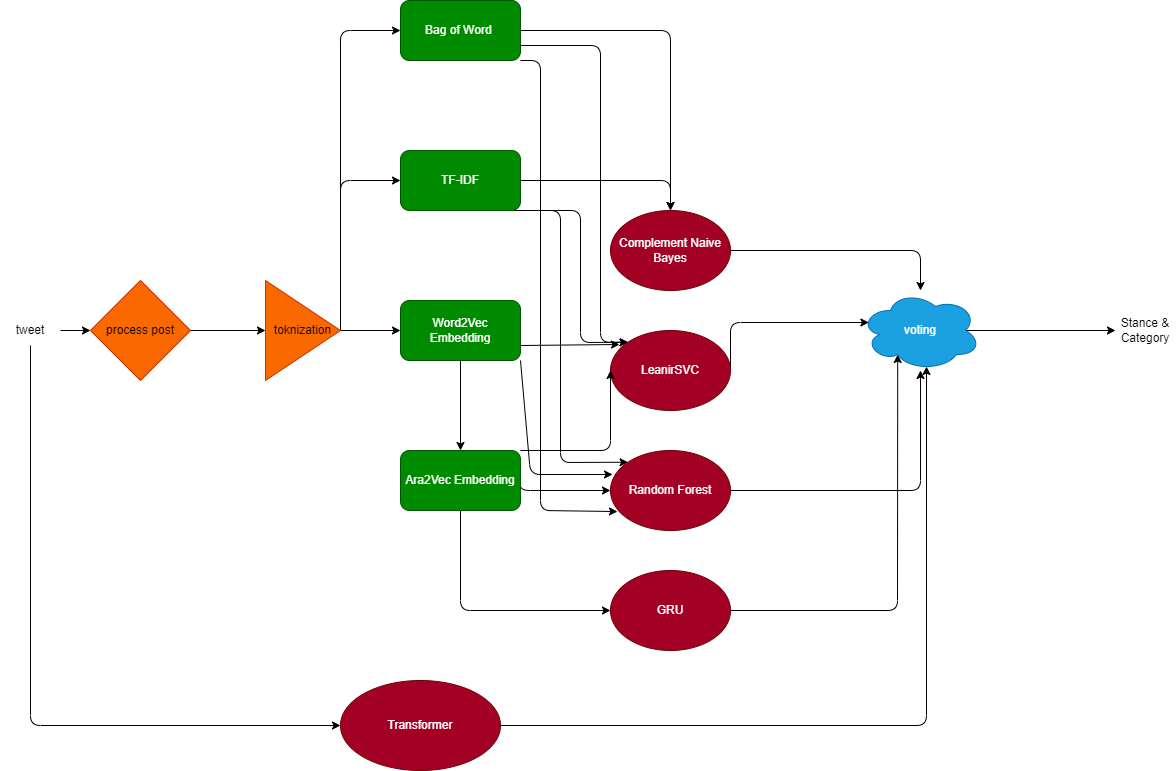
# Project Pipeline:



# A detailed description of each phase in your pipeline

## Data preprocessing:

In data pre-processing we try to remove any unrelated characters; like links, tags (<LF>), stop-words, hash-tags, punctuation, …

Then, we used lemmatization to get the roots. We tried more than one Lemmatizer but we didn’t find a huge difference.

## Feature extraction:

In this phase we tried TF-IDF as a powerful feature, especially, with classical machine learning algorithm (e.g. Naïve Bayes, SVC, …).

Also we tried Bag of Words and it was less than or similar to TF-IDF.

In word-embedding, we try to use pre-trained word-embedding feature, such as Word2Vec. Then we found pre-trained model on Arabic tweets. These word-embeddings help our models to learn faster.

Finally, we use contextual embedding in pre-trained transformer.

## Model training:

Classical models: we tried SVC, Random Forest and Naïve Bayes, and with each model we tried all the features mentioned above.

