



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

ICADE

CIHS

Automatic alert generation with NER and SA

Ulises Díez Santaolalla

Teresa Franco Corzo

Ignacio Felices Vera

Grupo A

Deep Learning - Natural Language Processing

3º Grado en Ingeniería Matemática e Inteligencia Artificial

Project Plan and Milestones

Milestones	Planification	Issues
Research and Data Collection	1 st Week (23-31 March)	<ol style="list-style-type: none">1. Conduct a literature review.2. Collect datasets (CoNLL-2003, Sentiment140, flickr30k).3. Set basics and approach.4. Set up work environment.
Model Selection and Preprocessing	2 nd -3 rd Week (31-10 April)	<ol style="list-style-type: none">1. Preprocess data (tokenization, tensors).2. Obtain pretrained embeddings (Glove, word2vec)3. Create the LSTM NER model and SA.4. Alert Generation model.5. Train/test models.
Multi-Modal Integration	3 rd – 4 th Week (10-17 April)	<ol style="list-style-type: none">1. Combine NER + SA2. Choose a mechanism for image captioning.3. Combine NER, SA, and image captioning outputs.4. Room for whatever not made before.
Evaluation and Optimization	4 th – 5 th Week (17-21 April)	<ol style="list-style-type: none">1. Conduct final testing and prepare for deployment.2. Write comprehensive documentation and Latex report.

Research and Data Collection

To train NER and Sentiment Analysis (SA) models from scratch, we will use PyTorch and implement architectures based on recurrent neural networks (RNNs) such as LSTMs and GRUs, avoiding the use of Transformers. Input text will be encoded using pretrained word embeddings, such as GloVe (from Stanford) or Word2Vec (from Google), which provide rich semantic representations.

For Named Entity Recognition (NER), we will implement a BiLSTM (bidirectional LSTM) to process text sequences. Optionally, this will be followed by a Conditional Random Field (CRF) layer to improve sequence labeling by modeling dependencies between output tags. There are multiple PyTorch tutorials and GitHub repositories that provide guidance and data handling utilities for this purpose.

For Sentiment Analysis (SA), we will follow a similar approach: input sequences will be converted into pretrained embeddings and passed through an RNN, typically an LSTM or GRU. The final hidden state will be used as a fixed-size representation of the input and passed through fully connected layers to predict sentiment polarity (positive, negative, neutral).

To complete the optional advanced component, we will integrate pretrained image captioning models. Among the available options, we opt for Transformer-based architectures over traditional CNN+RNN pipelines, as they have shown better performance lately. These models typically combine a visual encoder (ViT) with a language decoder trained specifically for generating image descriptions.

For the alert generation system, rather than relying on handcrafted rules, we plan to use a pretrained language generation model, such as a Seq2Seq model or a Transformer decoder, able of generating summaries by integrating outputs from both the NER and SA modules (and optionally, the image captioning component).

A thorough study will be done to preprocess and treat data before feeding it into the models.

Bibliography

1. Baaraban, A. (n.d.). *pytorch_ner*. GitHub. Retrieved from: https://github.com/baaraban/pytorch_ner
2. PyTorch Team. (n.d.). *Advanced: Bi-LSTM Conditional Random Field (CRF)*. PyTorch Tutorials. Retrieved from: https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html
3. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). *Neural Architectures for Named Entity Recognition*. arXiv preprint arXiv:1603.01360. Retrieved from: <https://arxiv.org/abs/1603.01360>
4. NBShare. (n.d.). *Tweet Sentiment Analysis Using LSTM With PyTorch*. Retrieved from: <https://www.nbshare.io/notebook/754493525/Tweet-Sentiment-Analysis-Using-LSTM-With-PyTorch>
5. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805. Retrieved from: <https://arxiv.org/abs/1801.07883>
6. Hever, G. (2021). *Sentiment Analysis with PyTorch – Part 4: LSTM & BiLSTM*. Medium. Retrieved from: <https://galhever.medium.com/sentiment-analysis-with-pytorch-part-4-lstm-bilstm-model-84447f6c4525>
7. Franke, M. (n.d.). *Image Captioning with Pretrained Vision-Language Models*. Retrieved from: <https://michael-franke.github.io/npNLG/08-grounded-LMs/08c-NIC-pretrained.html>
8. Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). *Show and Tell: A Neural Image Caption Generator*. arXiv preprint arXiv:1411.4555. Retrieved from: <https://arxiv.org/abs/1411.4555>
9. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. arXiv preprint arXiv:1502.03044. Retrieved from: <https://arxiv.org/abs/1502.03044>
10. Li, J., Selvaraju, R. R., Gotmare, A., Joty, S., Xiong, C., & Hoi, S. C. (2022). *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. arXiv preprint arXiv:1910.13461. Retrieved from: <https://arxiv.org/abs/1910.13461>