MOVIE REVIEW SENTIMENT ANALYSIS INTEGRATED WITH DIALOGFLOW

Final Project

uOttawa

Faculté de génie
Faculty of Engineering

Ahmed Hallaba **Bassel Hamshary**
Kamel El-Sehly

JULY 31, 2021

UNIVERSITY OF OTTAWA
Ottawa, Canada

Table of Contents

List of Tables

1. Objectives

This project aims to classify the sentiment analysis of the movie reviews whether positive or negative, integrated through the user interface of a dialog-flow chatbot.

2. Introduction

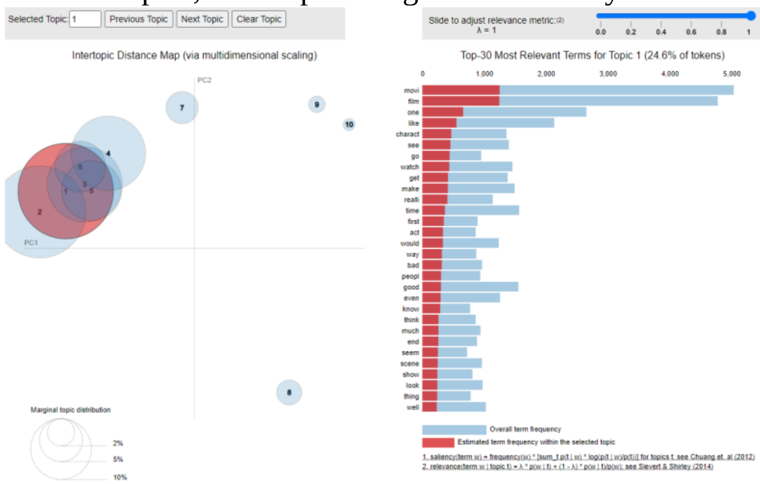
Sentiment analysis is a machine learning technique to identify the sentiment of the sentence through NLP, we will use this technique in the movies' reviews, we train our model from IMDB-Dataset having 50K movie reviews for natural language processing or Text analytics, this is a dataset for binary sentiment classification.
to learn from the data we have to do feature engineering to clean our data and determine the words that affect the sentiment, then we fed the machine learning classification and clustering algorithm to produce a model that will be able to analyze the sentiment from future input texts, and finally, deploy our champion model on an application using dialog-flow chatbot as a user interface to interact with easily.

3. Feature Engineering

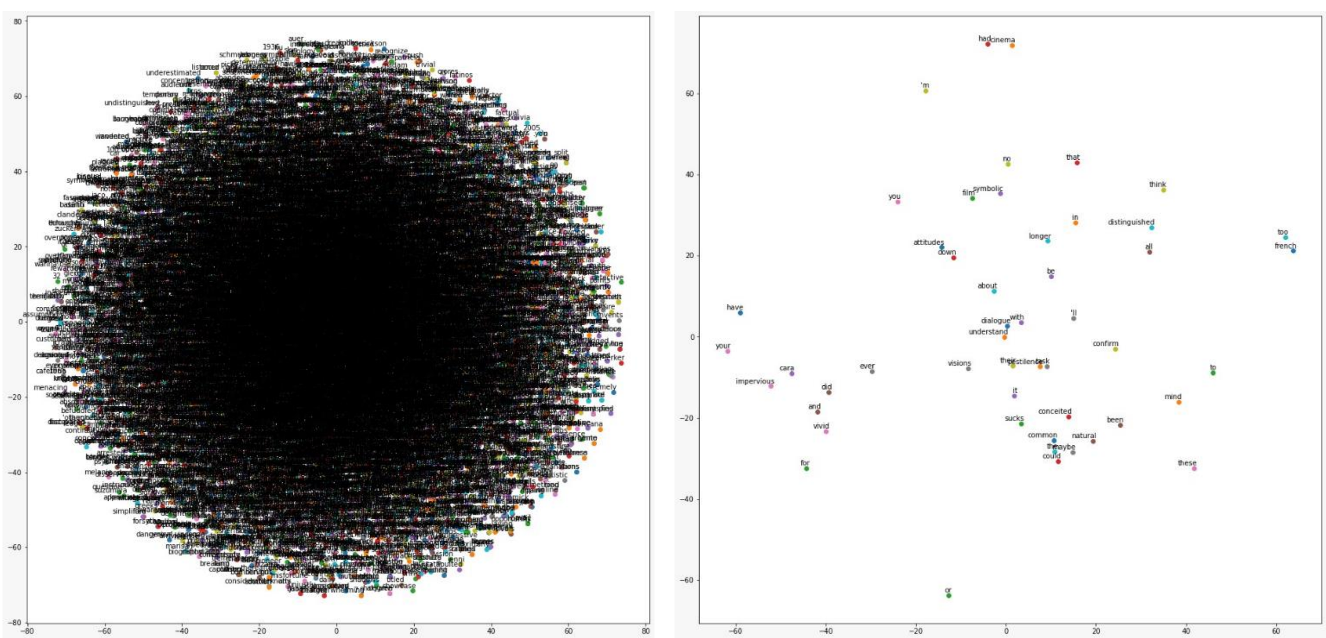
BOW: Bags of Words: This is a representation of text that describes the occurrence of words within a document. This is the most popular technique used to convert categorical features to numerical ones.

TF-IDF: Term Frequency-Inverse Document Frequency: this is another way or technique used for occurrence of words. TF-IDF measures relevance, not frequency. The TF part divides the number of occurrences of each word in the document by the total number of words in the document, while the IDF does the downscaling of weight for words that occur in many documents in the corpus. For example, if the words like 'the', 'and' appears in all documents, those will be systematically discounted. Each word's TF-IDF relevance is a normalized data format that also adds up to one.

