Semi Final — prototype

Hackpions GDS ES Hackathon (edition 3)

EY
Building a better working world

Team: Madras_Sharks

1. Aniruddha Kawade
2. Anuj Sindgi
3. Varad Joshi

Theme:

NLP based tagging solution

Problem Statement

- To assist the Intelligent Automation team process large amounts of exception data
- Achieve the above by classifying exceptions relevant to specific teams, i.e. Business Exception & System Exception
- Create a self-learning model to recognize patterns from input features, tag exceptions as they come and generate an output text file, with the respective exception tags

Solution(Initial Approach)

Case 1: Keywords for Exceptions are provided. (Rule-Based approach)

- This case is relatively easy to implement. We preprocess the given input string, i.e removing special characters and common stopwords in English language using nltk library and regex.
- In this approach, we simply iterate through the keywords for Business and System Exception and when particular keywords are found we return the corresponding Exception.
- However, due to certain overlap between words in keywords for exceptions, certain inputs were misclassified.
- Not highly scalable as manual extraction for keywords is time consuming on a large scale

Solution(Initial Approach)

Case 2: Keywords for Exceptions are not provided. (ML-based approach)

- First step is same as before. We preprocess the input data.
- We used CountVectorizer from sklearn library, which converts a text string into an array of token counts.
- We trained models using Support Vector Machine, Naive Bayes & Logistic Regression on the transformed array for prediction.
- We achieved an accuracy score of 85%. This is a reasonable score given that the training & test data sizes were only 40 & 7 respectively.
- This approach can be easier to scale for a larger training dataset.

Solution(Improved Approach)

Case : Keywords for Exceptions are not provided. (DL-based approach)

- First step is same as before. We preprocess the input data.
- We used Word2Vec from gensim library, which converts a text string into a vector representation.
- We use tensorflow to build a sequence model with two bidirectional and one dense layer, containing more than 1,00,000 parameters.
- We achieved an accuracy score of 70%. This is a reasonable score given that the training & test data sizes were only 37 & 10 respectively.
- This approach can be easier to scale for a larger training dataset. And it converges better with larger dataset.

Methodology

Vectorization:

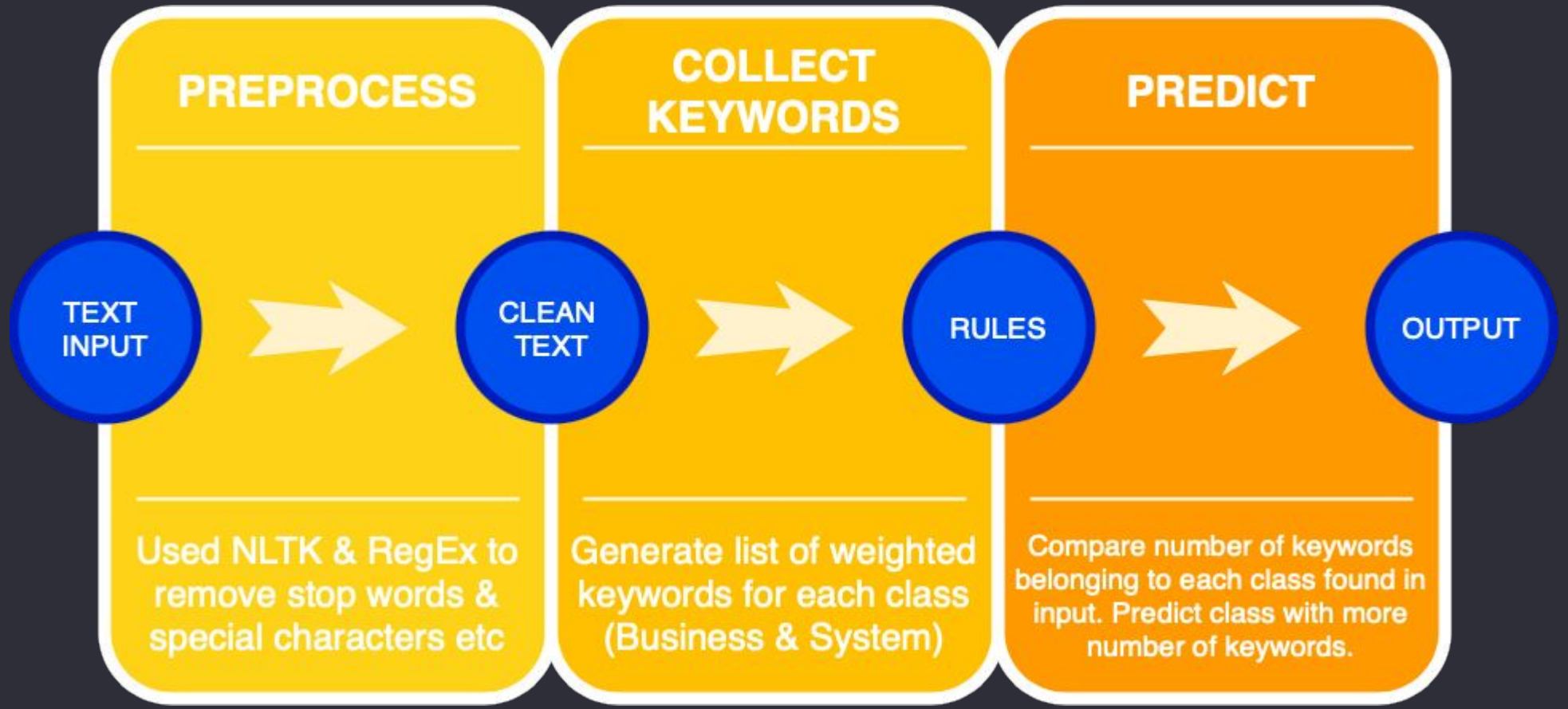
- Word2Vec maps an input word to a n-Dimensional space,allowing the data to retain semantic meaning and providing better context than TF-IDF and Bag of Words.
- Word2Vec also converts the input into vector of smaller size compared to TF-IDF/BOW decreasing the number of computations.

