Natural Language Processing

J Component Project Report

Meeting Summarizer

Submitted By

Henil Jayesh Thakor (18BCE0934)

Mrinmay Date (18BIS0147)

Harshith Chukka (18BCE0941)

Under the guidance of

Prof. Sharmila Banu K

Associate Professor Grade 1

School of Computing Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Fall Semester 2020-21

PROBLEM STATEMENT

Since the beginning of the Industrial revolution meetings have always been playing a major role in making key decisions for a small organisation to giant multinational corporates. These organisations hire skilled Minute-Takers for noting down and summarizing the key points discussed during the meeting. An automated meeting summarizer can play a key role in achieving the same task. It can record live meeting chats and process it to derive key points of the meetings and finally summarize them.

This project reads the meeting chat as input, searches for key statements using Text Rank algorithm and finally prints the meeting chat summary as output. The project also takes user's feedback on the generated summary and improves the quality of the summary for the next meeting chat based on the user's feedback.

PROJECT PIPELINE

The various steps included in the project pipeline are:

1. Reading Meeting Chat
2. Pre-Processing Text
3. Word Tokenization
4. Word Lemmatization
5. Generating Word Frequency Vector
6. Sentence Tokenization
7. Sentence Ranking
8. Finding Threshold Rank
9. Generating Summary
10. Taking User's Feedback
11. Updating Feedback Table

MODULE DESCRIPTION

The modules included in the project are:

1. **Reading Meeting Chat:**
   The module handles the task of reading meeting chat text as and input and pre-process it by removing the names of the speakers.

2. **Text Ranking:**
   This module contains the following submodules:

   2.1. **Forming Word Frequency Vector**
   The module tokenizes the meeting text, lemmatizes words and forms word frequency table.

   2.2. **Sentence Ranking**
   The submodule weights sentences based on the word frequency table and generates sentence ranks based on the sentence weight and user's feedback table.

3. **Generating Summary:**
   This module computes the average sentence ranks, sets threshold rank as *1.2*average_rank* and finally prints the sentences with rank higher than the threshold rank in the summary.

4. **Feedback module:**
   This module accepts key statements as user's feedback, tokenizes words, lemmatizes words and forms feedback table containing the word frequencies representing importance of a word for the user. This feedback table is then used to update the sentence ranks while generating summary for the next meeting chat.

ERRORS AND ITS SOLUTIONS

The following were the errors found in the traditional text rank algorithm used for meeting summarizer and its corresponding proposed solutions in the project:

1. Different forms of the same words like *"working"* and *"work"* where considered to be different while counting word frequency.

   Solution: The above error was resolved by performing word lemmatization for extracting the stem word and computing the word frequency table.

2. Important sentences like the sentences conveying the main objective or the purpose of the meeting were not included in the summary, because of low sentence rank.

   Solution: The above error was resolved by taking key statements in the meeting chat as user's feedback and computing the feedback table representing importance of a word for the user. The feedback table is then used to increase sentence rank of important sentences during the next meeting summarization.

3. The summaries fail in conveying the exact conclusion of the meeting when read by a third person, because pronouns *(like he, she, I, you)* do not link to the person it refers to.

   Solution: The above error can be resolved by replacing all the personal pronouns with corresponding nouns before pre-processing of the meeting chat.

BIBLIOGRAPHY

1. *"Creating Generic Text Summaries"* by Yihong Gong and Xin Liu (2001) [*Link: 10.1109/ICDAR.2001.953917*]
2. *https://en.wikipedia.org/wiki/Automatic_summarization*
3. *https://medium.com/analytics-vidhya/simple-text-summarization-using-nltk-eedc36ebaaf8*
4. *https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/*
5. *https://www.geeksforgeeks.org/python-lemmatization-with-nltk/*

APPENDIX

1. **Project Notebook Link:**
   https://colab.research.google.com/drive/1bBhW5UgMzuLuH4tPj89YsLNLL7czi-XC?usp=sharing

2. **Nat-geo task and Hands-on notebook link:**
   - Henil Jayesh Thakor(18BCE0934): *https://github.com/henilthakor/NLP-Tasks*
   - Harshith Chukka(18BCE0941): *https://github.com/harshithChukka/NLP-FALL-2020-21*