LDA: Latent Dirichlet Allocation: It is a form of unsupervised learning that views documents as bags of words. define it as a generative probabilistic model of a corpus with the idea of representing each document as a random mixture over latent topics, each topic being characterize by a distribution over words.



Paragraph Vector (Doc2Vec) is supposed to be an extension to Word2Vec such that Word2Vec learns to project words into a latent d-dimensional space whereas Doc2Vec aims at learning how to project a document into a latent d-dimensional space.



BERT: Bidirectional Encoder Representations from Transformers is an open-source machine learning framework for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. The BERT framework was pre-trained using text from Wikipedia and can be fine-tuned with question-and-answer datasets. It is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection.

4. Classification Models

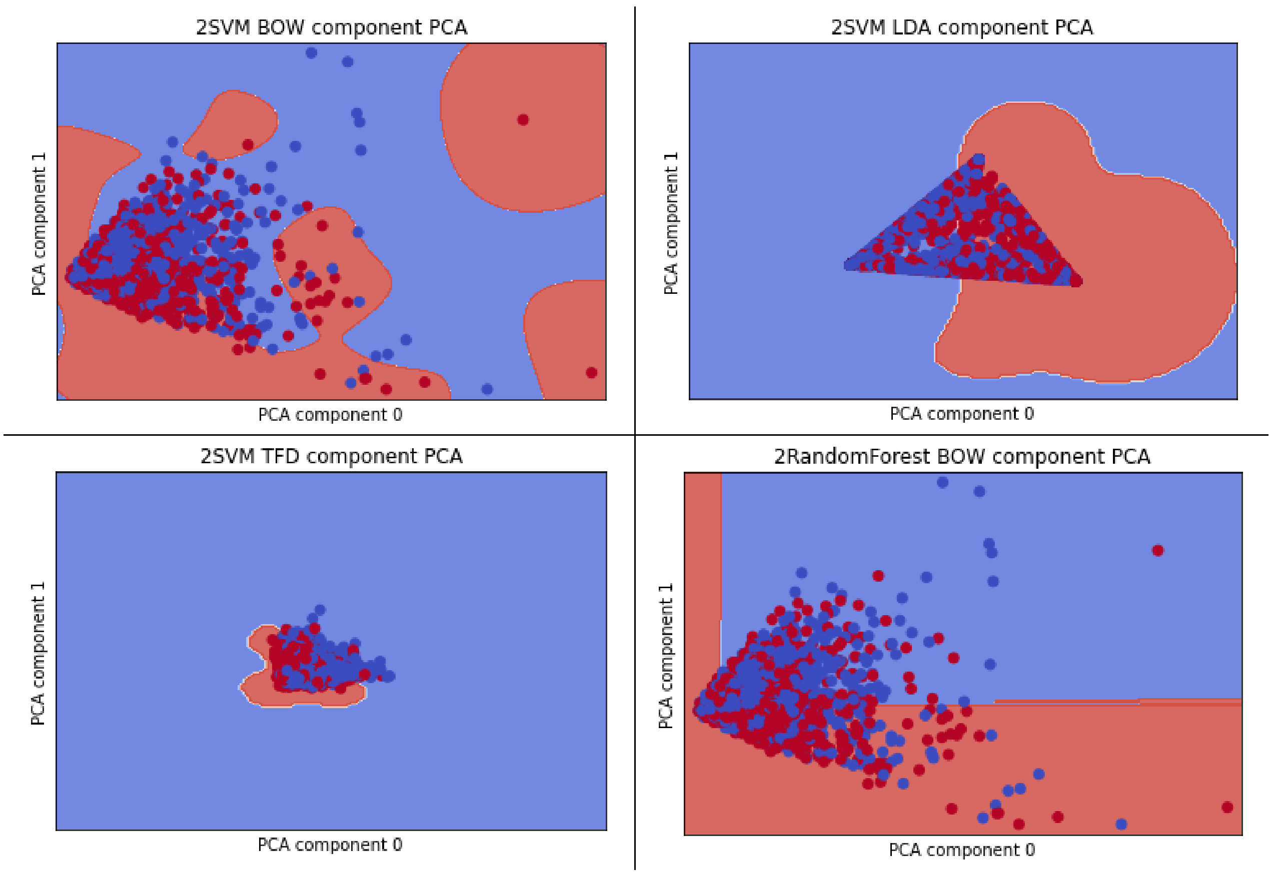
We used in this application five different classification algorithm to classify the sentiment which are: Gaussian NB, Bernoulli NB, Decision tree, SVM, and Random Forest.

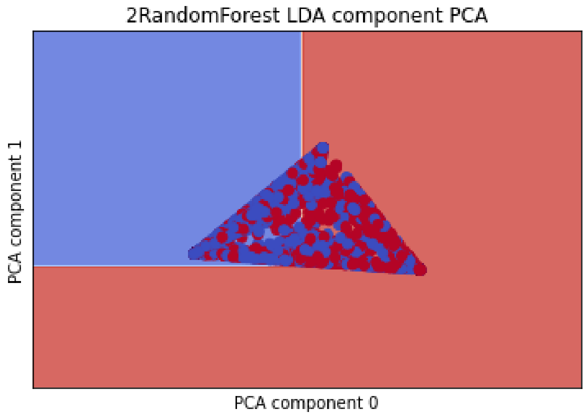
Table 1: Accuracy of classification models with different feature engineering techniques