Model Training:

- With our input in the form of an n-dimensional, we can train our data using neural network.
- In training, our model learns which tokens correspond to which exception by going through the training set several times.
- The model can assess tokens present in test input & make predictions accordingly.

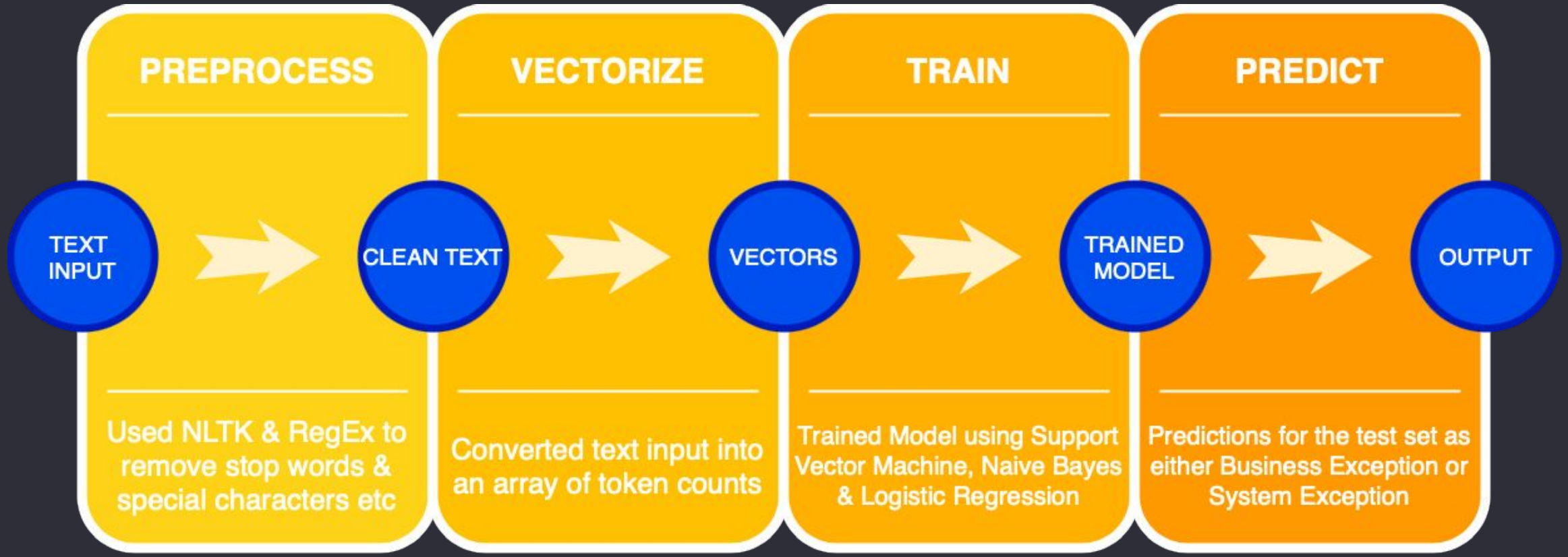
System Architecture Proposal(Initial Approach)

