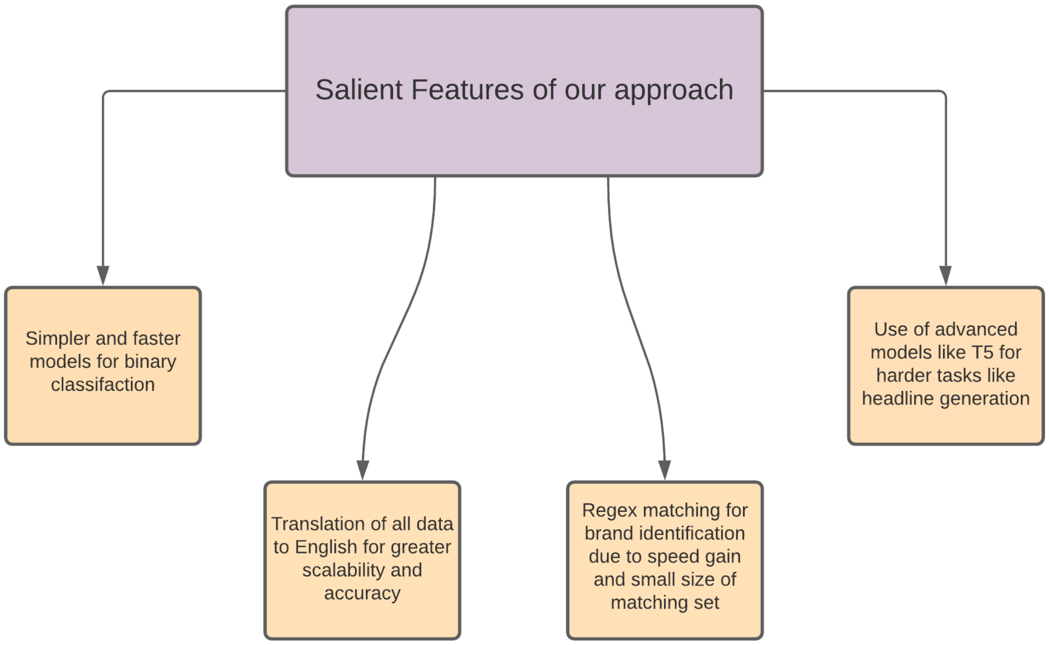
Our Approach:

