ZODIAC

Aryan Singh
Prashant Kumar
Kshitiz Srivastava
Vipul Bandi

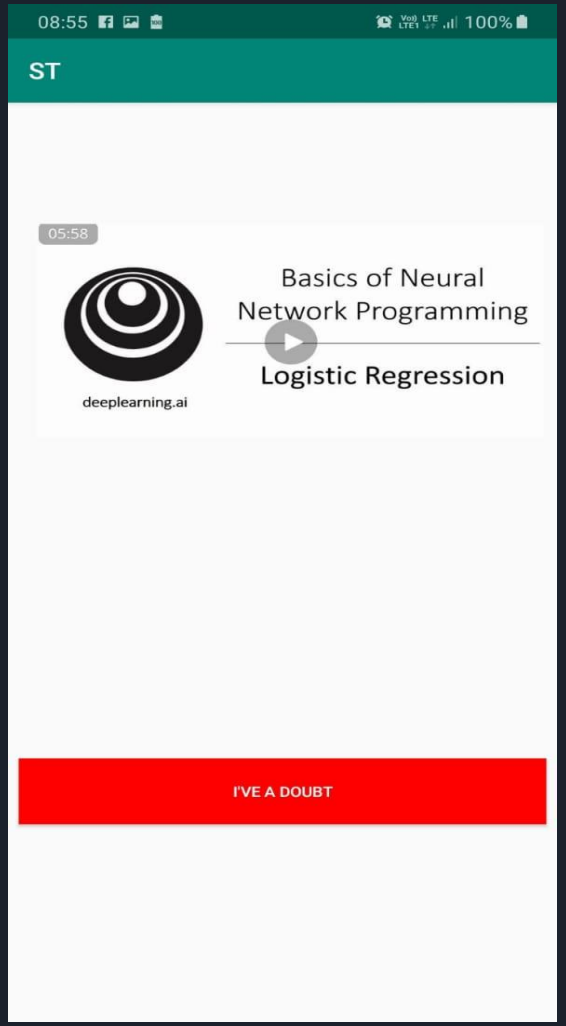
Problem Statement

Prepare an NLP model to convert speech to text for a Category/Tag name inputted by the user during a video being played and jump the video to the Suggested Part of the video where that specific tag is assigned to.

Our Approach

- Speech to text conversion done using Google API.
- Preprocessing the text using tokenization, lemmatization etc.
- Creating bag of words on dataset and running LDA using bag of words.

OUR WORKING APPLICATION

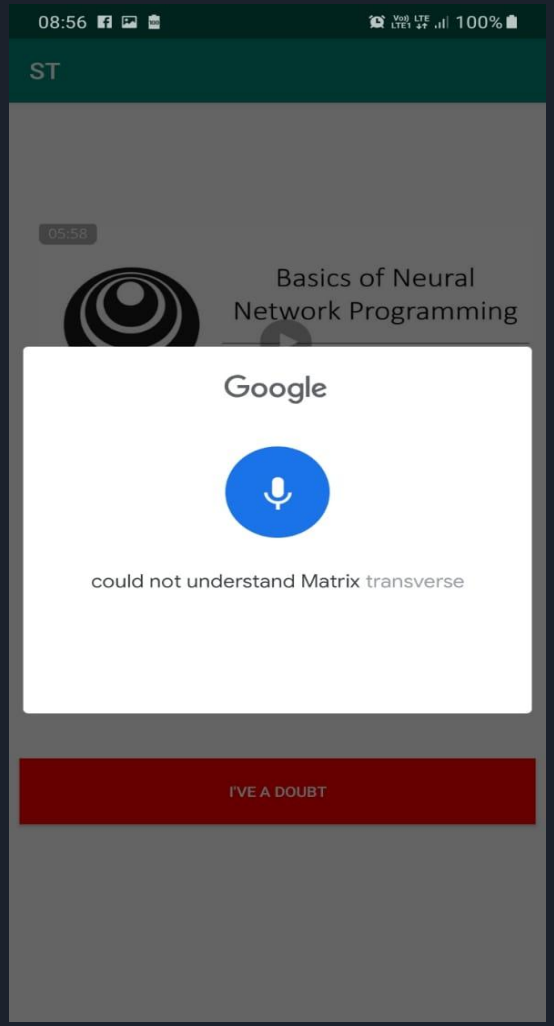


PURPOSE

Converting the query of user to text from audio and fed as input to the model.