Case 1: Rule-based Method



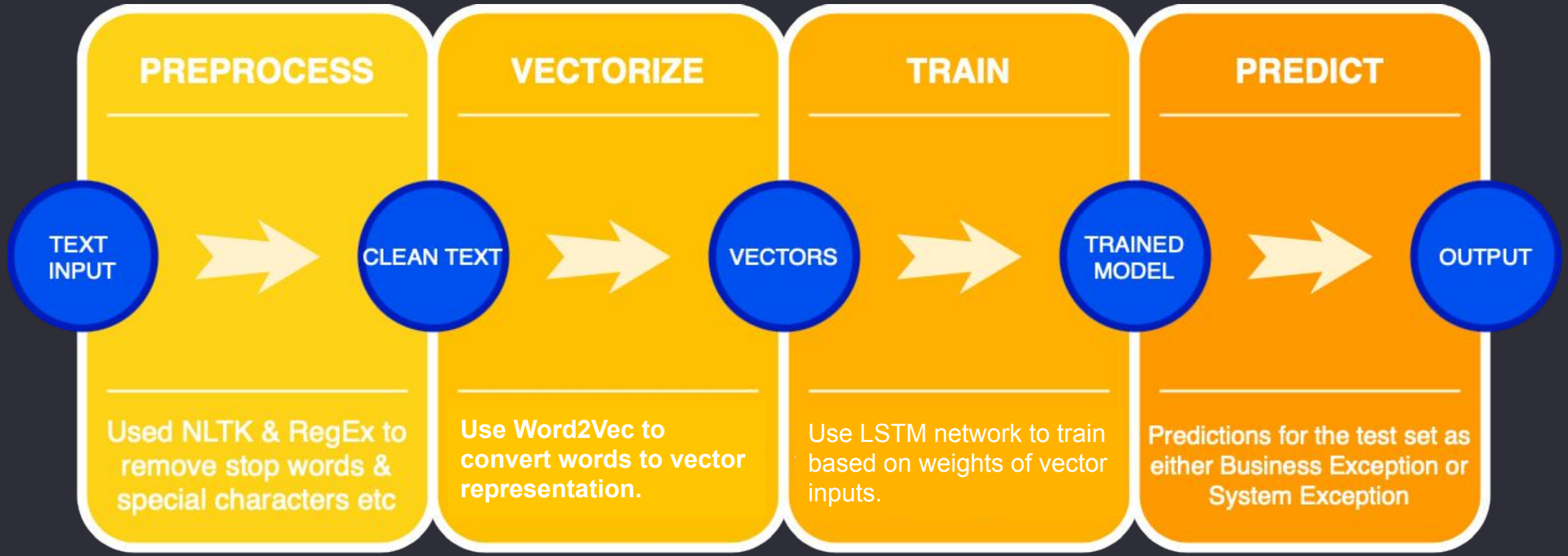
System Architecture Proposal(Initial Approach)

