

**CSE523 Machine Learning**

**Weekly Report 2**

**Group Name: Precision Précis**

| **NAME** | **ENROLLMENT NUMBER** |
| --- | --- |
| Twinkle Popat | AU2040216 |
| Kathan Bhavsar | AU2040170 |
| Rushali Moteria | AU2040210 |
| Neel Buddhdev | AU2040176 |

1. **Tasks performed in the week.**

As a part of feature extraction we explored and implemented the method of Bag of Words. We use this method to represent the text data as numerical features. The following steps are involved in the method.

**Basic Concept of Bag of words:**

* Tokenization: The text or input data is converted into tokens or individual words.
* Vocabulary Count: A set of unique words is created. Firstly, an unique index is assigned to each word. Then, a dictionary of words is declared and a loop is run over the tokenized words. If the word is present in the dictionary then the value of the counter is increased for that word by 1 and if the word is not present then it is added to the dictionary. This is how the frequency of each word is counted.
* Vectorisation: Finally a bag of words is built, which is a vector representation of the text data. The matrix tells whether a particular word is present in the sentence or not. If a particular word is present in the sentence then the matrix entry is assigned value 1 otherwise it is assigned value 0. Thus if the word in the sentence is a frequent word, then it will have value 1. We use this frequency matrix to find scores of sentences in our project.

Note: Vocabulary counting and Vectorisation is done by CountVectorizer() in the code implemented below.

**Code implementation:**

* We will declare all the necessary arrays needed as follows.

| article = [] refehighlights = [] originalLen = [] |
| --- |

* We will append the texts in the article array for future use.
* We also use the originalLen array for comparing the length of the final summary with the original article.
* We will be using a refehighlights array to store the summary generated by a human from the data set so that it can be employed to compare the original summary (human generated) with the final summary of our model.

| # Sample array to hold all the article article.append(text1) article.append(text2) article.append(text3) |
| --- |

| # Sample array to hold the original length of the article originalLen.append(len(text1)) originalLen.append(len(text2)) originalLen.append(len(text3)) |
| --- |

| # Sample reference array to hold all the highlights  refehighlights.append(refSumm1) refehighlights.append(refSumm1) refehighlights.append(refSumm1) |
| --- |

* We initialize an empty list where we store the generated summaries and their length.

| # Clearing the array everytime it runs generatedArrayBOW = []  newLenBOW = [] |
| --- |

* Then we create a vectorizer, which is an instance of the Countvectorizer() class which is used to create a bag-of-words model.

| # Create the bag-of-words model vectorizer = CountVectorizer() |
| --- |

* Then we loop over the entire article and split the dataset into a list of sentences. Then we convert this list of sentences into bag of words.

| i = 0 while(i<len(article)):  sentences = article[i].split(".")    # Transform the sentences to vectors  sentence\_vectors = vectorizer.fit\_transform(sentences) |
| --- |

* Then we find the score of each of each sentence. We calculate the sum of each feature or word in each sentence. Then we normalize the scores.

| # Calculate the score for each sentence  scores = sentence\_vectors.sum(axis=1)   # Normalize the scores  normalized\_scores = scores / sentence\_vectors.sum(axis=1).sum() |
| --- |

* Then we select the top sentences (with high scores) for the summary and then print the summary.

| n = 2  indices = np.argsort(sentence\_vectors.toarray().sum(axis=1))[-n:]  imp\_sen = [sentences[k] for k in indices]   summary = " ".join(imp\_sen) + "."  print(summary)  generatedArrayBOW.append(summary)  newLenBOW.append(len(summary))     print("------------------------------------------------------------------------------")  i+=1 |
| --- |

1. **Outcomes of the tasks performed.**

Following is the output of summaries generated from selecting n most important sentences from the original article.

| [' Bishop John Folda (pictured) of the Fargo Catholic Diocese in North Dakota has exposed potentially hundreds of church members in Fargo, Grand Forks and Jamestown to the hepatitis A The bishop of the Fargo Catholic Diocese in North Dakota has exposed potentially hundreds of church members in Fargo, Grand Forks and Jamestown to the hepatitis A virus in late September and early October.',  " 'We weren't able to take breath or blood tests from him immediately and although blood taken several hours after the collision showed he was marginally under the limit, we maintain he would have been over the limit at the time of the collision and in summing up today, the judge agreed â€ Mr Eccleston-Todd took the decision to pick up his mobile phone whilst driving and, either reading or replying to this text message, was so distracted that he failed to negotiate a left-hand bend, crossing the central white line into the path of Miss Titleyâ€TMs oncoming car.",  " The Cod Army are playing in the third tier for the first time in their history after six promotions in nine years and their remarkable ascent shows no sign of slowing with Jamie Proctor and Gareth Evans scoring the goals at Glanford Park Bristol City, who beat Nigel Cloughâ€TMs men on the opening day, were held to a goalless draw by last season's play-off finalists Leyton Orient while Chesterfield, the League Two champions, were beaten 1-0 by MK Dons, who play Manchester United in the Capital One Cup in seven daysâ€TM time."] |
| --- |

Comparison of the length of original article and generated summary to evaluate the performance of the summary (in future we will be using “Rogue” library of python to evaluate the score of generated summary in comparison to human summary).

| Original Text: 1213 , Summary: 389 Original Text: 4785 , Summary: 566 Original Text: 3461 , Summary: 529 |
| --- |

1. **Tasks to be performed in the upcoming week.**

While working on the bag of words method, we find the frequency of words, however we could assign importance or weight to the words since it uses a binary approach and thus it weighs the presence of each word in a sentence equally. Thus it does not help to do proper sentiment analysis of the text. In order to further improve the analysis of the text we will use TF-IDF Vectorization.