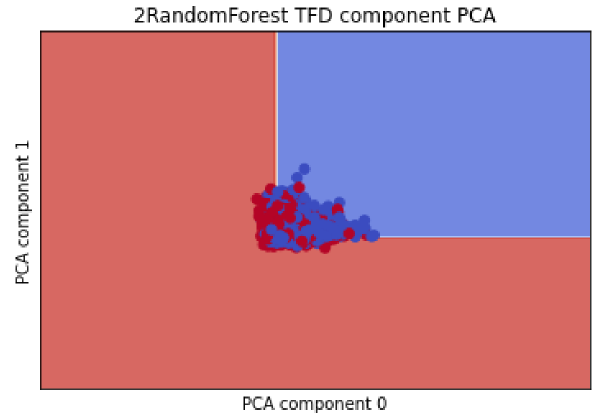
	BOW	TF-IDF	LDA	BERT	Doc2Vec
Gaussian NB	0.592	0.602	0.692	0.756	0.824
Bernoulli NB	0.832	0.832	0.53	0.76	0.8266
Decision Tree	0.698	0.696	0.582	0.668	0.8013
Random Forest	0.8	0.778	0.71	0.772	0.82
SVM	0.852	0.866	0.71	0.82	0.828

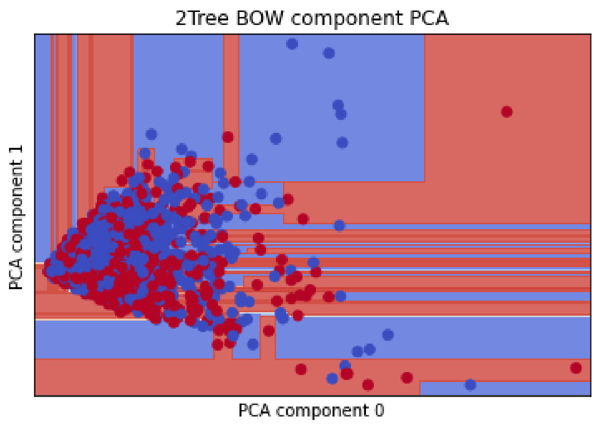
From the table we can deduce that our champion classification model is the SVM using TF-IDF with accuracy of 86.6 %, and this model can be deployed to our chatbot application.

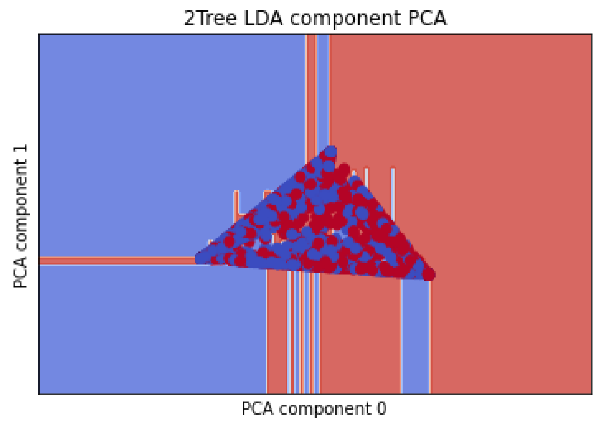
- ***Some Visualizations with Feature Engineering Algorithms***