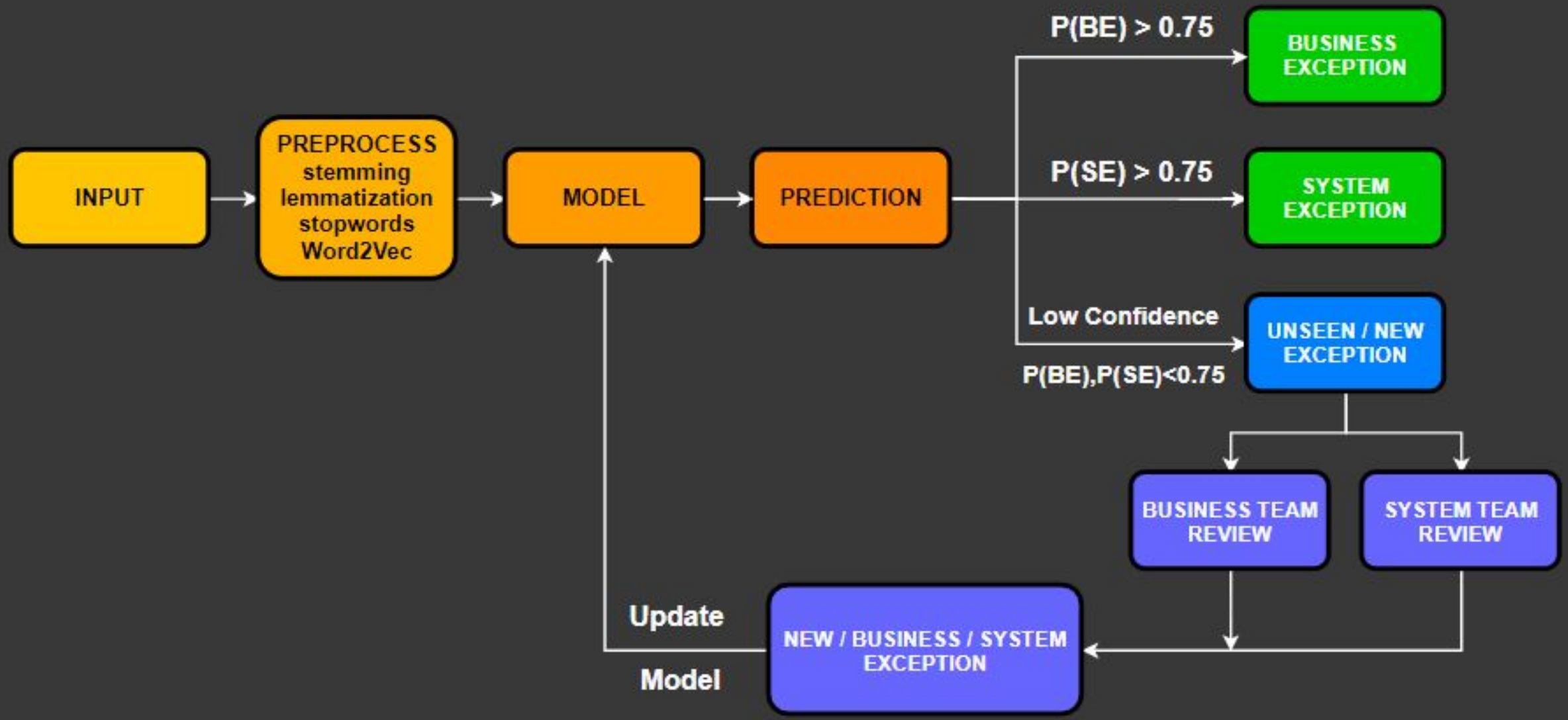
Case 2: ML Based Method



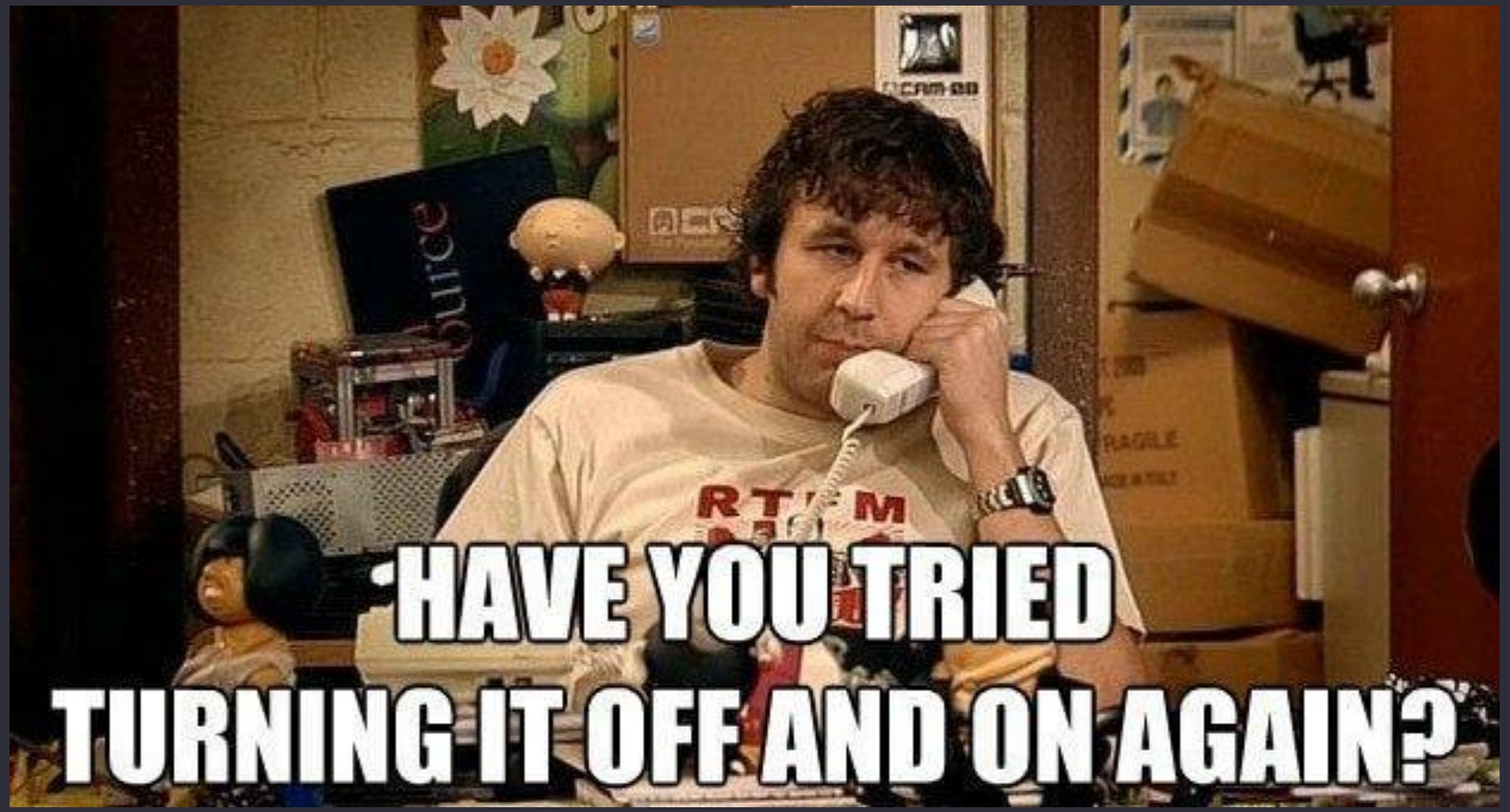
System Architecture Proposal(New Approach)

