COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

Automatic alert generation with NER and SA

Ulises Díez Santaolalla

Teresa Franco Corzo

Ignacio Felices Vera

Maria Ascanio Morcillo

Grupo A

Deep Learning - Natural Language Processing

3º Grado en Ingeniería Matemática e Inteligencia Artificial

1. SA

In the preprocessing phase, we cleaned and normalized the Sentiment140 dataset by converting text to lowercase, expanding contractions (e.g., "can't" → "can not"), and replacing usernames, URLs, hashtags, and emojis with special tokens like <USER>, <URL>, <HASHTAG>, and <EMOJI>. We standardized repeated characters and replaced emotion-marked words (e.g., happy) with tags like <GOOD_EMOTION>. The cleaned text was tokenized and saved in CSV format. To improve efficiency, we reduced the original dataset (1.6 million tweets) for experimentation. For model input, we used a pre-trained Word2Vec model to convert tokens to vectors and created a custom PyTorch Dataset class to handle tweets and labels. A collate_fn function with dynamic padding enabled efficient batch processing. We split the dataset into training, validation, and test sets, then created corresponding DataLoaders. To classify tweet sentiment, we built an RNN with a pre-trained embedding layer, LSTM, batch normalization, and a final linear layer for binary output. We tuned key LSTM parameters, used packed sequences, and trained the model with Adam optimizer and BCEWithLogitsLoss. Accuracy was computed using sigmoid activation and a 0.5 threshold. Since the dataset only contained positive and negative labels, we focused on binary sentiment classification.

2. NER

In the development of the NER model, we began by cleaning the CoNLL-2003 dataset and selecting the two essential columns: "tokens" and "ner_tags". These served as the input and target sequences for training our model. To represent the tokens, we used pretrained Word2Vec embeddings. We conducted research to explore different approaches to the model by reviewing relevant literature and papers. Based on this research, we concluded that the most effective architecture would be a bi-directional LSTM followed by a Conditional Random Field (CRF) layer. The bi-directional LSTM enables the model to capture context from both past and future words, resulting in a better understanding of each token’s meaning. The CRF layer, added on top of the LSTM, improves sequence labeling by considering the dependencies between output tags. This helps the model learn that some tag sequences are more likely than others (e.g: an I-tag should not start a sequence without a preceding B-tag) and assigns higher probabilities to more common tag transitions. Currently, the model achieves an accuracy of around 70%, but it shows signs of overfitting on the training data. Therefore, further tuning and improvements are needed to enhance generalization and overall performance.

3. AG

To implement the Alert Generation system, we first prepared a dataset with input text, sentiment information, named entities, and the original sentence. We also generated corresponding target texts. Initially, we tested models like Vision Transformer and T5, but their outputs often failed to explicitly mention sentiment or entities. We then created a custom function, generate alerts, to manually construct structured alerts, e.g., "Alert with a clearly positive tone. Involving organizations such as Bnei Yehuda." After creating the synthetic dataset, we fine-tuned a T5-small model on Google Colab, using the ROUGE metric to assess its ability to replicate target text structure and content. The next step is to validate the model's effectiveness and integrate it into a complete pipeline, migrating the process from Colab to a local environment.

4. Image Captioning

For the image captioning task, we implemented a pipeline using the pretrained BLIP model (Salesforce/blip-image-captioning-base). This model generates captions from images without requiring additional training. We tested it on 10 images, and it produced coherent and relevant descriptions. BLIP stood out as the best option due to its

lightweight design, smooth performance, and high-quality results. Since the model works well in a zero-shot setting, it serves as a solid baseline. Next, we plan to integrate the generated captions with NER and sentiment analysis models to evaluate the consistency of extracted entities and sentiments with the visual content in the images.