

PROJECT PROPOSAL

Introduction

For the final project we are planning on first completing the intermediate level and then adapting the project to include an image captioning model if we have enough time. We are approaching this project this way because we think it's important to first consider the main problem at hand and, once we have achieved this first milestone, we will work on investigating and deploying an image captioning model.

Embeddings

We are going to start using the pretrained embeddings that we have been given for this project, but we will evaluate other options as we continue advancing with our project. We can use static or dynamic embeddings. For the static we are going to start using the three examples recommended by the teacher: Word2Vec, GloVe, FastText. We are going to start using static embeddings for NER, but we will explore using dynamic embeddings if necessary.

NER + SA

Since we have not found a dataset that combines both NER + SA, we have decided that we are going to use a LLM model like deepseek or chatgpt that we can access through Python in order to generate the sentiments of a NER dataset. We have found some examples of NER datasets like: OntoNotes 5.0 Dataset, but we are going to continue investigating as we continue with our project.

[OntoNotes 5.0 Dataset | Papers With Code](#) -> various genres of text

[Name Entity Recognition \(NER\) Dataset](#)

[CrossNER Dataset | Papers With Code](#) -> data spanning over five diverse domains

Automatic Alert Generation

After training our NER + SA models, we aim to use a pre-trained transformer from Python transformers' library (<https://pypi.org/project/transformers/>). We would focus on BERT or Llama for text generation. We will train its last layer to adapt to our inputs and for our desired outputs. We would generate our inputs manually so that they fit the "REPUTATION RISK: [Organization] mentioned NEGATIVELY" structure or similar. This structured input will help the model learn how to generate concise and relevant alerts based on NER + SA outputs. If this approach does not work, we will explore fine-tuning the entire model to improve performance and alignment with our alert generation task.

Image Captioning model

As we have said before, we will first focus on the intermediate level model. Once we have the models well trained, ensure they work properly, and if we have enough time, we will move on to the final model, which is based on adding image captioning.

Our idea, if we reach this point, is to use images of datasets from the internet and even take news images that come with text to use them as inputs. Once we have the inputs ready, we will start training this model, based on CNN + RNN, with the goal of obtaining a descriptive text of the image as output. This will allow us to combine it with the other text input and proceed with the next steps of the final model.

We have found some datasets, one of them is COCO ([MS COCO Dataset | Papers With Code](#)), which is a large-scale dataset for object detection, segmentation and image captioning, containing images with detailed annotations of everyday scenes. Others are this one: [Image captioning](#), which recognises different pokemons, and the last one is: [Image captioning](#) based on images from the Flickr dataset.