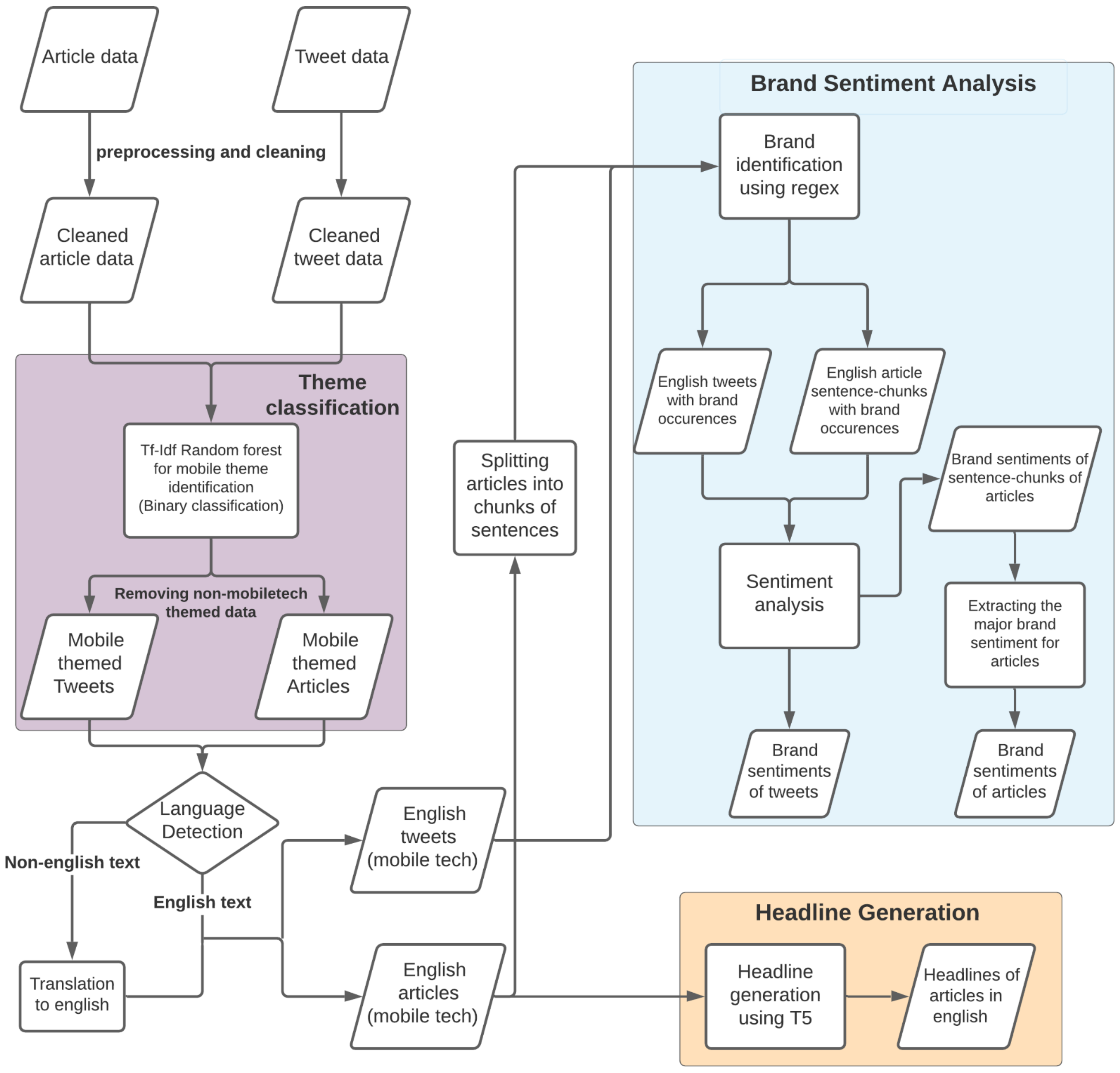

Salient Features of our approach
Simpler and faster models for binary classifaction
Translation of all data to English for greater scalability and accuracy
Regex matching for brand identification due to speed gain and small size of matching set
Use of advanced models like T5 for harder tasks like headline generation

We briefly explain the salient features of our approach here. In later pages, we explain each task in detail.

1) **Simpler and faster models for binary classification**
   - Binary classification for mobile-theme identification is not a very difficult task.
   - The amount of data being processed in this step is about 4 times that being processed in the other steps. This is because the ratio of mobile-themed to non-mobile themed data is about 1:3, and we only need to do the other tasks on mobile-themed data.
   - Therefore it makes sense to use simpler and faster models for this step.

2) **Translation of all data to english for headline generation and sentiment analysis**
   - Headline generation is a difficult task, which yielded poor results on multilingual data.
   - Translating all data to English language using an accurate model not only provides greater scope for scalability to additional languages, it even improves performance on other tasks for which we may already have superior pretrained models in English.
3) **Regex matching for brand identification**
   - The set of all possible mobile brands is a modestly-sized set
   - Using regex matching instead of framing it as an NER problem is much faster and often more reliable.
4) **Using advanced models like T5 for headline generation**
   - We tried a lot of possible variants but T5 performed the best.

A flowchart showing the complete end-to-end workflow


Article data
Tweet data
preprocessing and cleaning
Cleaned article data
Cleaned tweet data
Theme classification
Tf-Idf Random forest for mobile theme identification (Binary classification)
Removing non-mobiletech themed data
Mobile themed Tweets
Mobile themed Articles
Language Detection
Non-english text
English text
Translation to english
English tweets (mobile tech)
English articles (mobile tech)
Splitting articles into chunks of sentences
Brand Sentiment Analysis
Brand identification using regex
English tweets with brand occurences
English article sentence-chunks with brand occurences
Sentiment analysis
Brand sentiments of sentence-chunks of articles
Extracting the major brand sentiment for articles
Brand sentiments of tweets
Brand sentiments of articles
Headline Generation
Headline generation using T5
Headlines of articles in english

Train-test split and preprocessing:

- ❖ We randomly split the given data in a stratified manner for articles and tweets into an article training set (3000 samples), an article validation set (600 samples), an article test set (400 samples), a tweet training set (3000 samples), a tweet validation set (600 samples), and a tweet test set(400 samples) on which we performed all the experiments.
- ❖ We cleaned the data by removing redundant information present in both texts of articles and tweets. Eg removing links, dates, symbols, whitespaces, author name, twitter data specific term like QT & RT, etc.
- ❖ We decided to replace emojis unicodes with their corresponding text descriptors.
  An example tweet demonstrating the usefulness of this approach:
  'Samsung Galaxy S21 & S21 + Unboxing & First Look | Epic Game-Changer Smartphones | 5G: fire :: fire :: fire: via'

Binary Classification for Mobile Theme Identification:

- ❖ The classifier was built using the TF-IDF vectorizer combined with the xgboost classifier.
- ❖ The reason for using TF-IDF vectorizer is to give a higher priority to words which are in the document currently being considered over the rest of the given collection. This ensures that the classification model pays more attention to such words as they are mostly present in the document and less in the collection, as a result of which the information gain would be significantly higher which is essential for good classification.
- ❖ The key reason behind developing the central classifier as an XGBoost model over an ensemble based classifier was to increase the classification accuracy(precision and recall). The XGBoost classifier gets more accurate after each round of boosting. The increase in classification accuracy comes at a negligible cost of time. Also since it is a boosting algorithm, the final model can be smaller and more accurate.
- ❖ Accuracy and weighted f1 scores were the evaluation parameters.