Case 3: DL Based Method




INPUT
PREPROCESS
stemming
lemmatization
stopwords
Word2Vec
MODEL
PREDICTION
P(BE) > 0.75
BUSINESS EXCEPTION
P(SE) > 0.75
SYSTEM EXCEPTION
Low Confidence
P(BE),P(SE)<0.75
UNSEEN / NEW EXCEPTION
BUSINESS TEAM REVIEW
SYSTEM TEAM REVIEW
NEW / BUSINESS / SYSTEM EXCEPTION
Update
Model

Solution Prediction Proposal -



Solution Prediction Proposal -

- Assuming we have historical data of input exception and their corresponding solutions, we can build a model to predict solutions for future incoming exceptions. Our hypothesis is that, the exceptions will occur in clusters.

- The DL model predicts the type of Exception, we also plan to predict a plausible solution for the problem based on past recommendations by the Business and System Teams.

- We can perform Agglomerative Clustering on the training data based on the prominent keywords in the exceptions.

- When we get an exception as an input, we can find the cluster this exception belongs to and recommend the 3 most frequent solutions in this cluster.

Results

	Case 1 (Rule-based)	Case 2 (ML based)	Case 3 (DL based)
Test Set Size	47	7	10
Categories	Business/System	Business/System	Business/System
Classification Accuracy	93.61%	85.71%	70%

Conclusion:

- Rule based method works well for smaller dataset & if keywords are known apriori
- ML based methods works better for larger dataset & accuracy too increases with dataset size. This solution also has the advantage of scalability.
- DL based approach has the prerequisite of large dataset and in theory would provide better result than Rule based and TF-IDF.

Attachments

Github repository : https://github.com/varadvjoshi99/ey_hackathon

- Contains dataset for training
- Contains code for Rule-based Exception classification
- Contains code for ML based Exception classification
- Contains code for DL based Exception classification.
- Contains sample output file of test set

varadvjoshi99 / ey_hackathon

Unwatch 1 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

varadvjoshi99 Update README.md dc154fa now 6 commits

Datasets _ NLP based tagging solutio...	Add files via upload	25 days ago
ML_based.ipynb	Add files via upload	25 days ago
README.md	Update README.md	now
Rule_based.ipynb	Add files via upload	25 days ago
Word2Vec.ipynb	Add files via upload	4 days ago
output (2).xlsx	Add files via upload	25 days ago

README.md

EY_Hackathon

Contains file pertaining to EY-GDS Hackpions 2.0

Problem - NLP based tagging solution.

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Jupyter Notebook 100.0%

Thank You

Hackpions
GDS ES Hackathon
(edition 3)

Ernst & Young LLP

EY | Building a better working world

EY exists to build a better working world, helping to create long–term value for clients, people and society and build trust in the capital markets.

Enabled by data and technology, diverse EY teams in over 150 countries provide trust through assurance and help clients grow, transform and operate.

Working across assurance, consulting, law, strategy, tax and transactions, EY teams ask better questions to find new answers for the complex issues facing our world today.

EY refers to the global organization, and may refer to one or more, of the member firms of Ernst & Young Global Limited, each of which is a separate legal entity. Ernst & Young Global Limited, a UK company limited by guarantee, does not provide services to clients. Information about how EY collects and uses personal data and a description of the rights individuals have under data protection legislation are available via ey.com/privacy. EYG member firms do not practice law where prohibited by local laws. For more information about our organization, please visit ey.com.

ey.com