DL+NLP Final Project: Automatic alert generation with NER and SA

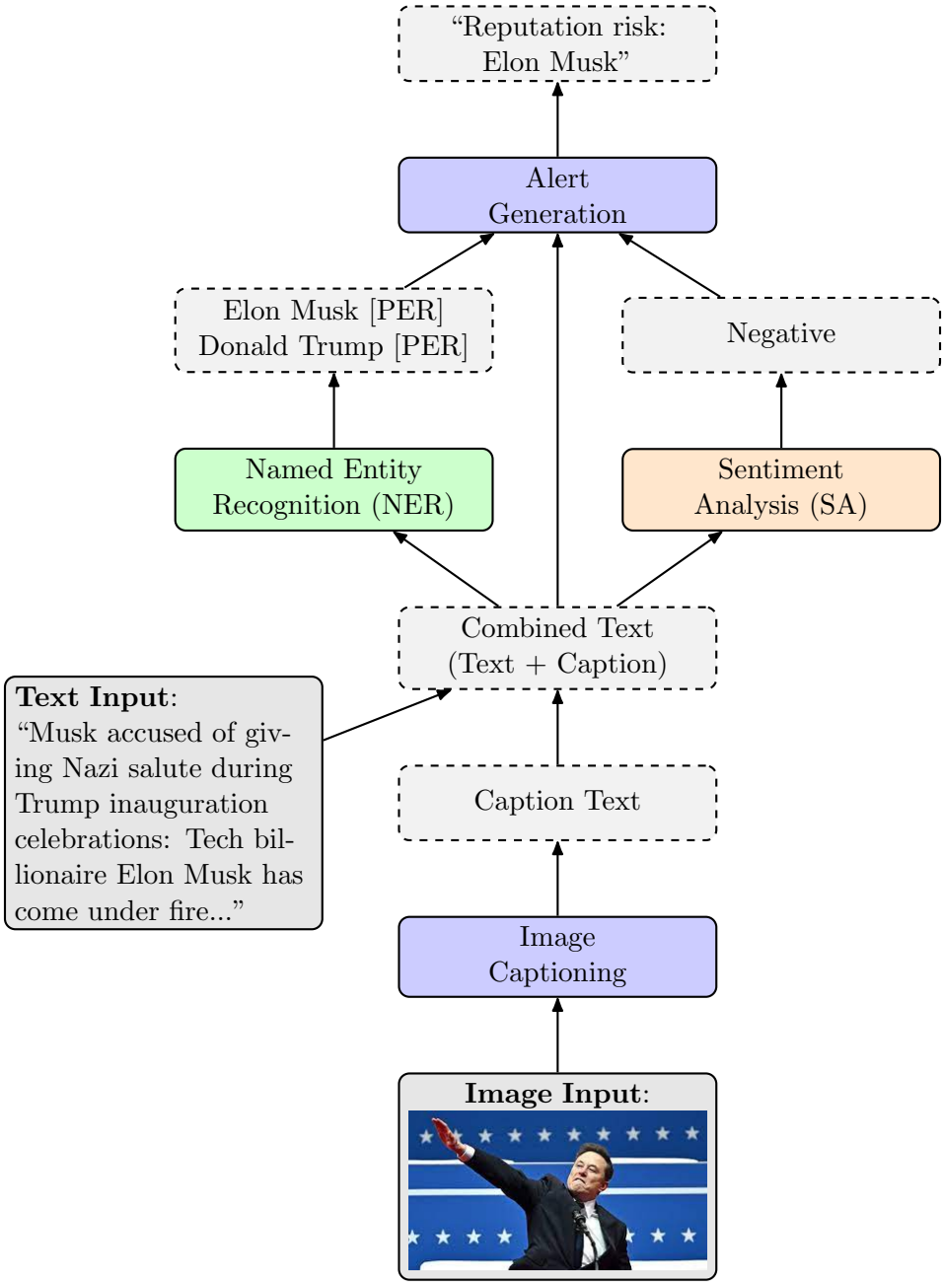
Project Description

This project involves the development of an automatic alert generation system from news articles and social media posts. The system processes news articles and social media posts (and optionally images) to produce concise, context-specific alerts. The system combines Named Entity Recognition (NER) and Sentiment Analysis (SA) techniques to generate alerts for reputation tracking, economic updates, geopolitical risks, etc. The implemented model should be comprised of the following components:

1. **Named Entity Recognition (NER):** Identifies entities such as people, organizations, monetary values, locations, etc., in the input text, using a custom-trained LSTM network.
2. **Sentiment Analysis (SA):** Classifies the sentiment of the input text (positive, neutral, negative) with another neural network.
3. **Alert Generation (AG):** Combines outputs from NER and sentiment analysis to create context-specific alerts.

Example: Input/Output

Here's an example of how the system could **ideally** work. The system processes a multimodal input consisting of a news article (text) and an associated image. In this example, the news item presents Elon Musk in a negative light. The model performs image captioning to generate descriptive text from the image, which is then combined with the original article. A Named Entity Recognition (NER) module identifies key individuals mentioned in the text, while a Sentiment Analysis (SA) module assesses the tone of the content. Finally, an Alert Generation (AG) module consolidates the extracted information to generate an alert indicating a potential reputational risk. In this case, the output is: "Reputation risk: Elon Musk."