For Recursive models: we try LSTM and GRU, we try them with one hot encoding first, then we use Ara2Vec embedding as an initial hidden layer, finally we choose GRU.

For Deep Learning Recursive models: we used transformers, we try “asafaya/bert-arabic”. There were types of this model such as (mini, medium, …).

# Evaluation: Report the macro F1-score (and all other metrics you tried) for all trials you did.

Table 1 for "category" problem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | TF-IDF | BoW | CBOW | Ara2Vec | Bert-Arabic |
| LinearSVC | 0.32 | 0.30 | 0.05 | 0.14 |  |
| Random Forest | 0.28 | 0.29 | 0.16 | 0.18 |  |
| Complement NB | 0.25 | 0.28 |  |  |  |
| LSTM |  |  |  |  |  |
| GRU |  |  |  | 0.30 |  |
| Transformers |  |  |  |  | .34 |

Table 2 for "stance" problem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | TF-IDF | BoW | CBOW | Ara2Vec | Bert-Arabic |
| LinearSVC | 0.50 | 0.50 | 0.30 | 0.33 |  |
| Random Forest | 0.48 | 0.49 | 0.38 | 0.37 |  |
| Complement NB | 0.53 | 0.52 |  |  |  |
| LSTM |  |  |  |  |  |
| GRU |  |  |  | 0.53 |  |
| Transformers |  |  |  |  | .62 |

# Specify what model you used for the test set submission on Kaggle and the reason for choosing it.

Our Trials for final module:

* We were trying in three different ways; classical models, recursive learning model, deep learning model as shown above.
* After that, we want to concatenate best three or four models in voting system; in this system we sum the probabilities for given class from all modules, then take the argmax for classes probabilities. In merging three models (two classical and the recursive model) it increases the f1-score by 1~2% and it was good for us.

Then we try to merge this model with transformer model but it decreases the f1-score.

* The final conclusion is that we used only transformer to build the final output

Using pre trained model asafaya/bert-arabic, we used model type “mini”, number of epochs = 3, learning rate = 0.0001, and maximum length = 80.

# Trials.

**For unbalanced data:**

We tried different approaches to oversample the data but it all failed, so we go to train the model with weighted classes using attribute “class\_weight” we can assign the weights or just pass “balanced” to it and the model calculates the weights depending on classes distribution.

In Naïve Bayes model we found that complement Naïve Bayes can handle biased data. We try multinomial Naïve Bayes and complement, yet complement makes a better prediction.

**For recursive models:**

We tried first LSTM model, but we faced a problem with this model which is that it can’t learn the problem, the model just gets stuck in specific accuracy and it didn’t increase, even if we change any related parameters (e.g. learning rate, number of epochs, …)

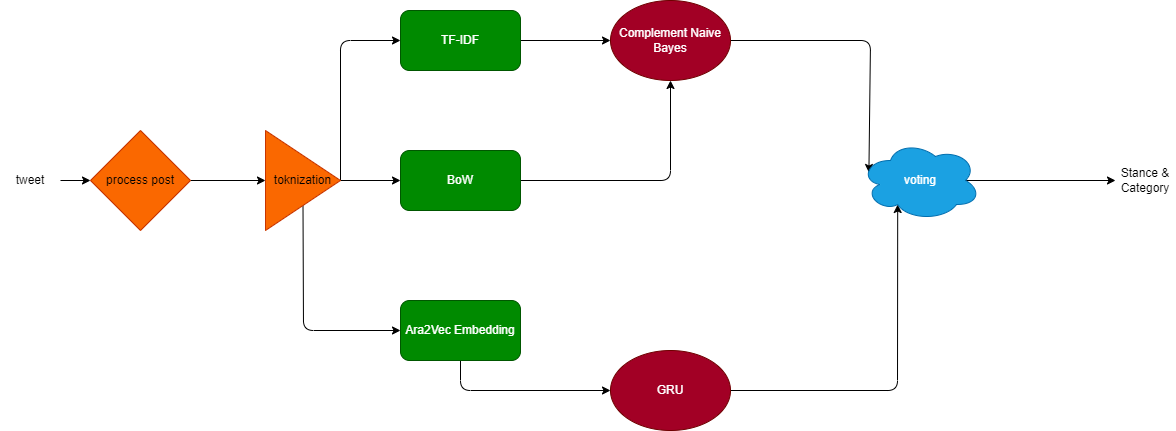
Then we tried GRU which was a better choice and with parameter tuning the model can over-fit the train data.

