BitNews

Product Proposal – BitNews

Problem Statement

We are living in the age of information overload. If we do not stay up to date with the rising trends and news of the industry, it is very easy to get left behind. However, we often find it challenging to allocate time to read long news articles. Oftentimes, the actual piece of information makes up a small portion of the article and it may not be worth going through the entire thing. Also, every day there may be several articles that could contain information that we would like to know. So how can we stay up-to-date with all of these sources of information while cutting down the time spent?

Solution

We can use techniques in NLP and neural networks to solve this problem by building a summarizer. We aim to scrape data from a news feed (an RSS feed or a few popular news sites), extract the "important parts" from the article using a large pre-trained model, and present the output to the user as an easily digestible short paragraph.

This will be an unsupervised learning model and falls in the category of autonomous systems. Our products would fit well into the data flywheel architecture as we would know which articles users like to read. We can also have a feedback system to learn about our model's performance. This helps us maintain a feedback loop.

In this approach, we would gravitate towards an algorithmic modeling approach as we would be in a position to sacrifice interpretability for better model performance.

The ideal end product that flows out of this solution would be a fully automated AI-powered software that functions with zero user input. For example, we can use CI/CD architectures to pre-fetch an expected list of articles and store them as a cache on a VM. Upon request, these cached responses can be fetched by the user-facing front end, causing minimal load on the UI and almost instant load time faced by the end-user.

Expected Technologies

- NLP (nltk) for understanding text context and figuring out the most important parts of a large text
- GloVe/ Word2Vec: Word Embeddings for advanced text summarization.
- Computer Vision/ Image processing for selecting the best image from the article.
- StreamLit for the user-facing endpoint. Users can choose an article that they want to summarize
- VM for hosting the back-end code and daily caching for trending articles
- Git for code version control
- Scrapy for scraping news from websites.