Mobile brand identification and sentiment analysis:

- ❖ For Brand Identification we curated a list of brand names of around 150 mobile brands by scraping a few magazine websites and then we use regular expression string matching. This gives very good results, moreover it allows us the flexibility to directly improve our search expression to include corner cases.
- ❖ It could have been posed as a NER (Named Entity Recognition) task, but since the list of brands of interest is not too big, and also publicly listed, using a learning algorithm wasn't necessary.
- ❖ In our experiments we observed that using only English data instead a mix of Hindi, Hinglish, and English significantly improved performance for sentiment analysis and headline generation. Thus all our approaches here after use English data. We use Google's translation API service(unofficial free API) to translate the given data into English. In case of articles since the whole article cannot be translated in a single pass, so we split the articles into manageable chunks and translate them. We use google-trans-new Python package to access Google's API. As this is an unofficial use of Google Translate's web API, it is common courtesy to add an adequate time.sleep() so as to not flood their server and/or get flagged. Currently we add an average sleep time of 1.5 second per API call.

- Since we are using the unofficial free API provided by Google, the translation time is high and it might be difficult to deploy the system for real-time sentiment analysis and headline generation. Nevertheless, the paid API is moderately priced and value-addition is enormous. Hence we would recommend any service employing our methods to use the official paid Google Translate API.
- Since we had no dataset for brand specific sentiments, we tried curating one. From an external mobile product reviews data, we combined random pairs of texts into a single data point as an attempt to approximate the dataset distribution provided to us. We trained a model on this newly curated dataset, but the performance didn't carry on to a small subset of data given to as a part of the problem statement that we had hand labelled(since the sentiments weren’t provided to as).
- To improve our results we fine tuned model checkpoints(of BERT) that were trained specifically for Hinglish data.
- We have tried a couple of methods for brand specific sentiment extraction. FIrst we framed the problem as a text pair sequence classification task using BERT where the inputs are brand name along with the tweet/article text and added a linear layer to extract the sentiment. Another approach is to use only the contextualized representation of words within a particular distance of brand name.
- Another approach tried for sentiment analysis is that of incorporating dependencies between brand names and adjectives in the tweets and articles using their dependency trees.
- We have also used a Lexicon based approach which uses a pretrained dictionary of words with semantic scores. A sentiment score is assigned to each word (token). We calculate the total sentiment score towards a brand as a weighted sum of all the tokens. Weights are assigned relative to the position of the word with brand name.
- For handling articles with large word counts, we divide it into chunks of consumable sizes and combine the individual outputs for a final sentiment.
- For articles and tweets with only one brand we restrict ourselves to simpler models such TF-IDF based classifiers which reduce latency.
- We have used an ensemble of all the aforementioned approaches to compute the final sentiments for all brands.

Headline generation:

- We Fine Tuned T5(Text-to-Text Transfer Transformer)-base model for 2 epochs on the article train set, and tested on the article test set. Early stopping based on article validation set ROUGE, and BLEU.
- T5 has been trained on huge amounts of raw text(for example Wikipedia) in sequence to sequence fashion. T5 poses every problem as sequence to sequence problem and thus can be fine tuned for Natural Language Understanding tasks and Natural Language Generation tasks. T5 has only been trained on English data(a multilingual version mT5 has also been developed by the authors of T5).
- Since we translate Hindi and Hinglish data to English for sentiment analysis of articles on mobile technology, we use this same data for headline generation as well. Additionally we translate the headlines.
- We observed that since the data is extremely noisy, multilingual models are not able to deliver good performance. We observe a large gap in performance between the

multilingual model(mT5) trained on original data and the T5 model trained on translated data. Thus we fine tune our model on translated English data.

- We experimented with PEGASUS(Pre-training with Extracted Gap-sentences for Abstractive Summarization) as well. PEGASUS has been pre-trained for English abstractive summarization. But it is unable to beat our T5-base in terms of the evaluation metric, so we drop it.
- Likewise lead-3 baseline(first 3 sentences of article taken as the headline) is also not able to beat T5-base.

Innovation:

- We have combined several ideas to obtain what we think is a good performance/latency tradeoff.
- We save time by using simple TF-IDF based models for sentiment analysis of tweets and articles with one brand and use bigger pretrained models only if there are more brands in the same tweet/article.
- Likewise using TF-IDF based models for the initial binary classification reduces space and time complexity.
- Translation into English significantly improves performance. Time saved in sentiment analysis and binary classification is utilized for translation to significantly improve headline generation.

Scalability:

- Our proposed pipeline is scalable to many more languages since it translates the given text to English for sentiment analysis and headline generation.
- Since our pipeline uses only free services(and no paid cloud services), it is more scalable to domains other than the ones in this problem.
- Our pretrained models can be cached and stored which reduces inference time.
- Reproducibility of our code and results are fairly simple, so in case of availability of more data(for example there is continuous data inflow in case of online learning) our pipeline can easily be finetuned on it.
- Translating sentences is the bottleneck of our pipeline and may reduce the pipeline’s scalability. Using the paid API would alleviate this problem.