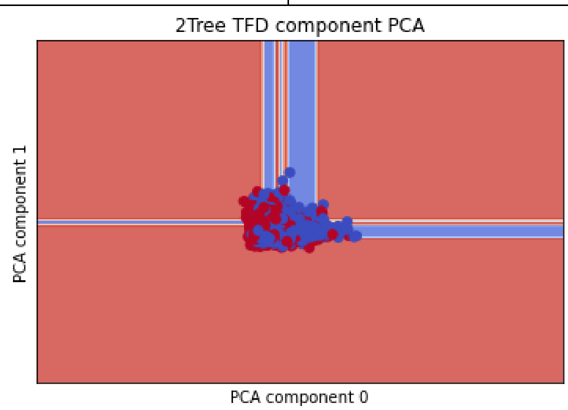



2RandomForest LDA component PCA
PCA component 1
PCA component 0


2RandomForest TFD component PCA
PCA component 1
PCA component 0


2Tree BOW component PCA
PCA component 1
PCA component 0

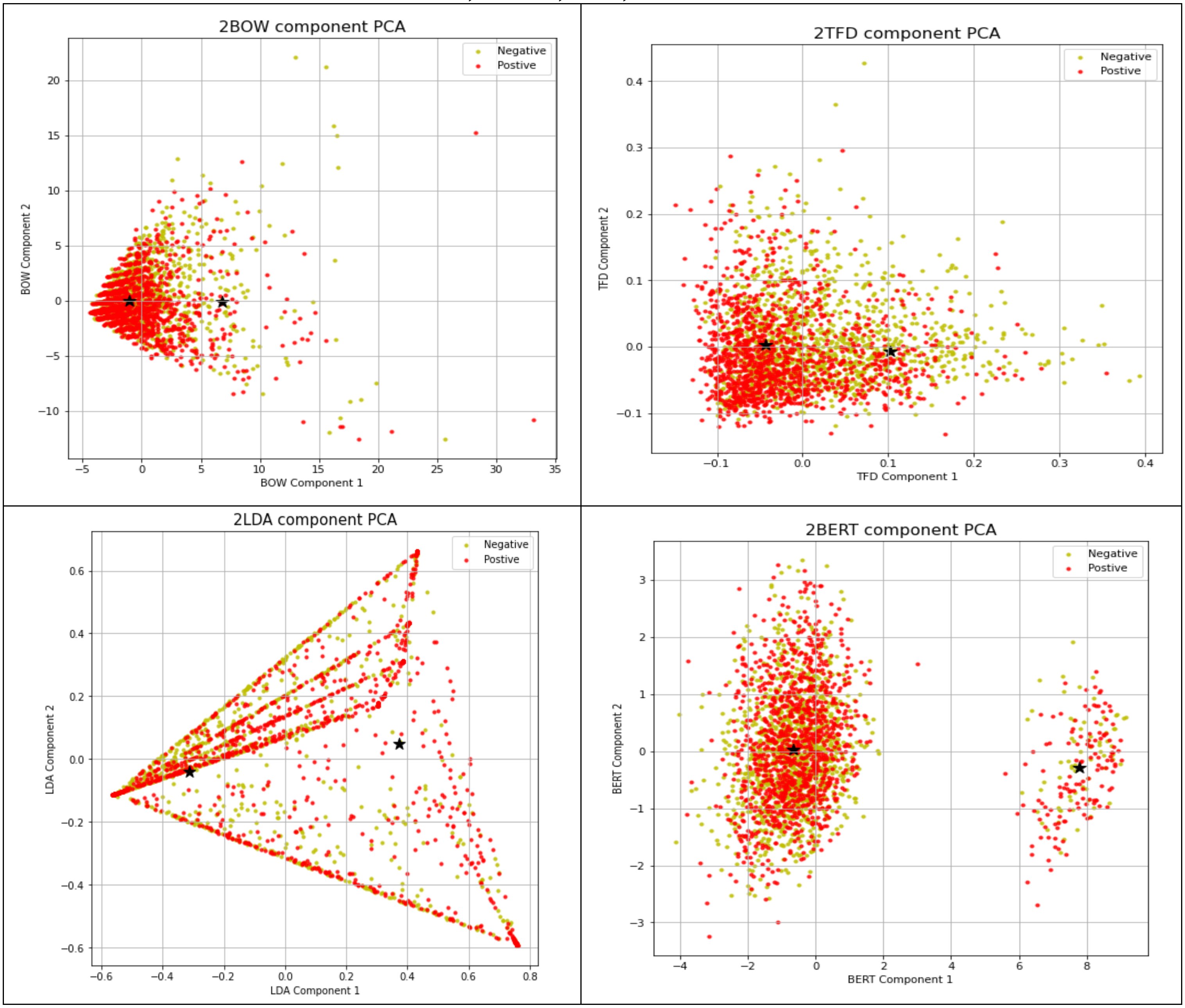

2Tree LDA component PCA
PCA component 1
PCA component 0


2Tree TFD component PCA
PCA component 1
PCA component 0

5. Clustering Models

We used two different clustering algorithms which are the K-Means and the Agglomerative Clustering.

- Performance of K-Means with BOW, TF-IDF, LDA, BERT.



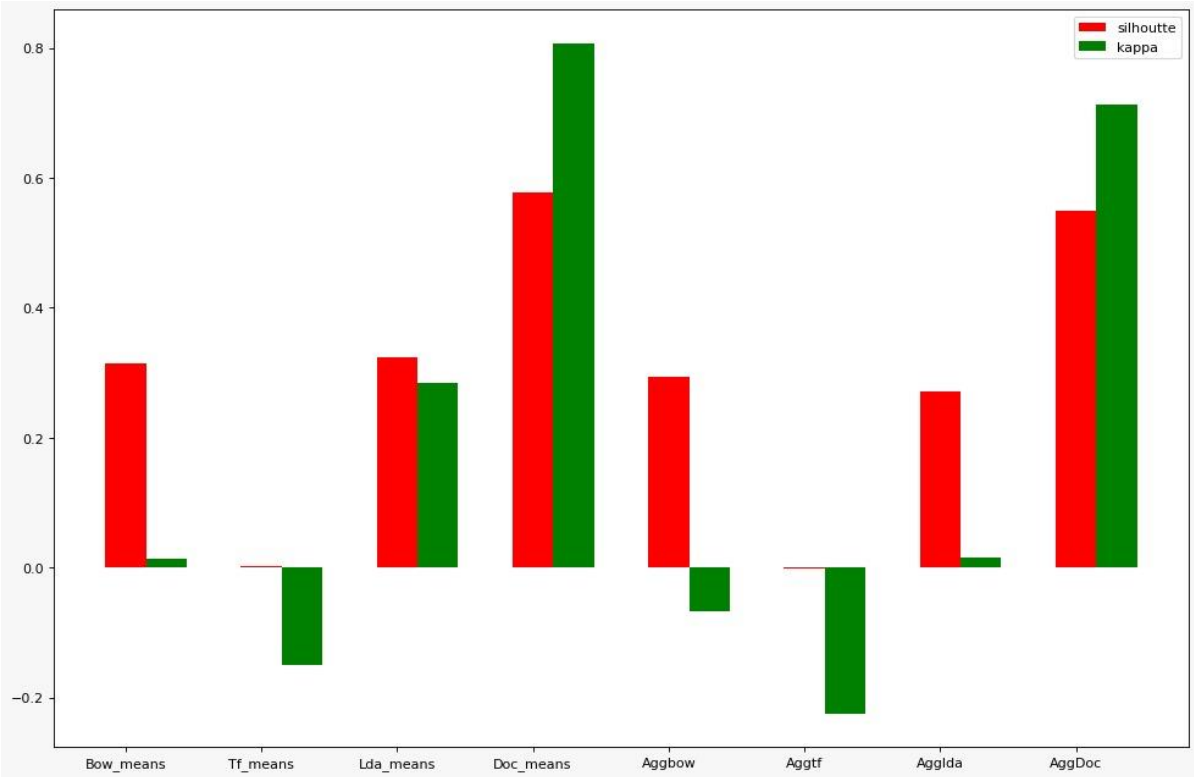
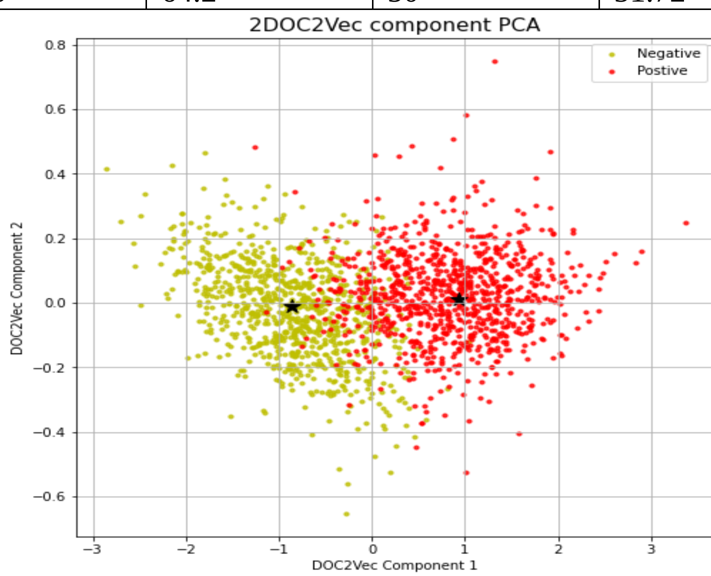


