EARNING CALL TRANSCRIPTS

Methodology

Mentor: Manpreet Makkad

Presented by:

	GEID
Nimit Shah	1011139629
Rahasya Barkur	1011139547
Kancharapu Anil Kumar	1011139542
Annlin Chacko	1011139585

TASK 0: Data Extraction

- Aim : To extract earnings call transcripts of at least 10 companies from seekingalpha.com
- Method of approach :
  a. Login into seekingalpha.com remotely to get full access to the transcripts
  b. Open https://seekingalpha.com/symbol/<token>/earnings/transcripts on the remote browser, where <token> is replaced by the company keyword that seekingalpha uses. (Ex., GOOG for google, C for Citicorp)
  c. Make the browser repeatedly scroll to the end of the page, so as to load the invisible links.
  d. Extract only the relevant URLs (URLs containing the earnings call transcripts only) from the page.
  e. Parse all the URLs, and extract the transcript from those pages.
  f. Repeat a-e for different companies

Data Extraction

- Libraries Used :
  - Selenium - for remotely opening web pages
  - BeautifulSoup - for accessing and parsing the raw html data from a URL
  - Requests - for easily making HTTP requests
- Issues faced :
  - A common issue in web scraping is dealing with captcha. When a website gets accessed multiple times from a single IP, it throws captcha for security purposes. Our web scraping code cannot currently handle captchas.
- Scope of future work :
  - Rotating IP addresses using Selenium
  - Make proxied requests to seekingalpha using newly active proxies listed within the Free Proxy List.

TASK 1: Summarization of the calls

Method 2: Sentence compression for important sentences

Brief description: for every token(word) in the input sentence it’s predicted whether it goes to the summary or not, which translates into a sequence labeling problem with just two labels: one and zero. This makes a binary classification problem. Model is a LSTM model followed by binary classification.

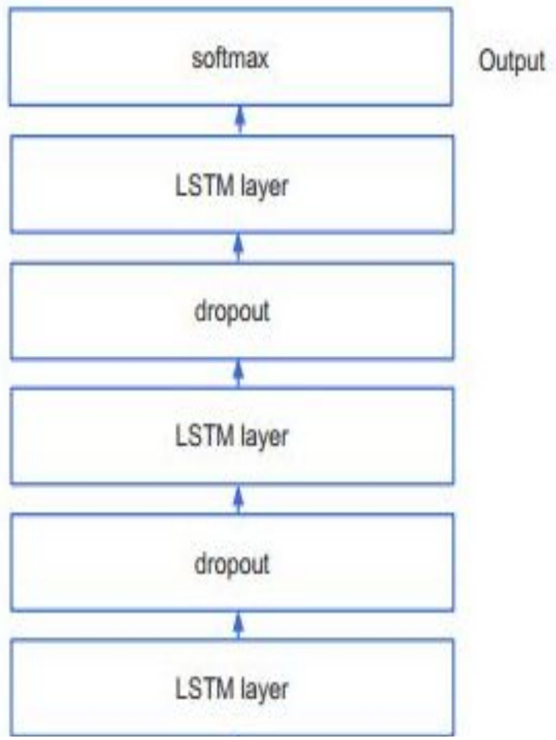
Important sentences can be calculated using textrank algorithm.

Pre processing : Embedding is done using Word2Vec model.

Model : Architecture block diagram is as follows. A Greedy approach is followed. Out of the 3 LSTM layers the last one is kept unidirectional to reduce the ambiguity while testing, other two are Bidirectional.

Input - Embedded text / Vectorised text

Output - binary array which determines if the word has To be included in the summary.



Dataset used: Manually labelled by going through every word and guessing if it goes to the summary

Results obtained after testing on our transcript was satisfying. So dataset was created with the available transcripts.