Project goals:

- **Implement and train models** for NER and SA (e.g., using LSTM-based or other neural architectures).
- **Integrate both tasks** into a pipeline or joint model to automatically generate relevant alerts.
- **Optionally incorporate image data as input** to generate alerts based on both text and associated photos.

Group formation

- **Group size:** In IMAT A (37 students), there will be 7 teams of 4 students and 3 teams of 3 students. In IMAT B (37 students), there will also be 7 teams of 4 students and 3 teams of 3 students. The expectations for team work will be adjusted accordingly for the smaller teams.
- **Formation strategy:** Students will form their own teams. Each team should enter their members in this document. A nickname for the team is optional but encouraged.

Project levels

There are three types of models that can be developed, each granting access to a higher maximum grade. Figure 1 summarizes the model expectations at each level.

Part of the project involves finding suitable datasets, preparing the data, and making sure they work for both Named Entity Recognition (NER) and Sentiment Analysis (SA).

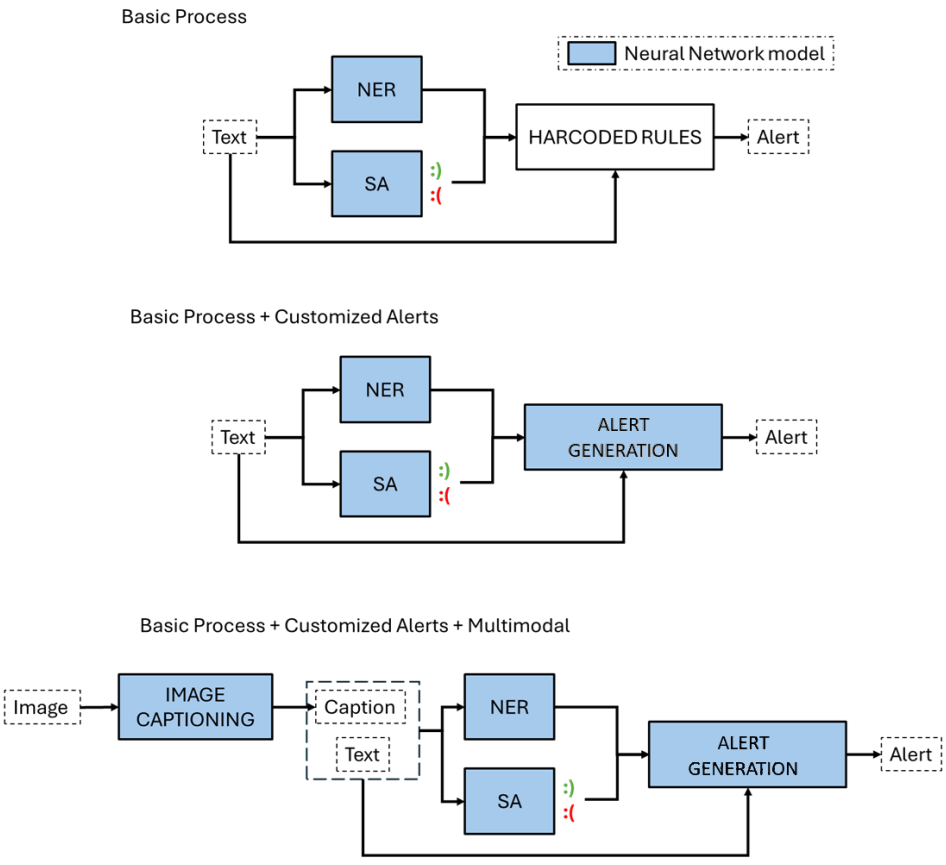


Figure 1: Scheme with the neural network models expected in each of the levels described.

1. Basic level (up to 7.0 points)

- Train separate **NER** and **SA** models.
- Implement **rule-based alert generation** via hard-coded rules: E.g.,

  IF [`Organization`] entity is found and sentiment is [`negative`], **THEN** generate the following alert:
  "⚠ `REPUTATION RISK: [Organization] mentioned NEGATIVELY.`".

2. Intermediate level (up to 9.0 points)

- Develop a **joint architecture** for simultaneous NER and sentiment analysis tasks.
- Implement a **combined loss function** to optimize both tasks jointly.
- Create an **automatic alert generator** that:
  - Takes the NER + SA outputs and optionally the original text as inputs.
  - Uses a neural model to generate a concise alert from these inputs. You will need to think creatively about this step. Consider how the alert generation problem can be framed as a learning task, with a neural network providing at least part of the solution.
  - The alert should reference the entities identified by the NER model and reflect the sentiment detected by the SA model.