Figure 1: Silhouette and Kappa for clustering models with different feature engineering techniques

Table 2: Accuracy of the clustering models with different feature engineering techniques

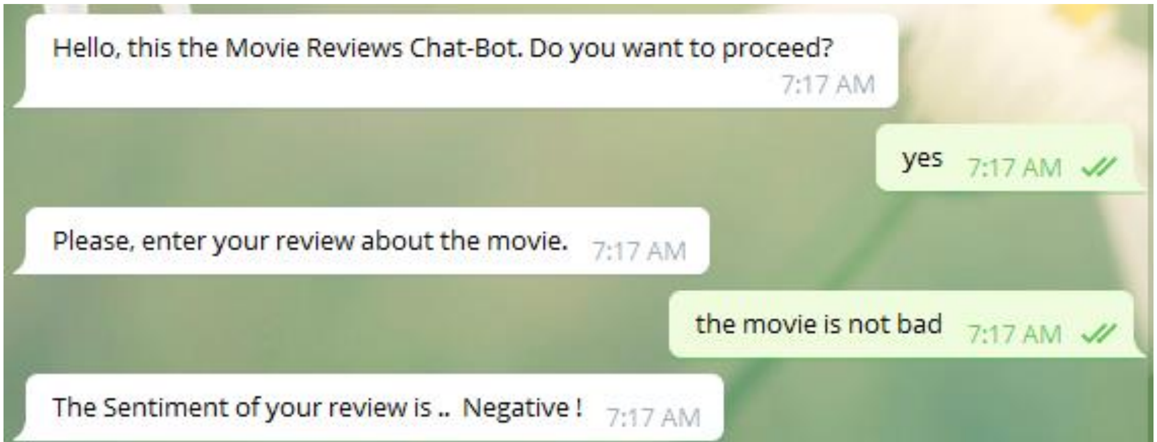
	BOW	TFD	LDA	BERT	Doc2Vec
K-Means	57.64	65.64	50.36	51.72	80.2
A-Clustering	54.56	64.2	50	51.72	70.8



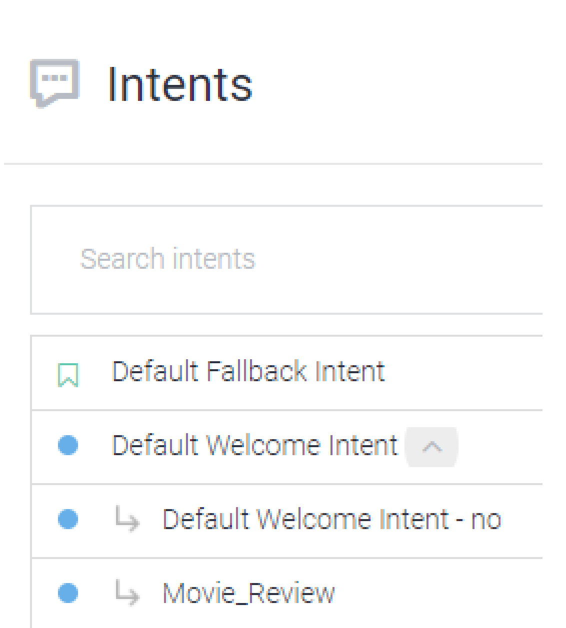
From the table and the figures above, we can deduce that our champion Clustering model is the K-Means using Doc2Vec with accuracy 80.2 %, Silhouette of 0.6 and Kappa of 0.8, and this model can be deployed to our chatbot application.
As we see, we have two models with high accuracy, we have the opportunity to choose which technique to run our application. (Both models are extracted and can be inserted in the application one at a time)