Accuracy : 85.6%

No.of epochs : 20

No.of trainable parameters: 143,169

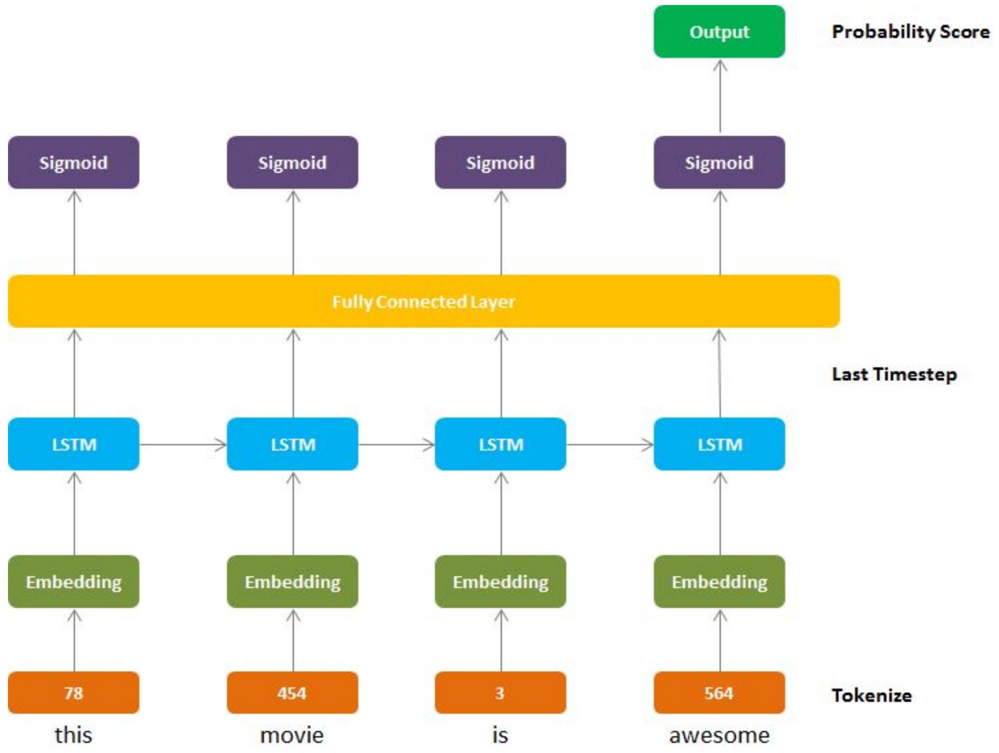
TASK 2: *Sentiment Analysis*

- Previous Work : Sentiment analysis using NLP Techniques such as TextBlob and VADER sentiment analysis.
- Sentiment Analysis using Deep Learning Techniques : LSTM

What is LSTM ?

- LSTM is a Recurrent Neural Network architecture, commonly used for creating sequence to sequence models.

LSTM model Architecture



Sentiment Analysis using LSTM

- Dataset used: IMDb movie review dataset
- Accuracy : 85.3%
- Embedding dictionary used : GloVe

WHY LSTM over classical NLP methods ?

- LSTMs are neural networks, which have feedback connections.
- This enables them to learn long term dependencies, therefore, making them very efficient in processing a sequence of input (In our case, several sentences of the summarized earnings call transcripts)

