only 2. This can be done by many dimensionality reduction techniques like PCA or t SNE.
Topic Modelling - Gensim package is widely used in Topic modelling. Topic modelling is

the process in discovering the most appropriate topics for a set of documents. Statistical models are available that can explain why some data points are similar to others.(Latent Dirichlet Allocation). We will need to specify how many topics we desire in a text corpus beforehand. We can visualize all the n grams divided into their respective topic and have an idea of the models performance by inspecting how many of those n grams are actually similar.

[7] Qualitative Comparisons
As explained in the previous section, word embeddings give us a great description of the text with respect to the dataset in the form of numerical values. They can provide us with qualitative comparisons of words as they represent words with high dimensional vectors whose dimensions can be reduced for visualizations using dimensionality reduction techniques. We can provide understandable visualizations using bubble charts where the color can imply the average word length and the similarity measure can be based on anything like the frequency of the bigrams appearing in a particular topic or the vocabulary length etc. Another visualization technique that can be used is the treemap visualization. This visualization is effective when we want to figure out length related patterns in the data. For example if we want to differentiate between complaints and praise, we generally observe that the complaint messages are usually longer than praises. The treemap generally divides the data into a particular topic and checks for any such differentiating tokens in those samples.The box size indicates group sizing and the color indicates the average length.
In some cases we might also want to compare the proportions of complaints through various products or companies. Here a stacked bar might come in handy. Pie charts can also be visualized if we are comparing proportions. We can also use the top 50 bigrams and compare the frequencies of these bigrams for each company or product. We can get insights into which product might be overperforming the other or which company has better feedback from the public. We can also visualize the increase or decrease in the complaint rate for any company over a period of time or plot the histograms of all companies complaints over a period of time. However, we can get most out of the Exploratory Data Analysis by visualizing the n grams and using word embeddings for similarity checks.

References:

[1]https://www.kdnuggets.com/2019/05/complete-exploratory-data-analysis-visualization-text-data.html

[2]https://www.analyticsvidhya.com/blog/2020/04/beginners-guide-exploratory-data-analysis-text-data/

[3]https://www.mosaicdatascience.com/2015/10/12/text-mining-word-frequency-models/

[4] https://monkeylearn.com/text-analysis/

[5]https://medium.com/@shabeelkandi/handling-out-of-vocabulary-words-in-natural-language-processing-based-on-context-4bbba16214d5

[6]https://towardsdatascience.com/text-analysis-feature-engineering-with-nlp-502d6ea9225d#:~:text=The%20most%20important%20part%20of,to%20build%20a%20classification%20model.

[7]https://medium.com/plotly/nlp-visualisations-for-clear-immediate-insights-into-text-data-and-outputs-9ebfab168d5b