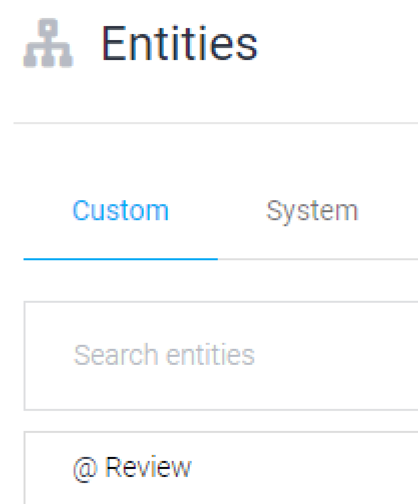
6. Chatbot

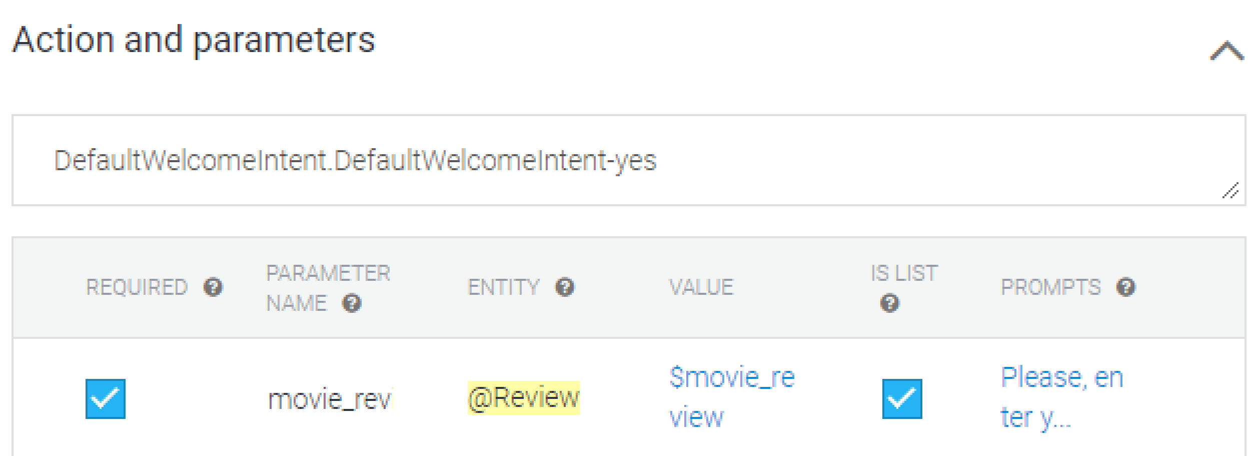
We managed to deploy our machine learning models (SVM using TF-IDF or K-Means using Doc2Vec) to a user interface using dialog-flow integrating with Telegram.
The chatbot asks the user to enter his movie review, in the background the chatbot integrates with an application that fed the user's review to the model and return the sentiment of the review back to the user on the chatbot page, as shown in the below figures.