3. Advanced level (up to 10.0 points)

- **Process images** as part of the input, alongside the text.
- Implement a **image captioning** model (CNN + RNN architecture) to generate captions from the associated image. Or, at least, use a pre-trained image captioning model.
- Combine the generated captions with the textual input to:
  - Either feed both text + caption to the NER and SA models.
  - Or integrate the caption with the NER + SA outputs in a final seq2seq model to produce alerts.

For the automatic alert generator and the image captioning models **you might want to use a pretrained model** and, possibly, finetune it to your task. You may want to consider creating a small synthetic dataset to train the alert generation module. It may be helpful to use a generative model such as ChatGPT to create the training examples.

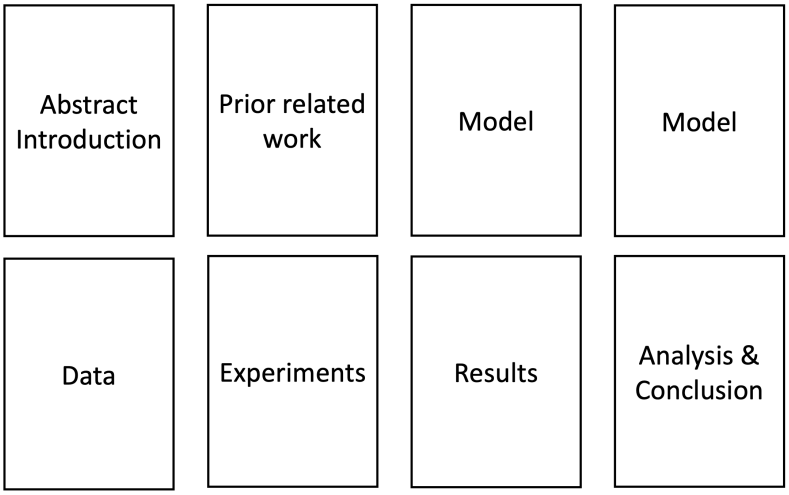
Deliverables

1. **Project proposal (Deadline: 31 March)**
   - Literature review and initial research ideas.

- Project plan and milestones.
- Each team will submit a **1-page document** explaining their proposal.

2. **Project progress report (Deadline: 10th April)**
   - Problem definition and technical approach.
   - Initial implementation and preliminary experimental results.
   - Detailed plan for the next stages.
   - Each team will submit a **1-page document** reporting their progress.
3. **Final deliverable (Deadline: 21st April)**
   - **Scientific paper:** maximum 8 pages, in LaTeX format.
   - Complete, documented, and reproducible **code**. A README file is expected with instructions on how to obtain the results presented in the paper. These instructions should, at least, clearly explain:
     - how to retrain your models.
     - how to perform a prediction on a newly created data sample in the format required by your solution.
   - **Team declaration** outlining roles and contributions.

   The scientific paper may only be up to 8 pages long. Here's how you may want to distribute pages, roughly:



   To format your scientific paper, you must use the **NeurIPS LaTeX template**, which follows a standardized format for academic papers. NeurIPS is a leading machine learning conference. You can download the LaTeX template and style guidelines from the following link: NeurIPS Template and Style Guidelines.

Evaluation criteria

Important: The focus is **not** on achieving state-of-the-art performance, but on designing a **well-thought-out model that demonstrates a solid understanding of the NLP and deep learning techniques covered in class**. Performance can often be improved with further training or more advanced architectures (e.g., Transformers). **The most important aspect is applying techniques correctly and analyzing the results rigorously**.

The project will be evaluated based on:

- Clarity, rigor, and structure of the scientific paper.
- Technical correctness, efficiency, and performance of the implemented models.
- Originality and soundness of the proposed model, particularly for intermediate and advanced-level implementations.
- Analysis of the sensitivity of the models to changes in the inputs, analysing the strengths and weaknesses of your solution and future developments that might be applied to your solution to improve the results.

Again, **a well-executed and well-written analysis of the state of the art, available datasets, and techniques used in the project would be far more valuable than a highly technical, well-performing solution.**

Initial resources

Below are some initial resources to help you get started on building your automatic alert generation system. These suggestions are just starting points—you're encouraged to explore and identify datasets that best match your project context and goals.

- **NER datasets:**
  - CoNLL-2003 – A benchmark dataset featuring news articles annotated with entities such as people, organizations, and locations.
  - OntoNotes 5.0 – Offers extensive annotations for various entity types found in news and other texts.
- **Sentiment analysis datasets:**
  - Sentiment140 – Contains 1.6 million annotated tweets, making it well-suited for social media sentiment analysis.
  - Financial PhraseBank – Focused on financial news, this dataset is useful for capturing sentiment in an economic context.
- **Recommended tools to get started:**
  - spaCy – spaCy offers pretrained NER models to test your ideas.

- ChatLabs AI Image Caption Generator – An online tool for generating image captions and gaining a hands-on understanding of image captioning. You can try out the system with Elon Musk's image provided above, and see what you get.
- TensorBoard – For visualizing and tracking your model's training progress.

Remember, these resources are only suggestions. A significant part of your project is to research, select, and preprocess the datasets that best align with your approach and the task of generating context-specific alerts.