Shifting the video to predicted moment.

TECH STACK USED: Android Studio, Firebase, Google API.



Data Preprocessing

Steps:

- Tokenization: Split text into sentences and sentences into words.
- Removal of stopwords.
- Lemmatization: Past, future to present tense.
- Stemming: Reduction of words to its root forms.

Packages Used:

Gensim(To extract symmentic topics from documnets), NLTK

Latent Dirichlet Allocation(LDA)

Underlying topics in text document

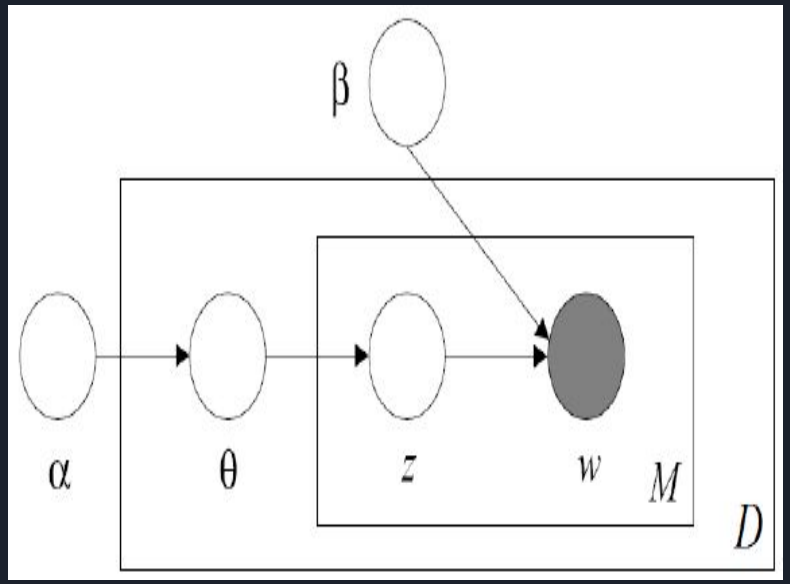
Keypoints:

- Documents are probability distributions over latent topics.
- Topics are probability distribution over words.

- We remove words appearing less than 15 times in a document and also those occurring in more than 10% of all documents.
- Create bag of words model for each document a dictionary reporting how many words and how many times those words appear.

  format==list of(token_id,token_count)

PLATE NOTATION



It is a concise way of visually representing the dependencies among model parameters.

- ‘Alpha’ is parameter for per document topic distributons.
- ‘Beta’ on per topic word distribution.
- ‘Theta-m’ is topic distribution for document-m.
- ‘Z-mn’ is topic for nth word in document m.
- ‘W-mn’ is specific word.

Generative Process

LDA assumes that new documents are created in following :

- Determine number of words in document.
- Choose a topic mixture for the document over a fix set of topics i.e. 20% topic A, 30% topic B, 50% topic C.
- Generate the words in document by:
  First pick topic based on document multinomial dist.
  Next pick a word based on the topic's multinomial dist.

In [42]:

```
# Data preprocessing step for the unseen document
bow_vector = dictionary.doc2bow(preprocess(unseen_document))

for index, score in sorted(lda_model[bow_vector], key=lambda tup: -1*tup[1]):
    print("Score: {}\t Topic: {}".format(score, lda_model.print_topic(index, 5)))
```

```
Score: 0.6758415102958679	Topic: 0.021*"file" + 0.019*"window" + 0.013*"program" + 0.008*"imag" + 0.007*"server"
Score: 0.29993242025375366	Topic: 0.015*"chip" + 0.011*"encrypt" + 0.008*"clipper" + 0.008*"card" + 0.006*"secur"
```

Advantages

- Less computational cost.
- The learning can proceed hierarchically from the observations into ever more abstract levels of representation.

THANK YOU