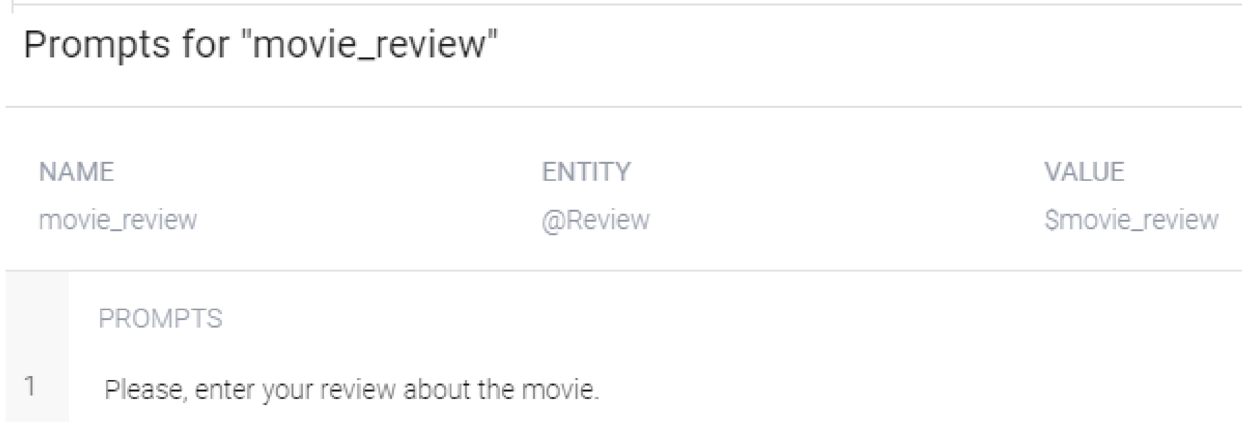


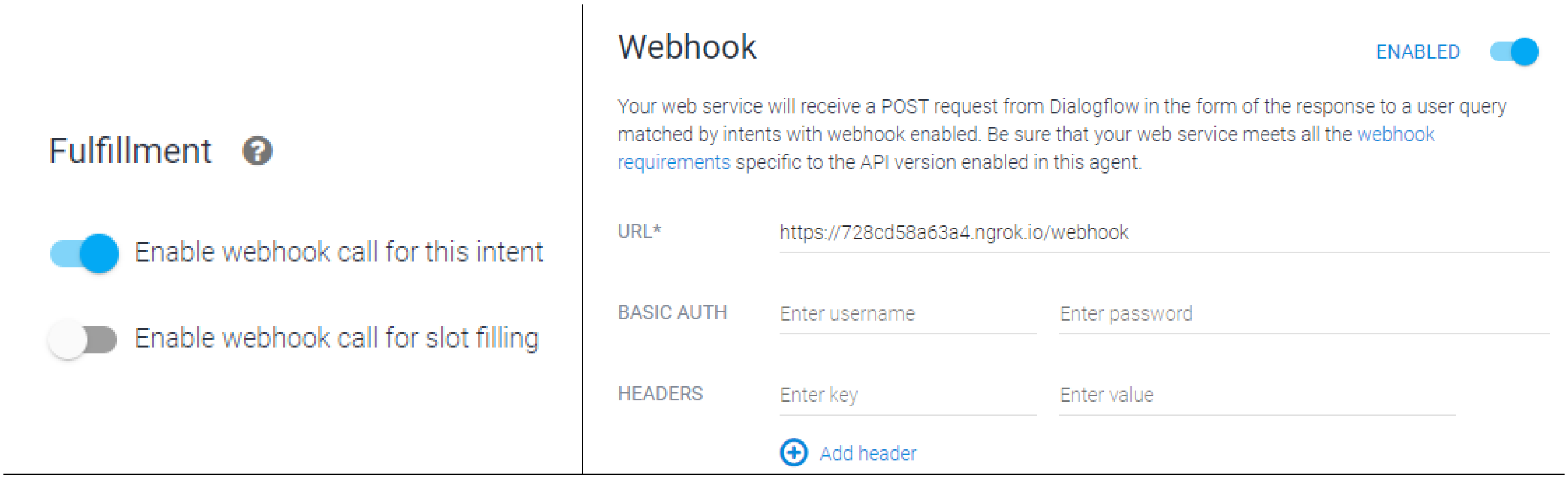
6.1. Background Process of the Chatbot

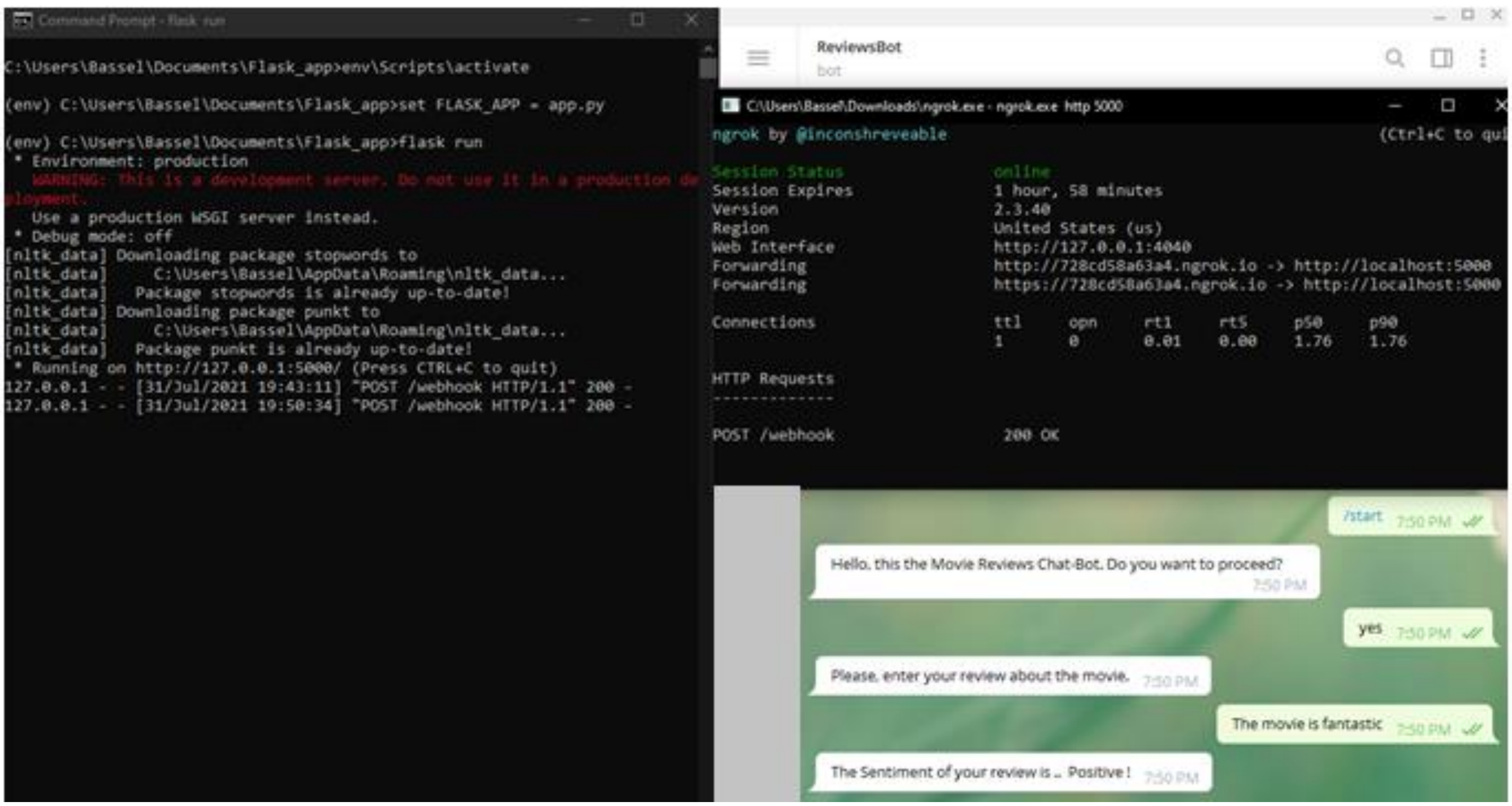

Intents
Search intents
Default Fallback Intent
Default Welcome Intent
Default Welcome Intent - no
Movie_Review


Entities
Custom
System
Search entities
@ Review


Action and parameters
DefaultWelcomeIntent.DefaultWelcomeIntent-yes
REQUIRED
PARAMETER NAME
ENTITY
VALUE
IS LIST
PROMPTS
movie_rev
@Review
$movie_review
Please, enter y...


Prompts for "movie_review"
NAME
movie_review
ENTITY
@Review
VALUE
$movie_review
PROMPTS
1
Please, enter your review about the movie.





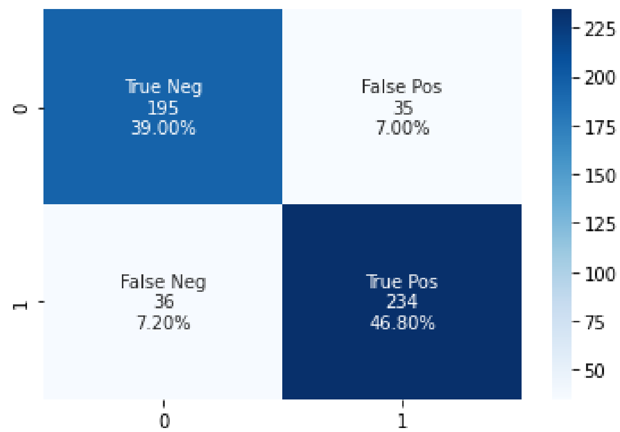
The bot simply starts by asking the user if he want to proceed, if yes the bot will ask the user to enter the review, when the user enter the review, the bot will send a fulfilment request to the webhook application that is integrated with the machine learning model for sentiment analysis, the application will analyze the review and send back a fulfilment response to the chatbot with either a positive or negative response.