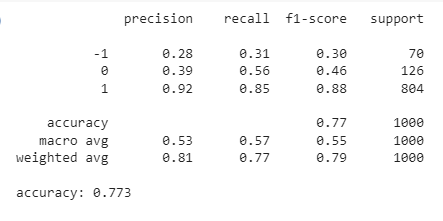
**For the voting system:**

We choose top 3 models in f1-score

* stance problem

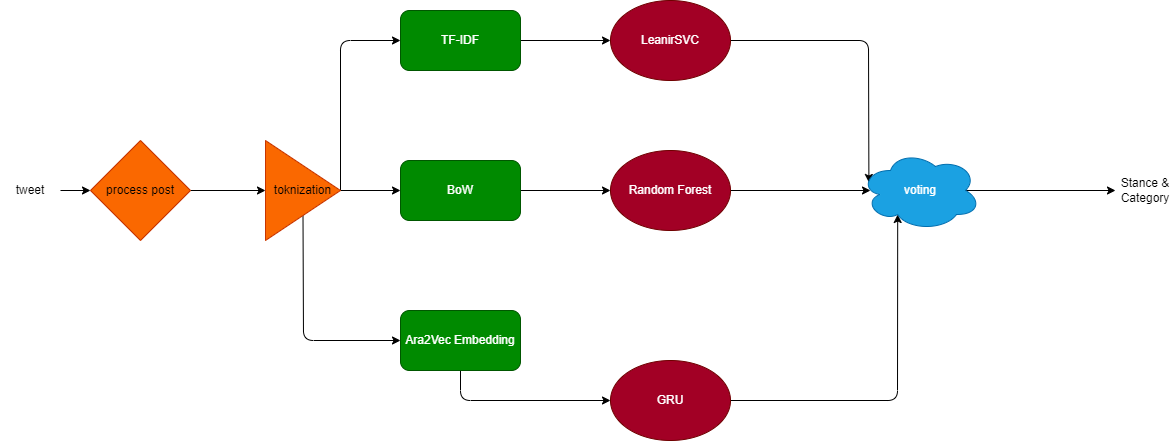
we use complement Naïve Bayes train on TF-IDF, complement Naïve Bayes train on Bag of Words, and GRU trained on Ara2Vec embedding.

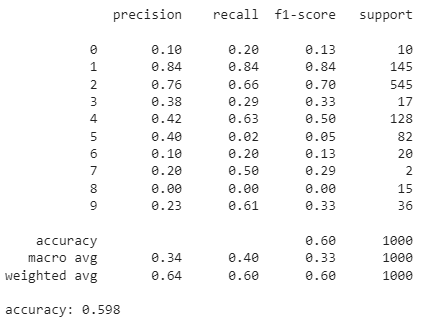




* category problem

we use Linear SVC train on TF-IDF, Random Forest train on Bag of Words, and GRU trained on Ara2Vec embedding.





**For transformer:**

We tried aub-mind/bert-base-arabertv02-twitter but the number of parameters was huge, it was crashing in training.

And we tried asafaya/bert-arabic, from model types we tried mini and medium; depending on the number of parameters and our data size, the performance was suitable with mini.

Length of tweet, number of epochs, and learning rate, are the parameters we followed in the papers and people’s opinions. It was the start point, but then we tuned it by trial and error.