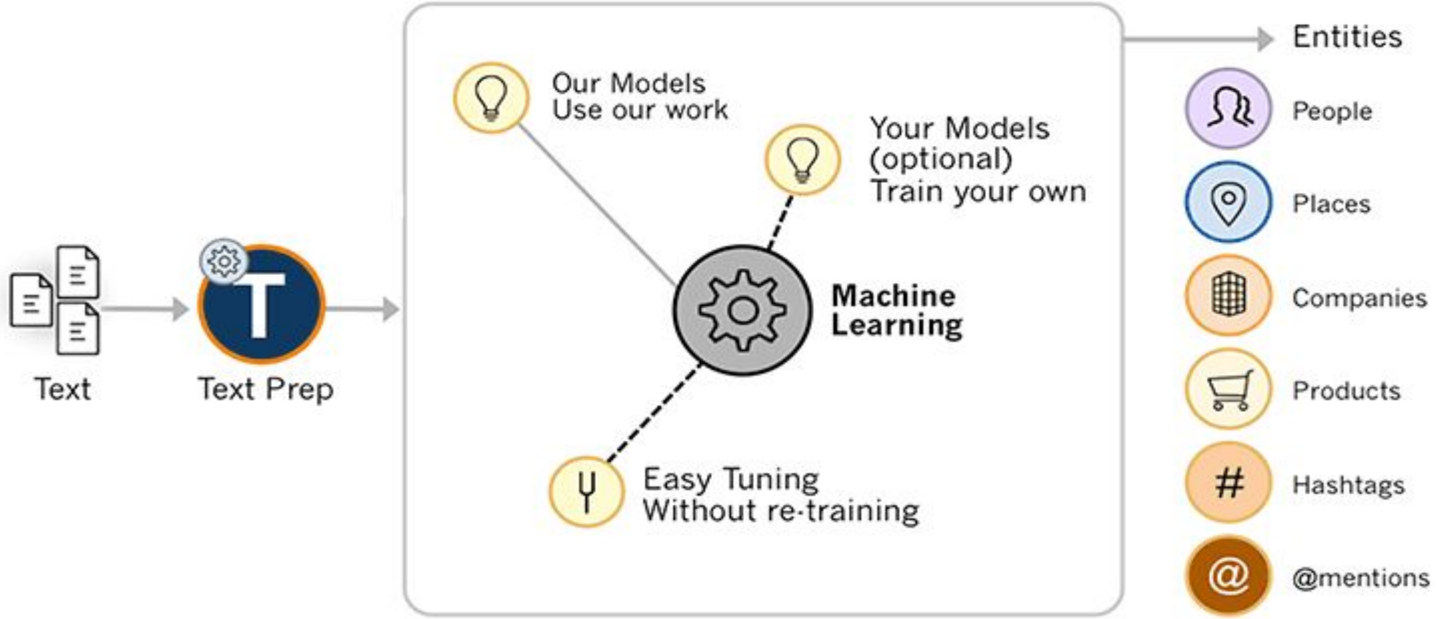
TASK 3: Scoring Linguistic Complexity

Checkpoint 1: Python package ‘textstat’ was used to calculate statistics from summary to determine readability,complexity and grade level of a particular corpus.

No changes are done to checkpoint 1 as satisfactory results were obtained with the previous submission.

Task 4: Named Entity Recognition (NER)

Method 2: Uses deep learning methods to custom train our model, which correctly recognizes entities.

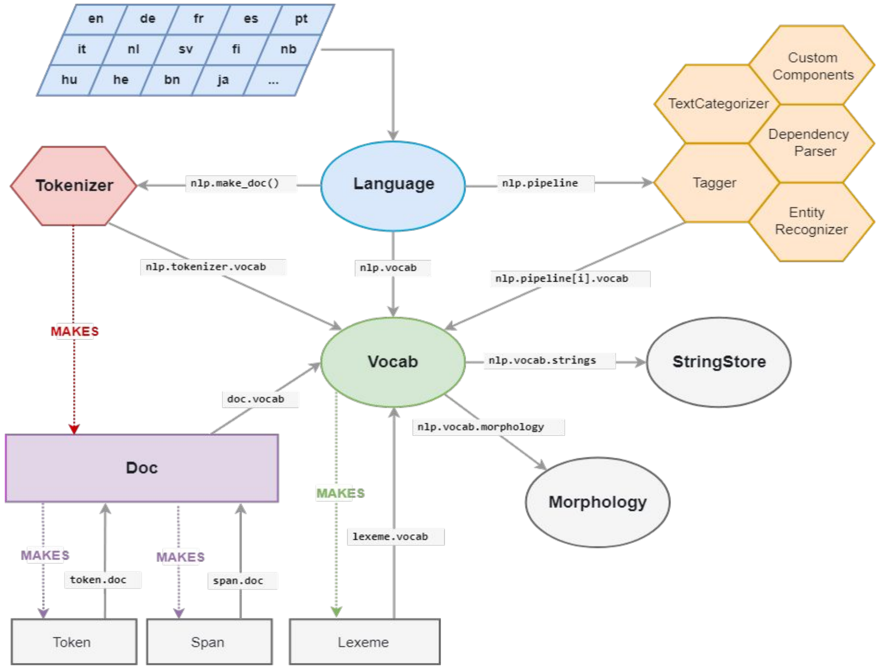


While for checkpoint 1: the model used was unsupervised.