7. Error Analysis

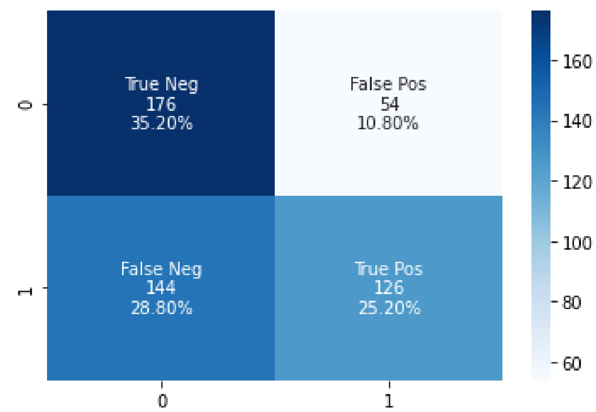
Let’s take a look at error analysis on 3 different levels – Predictions, Data, and Features.

7.1. Predications

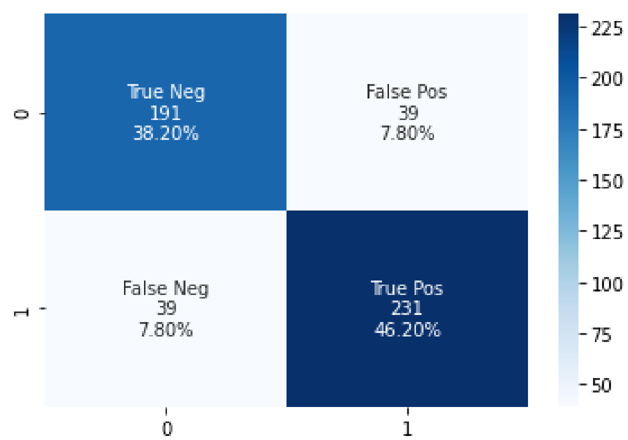
So, let’s have a look on the confusion matrix to see if we can get sense of any patterns in the inaccuracies to be able to analyze it.
We will focus on the SVM model as it’s the champion model with the various feature engineering methodologies.



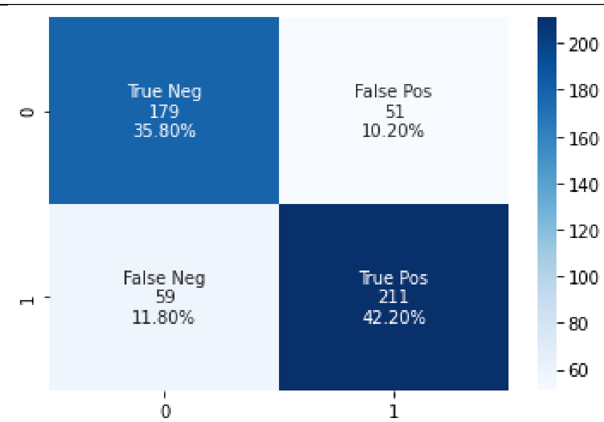
SVM with TF-IDF



SVM with LDA

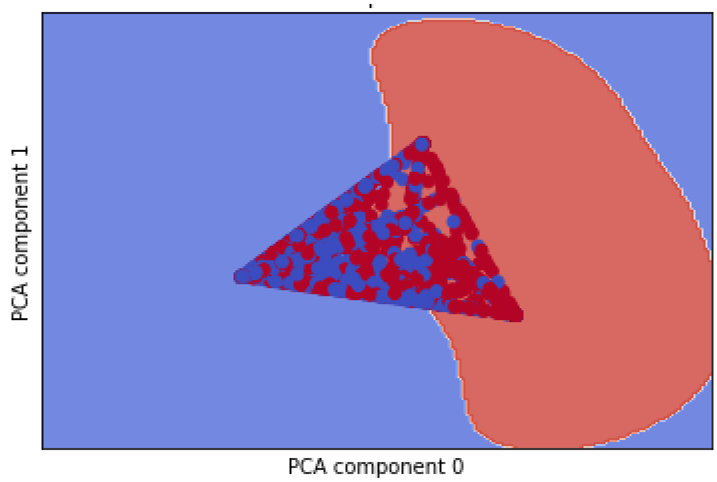


SVM with BOW



SVM with BERT

Starting with the LDA, obviously it has missed with about 40% of the sentiment classifications, for deeper analysis, the reason is that the LDA classifies according to topics meanwhile almost all our reviews are about the same topic, which is movies, that’s why there is obvious miss classification here.



And this figure of SVM with LDA shows how restricted it is. The other methods (including TF-IDF (the champion method)) have done fairly well with almost no majority for a misclassified class (there is no imbalance), But of course there may be other reasons.

7.2. Data

Actually, we faced a problem with data, we couldn't apply all of our dataset (just 5% of the total dataset could be used) because of the limited processing resources.

<u>Why could this make a problem?</u>

As we know that data is the fuel of machine learning so the more you feed the model the more accuracy your model will get (of course to a particular limit not to fall in over fitting)

7.3. Features

Our features are represented in the reviews itself so let's get deeper look on them.

<table>
<tr>
<td>
</td>
<td>Positive sentiments
Word Cloud</td>
</tr>
</table>

	Negative sentiments Word Cloud

From the first look at the Word Clouds of both Positive and negative classes we can observe the huge overlapping between them.

<u>Why could it make a problem?</u>

Words are the building brick of the sentence on which the classifiers will make their classifications, so this overlapping is a core problem which causes a lot of misclassifications.

8. Conclusion & Future Work

To sum up what was done in this project, the TF-IDF algorithm with the SVM classifier was found to be the best classification model with accuracy 82.6 %, also the Doc2Vec algorithm with the K-Means is the best clustering model with accuracy 80.2 %, after determining the best models we go through the application that can integrate the chatbot with the ML models, the SVM using TF-IDF is the model used in the deployment as it has the highest testing accuracy.

This project is just a first step in the movie reviews sentiment analysis application, it could be improved by adding many features like the movie name, movie type, and movie rating. These extra features could be implemented in the future that will help movies platforms to easily reach the spectators and can get their review as fast and genuine as possible. This approach could help in understand the sense of the audience, which will help in developing more efficient recommender systems.