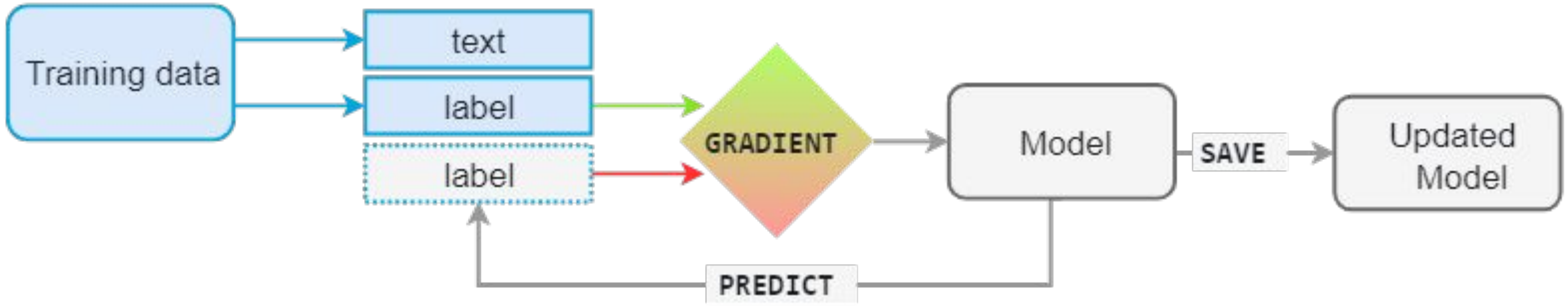
Method 2 is preferred as custom model for each company is more accurate and can be used on any transcript of a particular company.

Spacy’s sentencing parsing is the fastest and more accurate.

- the Doc object owns the data, and Span and Token are views that point into it. The Doc object is constructed by the Tokenizer, and then modified in place by the components of the pipeline. The Language object coordinates these components. It takes raw text and sends it through the pipeline, returning an annotated document. It orchestrates training and serialization.



- The model is then shown the unlabelled text and will make a prediction. Because we know the correct answer, we can give the model feedback on its prediction in the form of an error gradient of the loss function that calculates the difference between the training example and the expected output. The greater the difference, the more significant the gradient and the updates to our model.



Task 6: Clustering of Companies

Method : Used unsupervised machine learning model k-means clustering.

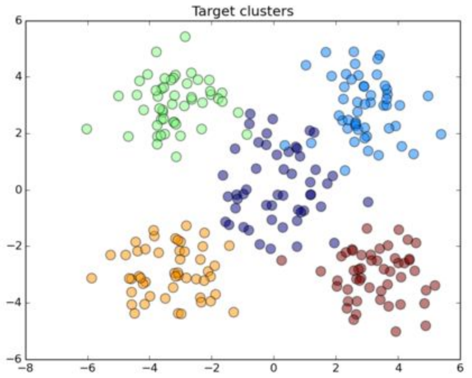
Input Parameters: Products

Since the number of companies in dataset are less, sentiment and linguistic score are not taken as parameters for clustering.

K-means clustering comes under partition method.

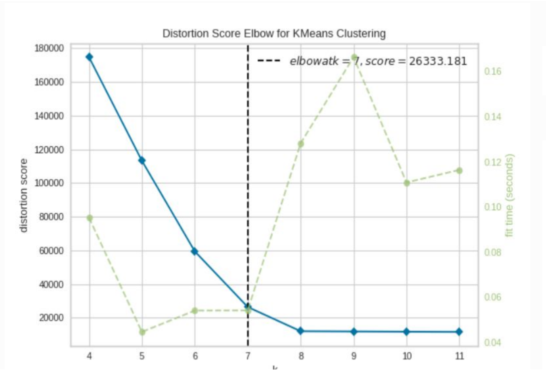
Partitioning Methods: These methods partition the object into k clusters and each partition forms one cluster. Distance parameter is used.

K-means: K-Means looks for a fixed number (k) of clusters in a dataset and identifies k number of centroids and allocates every data point to the nearest cluster. It is a unsupervised Machine Learning Algorithm.



How to find K in K-means?

Elbow Method: The “elbow” method select the optimal number of clusters by fitting the model with a range of values for K. If the line chart resembles an arm, then the “elbow”(the point of inflection on the curve) is a good indication that the underlying model fits best at that point.



X-axis : Number of clusters

Y-axis : Distortion Score